



## Aspen Plus

---

*Getting Started Customizing Unit Operation  
Models*

## **Version Number: V8.0 December 2012**

Copyright (c) 1981-2012 by Aspen Technology, Inc. All rights reserved.

Aspen Plus, aspenONE, the aspen leaf logo and Plantelligence and Enterprise Optimization are trademarks or registered trademarks of Aspen Technology, Inc., Burlington, MA.

All other brand and product names are trademarks or registered trademarks of their respective companies.

This document is intended as a guide to using AspenTech's software. This documentation contains AspenTech proprietary and confidential information and may not be disclosed, used, or copied without the prior consent of AspenTech or as set forth in the applicable license agreement. Users are solely responsible for the proper use of the software and the application of the results obtained.

Although AspenTech has tested the software and reviewed the documentation, the sole warranty for the software may be found in the applicable license agreement between AspenTech and the user. ASPENTECH MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THIS DOCUMENTATION, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

Aspen Technology, Inc.  
200 Wheeler Road  
Burlington, MA 01803-5501  
USA  
Phone: (1) (781) 221-6400  
Toll Free: (1) (888) 996-7100  
URL: <http://www.aspentech.com>

# Contents

<b>Who Should Read this Guide .....</b>	<b>1</b>
<b>Introducing Aspen Plus .....</b>	<b>3</b>
Sessions in this Manual .....	3
Files Created in these Sessions .....	3
Integration Features for Custom Unit Operation Models.....	4
Aspen Plus Excel Unit Operations .....	5
Aspen Plus Fortran Models.....	5
Supporting Files .....	5
Related Documentation .....	5
Technical Support .....	6
<b>1 Creating an Excel Unit Operation Model .....</b>	<b>7</b>
Defining the Simulation.....	7
Setting Up the Model in Aspen Plus.....	8
Starting Aspen Plus .....	8
Specifying the Properties.....	9
Building the Process Flowsheet .....	9
Entering Title, Components, Base Method, and Feed Specifications.....	10
Entering Excel Path and User Array Data .....	11
Setting up a Product Stream Flash .....	12
Setting Up the Excel Model .....	13
Copying and Examining the Excel Template .....	13
Editing the Excel Sheets.....	14
Entering Membrane Model Equations and Parameters .....	18
Running the Excel Model .....	19
Examining Simulation Results .....	20
Viewing the Results from the Excel Model.....	20
Viewing the Block Results.....	22
<b>2 Customizing the Excel/Aspen Plus Interface.....</b>	<b>25</b>
Revising the Excel Model .....	25
Revising the User Subroutine.....	28
Code Listing – Subroutine MEM2 .....	28
Code Description – Subroutine MEM2.....	36
Compiling and Linking the Subroutine .....	37
Compiling the Excel Interface Subroutine .....	37
Creating a Shared Library.....	38
Running the Simulation.....	39
Results from the Custom Excel Model .....	41

<b>3 Customizing the Aspen Plus Model Library .....</b>	<b>45</b>
Creating a Model Library .....	45
Creating an Empty Custom Model Library .....	46
Creating a Template for Your Custom Model .....	46
Creating the Custom Model from the Template.....	47
Editing the Custom Model.....	50
Creating a Custom Icon .....	50
Creating References to the Real and Integer Parameters .....	51
Inserting the New Model and Running the Simulation .....	53
Inserting the New Model .....	53
Checking and Editing Setup Sheets.....	53
Compiling, Linking, and Running .....	54
<b>4 Creating an Aspen Plus Fortran Model.....</b>	<b>57</b>
Putting the Model Equations in the User Subroutine.....	57
Code Listing – Subroutine HFUM .....	58
Code Description – Subroutine HFUM.....	62
Compiling and Linking the Subroutine.....	64
Running the Simulation.....	65
Results from the Custom Fortran Model.....	65

# Who Should Read this Guide

This guide is suitable for advanced Aspen Plus users who want to customize the unit operation models available in Aspen Plus and create new models. You should be familiar with the procedures covered in *Aspen Plus Getting Started Building and Running a Process Model* before starting these examples.



# Introducing Aspen Plus

Aspen Plus offers the option of using custom or proprietary models in Aspen Plus simulations. The tutorials in this guide illustrate key features of this capability by building a simulation based on an external membrane model and performing the model calculations using either an Excel file or a Fortran subroutine.

This guide assumes that you have installed Aspen Plus, Fortran development software, Visual Basic development software, and Microsoft Excel on your computer. This guide is designed for advanced users of Aspen Plus who have a working knowledge of Fortran, Visual Basic, and Excel.

Even though most users of the customizing options described in this guide are highly experienced with Aspen Plus, anyone who has completed the tutorials in *Getting Started Building and Running a Process Model*, will be able to work through the tutorials here by following the step-by-step instructions.

## Sessions in this Manual

Follow the steps in this chapter	To learn how to
1 Creating an Excel Unit Operation Model	Create a simulation that uses an Excel model to determine product stream properties.
2 Customizing the Excel/Aspen Plus Interface	Alter the Fortran interface subroutine to accommodate an expanded Excel model.
3 Customizing the Aspen Plus Model Library	Create a model with pre-defined parameters for repeated use.
4 Creating an Aspen Plus Fortran Model	Perform model calculations in a Fortran subroutine instead of in an Excel spreadsheet.

## Files Created in these Sessions

Filename	Chpt	Description
Membrane1.bkp/apwz	1	USER2 model, Excel calculations, default interface
MemCalc_PreRun.xls	1	Excel file for use with default interface (before simulation run)

Filename	Chpt	Description
MemCalc.xls	1	Excel file after the run
MemCalc_Custom_PreRun.xls	2	Excel file w/ viscosity and cgs units for use with custom interface
MemCalc_Custom.xls	2	Excel file after the run
mem2.f/obj	2	Code for custom interface
membrane2_List_OBJs.opt *	2	DLOPT file w/ location of Mem2.obj
membrane2.dll	2	DLL file w/ Mem2 subroutine
membrane2_Loc_DLL.opt *	2	DLOPT w/ location of Membrane2.dll
membrane2.bkp/apwz	2	USER2 model, Excel calculations, custom interface
Ultrafiltration.apm	3	Custom Model Library
mem2Vars.f/obj	3	Code written by Aspen Plus for named references
membrane2_List_OBJs.opt *	3	DLOPT w/ location of Mem2.obj and Mem2Vars.obj
membrane2.dll	3	DLL w/ Mem2 and Mem2Vars
membrane2_cm.bkp/apwz	3	Custom model, Excel calculations custom interface
HFUM.f/obj	4	Code w/ model equations
HFUM_List_OBJs.opt *	4	DLOPT w/ location of HFUM.obj and Mem2Vars.obj
hfum.dll	4	DLL w/ HFUM and Mem2Vars
HFUM_Loc_DLL.opt *	4	DLOPT w/ loc. of HFUM.dll
hfum.bkp/apwz	4	Custom model, Fortran calculations, no interface

\* If you use pre-supplied DLOPT files, you will need to edit them to match the path of your working folder.

## Integration Features for Custom Unit Operation Models

Aspen Plus offers powerful features that make it easy to use Aspen Plus simulations containing proprietary models. You can create:

- Custom model libraries
- User-defined variables that become part of the Aspen Plus data structure as an alternative to Real and Integer arrays
- Custom icons to better represent the equipment that your models describe



# Aspen Plus Excel Unit Operations

You can use an Excel Workbook as an Aspen Plus user-defined unit operation model. Once you identify the input variables and physical properties used by Excel, you can develop and test the model within Excel without performing iterative compile and link steps that are required when developing and testing a Fortran model. You can utilize the full power of Excel including all of its built-in functions and specialized Visual Basic routines that are provided in the Aspen Plus Excel template.

## Aspen Plus Fortran Models

Aspen Plus provides utilities for convenient read and write access to named user-defined variables from within the Fortran user model routine. This simplifies user-defined parameter mapping into external user routines.

You can develop rich data structures for user-defined unit operation models involving integer, real and character data. The parameters can either be scalar or vector. Vector data can be automatically dimensioned based on nine standard lengths such as **Number of Components** or **Number of Inlet Streams**, or can be dynamically dimensioned based on a user-defined integer variable.

## Supporting Files

The Aspen Plus backup, Excel, Fortran and Visual Basic files described in this book are delivered in the GUI\xmp\GSG\_Custom directory of the Aspen Plus installation.

## Related Documentation

Title	Content
Aspen Plus Getting Started Building and Running a Process Model	Tutorials covering basic use of Aspen Plus. A prerequisite for the other Getting Started guides
Aspen Plus Getting Started Using Equation Oriented Modeling	Tutorials covering the use of equation-oriented models in Aspen Plus
Aspen Plus Getting Started Modeling Processes with Solids	Tutorials covering the Aspen Plus features designed to handle solids
Aspen Plus Getting Started Modeling Processes with Electrolytes	Tutorials covering the Aspen Plus features designed to handle electrolytes

<b>Title</b>	<b>Content</b>
Aspen Engineering Suite Installation Manual	Instructions for installing Aspen Plus and other Aspen Engineering Suite products
Aspen Plus User Guide	Procedures for using Aspen Plus
Aspen Plus User Models	Specifications for custom models

## Technical Support

AspenTech customers with a valid license and software maintenance agreement can register to access the online AspenTech Support Center at:

<http://support.aspentech.com>

This Web support site allows you to:

- Access current product documentation
- Search for tech 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
- Send suggestions
- Report product defects
- Review lists of known deficiencies and defects

Registered users can also subscribe to our Technical Support e-Bulletins. These e-Bulletins are used to alert users to important technical support information such as:

- Technical advisories
- Product updates and releases

Customer support is also available by phone, fax, and email. The most up-to-date contact information is available at the AspenTech Support Center at <http://support.aspentech.com>.

# 1 Creating an Excel Unit Operation Model

Aspen Plus offers several interfaces for including custom or proprietary models in Aspen Plus simulations. Among these is the option to use a User2 unit operation block in your Aspen Plus simulation with an Excel spreadsheet to perform the calculations. In this session, you will create a User2 Excel model for an ultrafiltration membrane.

First, use Aspen Plus to specify the required properties, build a process flowsheet, specify feed and product streams, and enter real and integer parameters corresponding to the membrane. Then use Excel to create a spreadsheet to calculate product stream properties. Aspen Plus will write data to and read data from the Excel spreadsheet.

The tutorial sessions in this Getting Started Guide assume that you have completed all of the sessions in *Getting Started Building and Running a Process Model*. In particular, you must be able to:

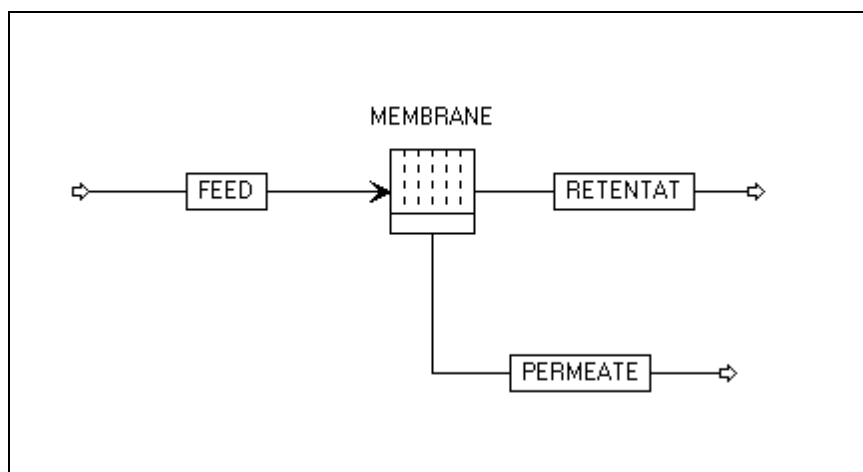
- Start Aspen Plus (using a blank simulation, template, or existing simulation).
- Navigate from form to form using the Navigation Pane.
- Enter data into the required fields in the input sheets.
- Place blocks and streams.
- Run a simulation.
- View the data in the results sheets.
- Save a simulation.

Allow about 60 minutes for this session.

## Defining the Simulation

Construct an Ultrafiltration Membrane model. Ultrafiltration is a pressure-driven membrane process used to separate components based on molecular size and shape. The solvent and small solute species pass through the membrane and are termed the **permeate**. Large solute species, such as proteins, are retained by the membrane and are termed the **retentate**.

The simulation that you will construct is shown below in Figure 1.1. There is one feed stream (a water-protein feed), one unit operation block (an ultrafiltration membrane), and two product streams (permeate and retentate).



**Figure 1.1 Ultrafiltration Membrane Model**

Aspen Plus will write feed stream data and real and integer membrane parameters (such as physical dimensions) to Excel. Excel will calculate product stream data such as mole flow rates and solute concentrations using a user-specified set of equations. Aspen Plus will read the product stream data from Excel. Results will be displayed in the Excel file and on Aspen Plus results forms.

## Setting Up the Model in Aspen Plus

### Starting Aspen Plus

- 1 From your desktop, click **Start** and then select **Programs**.
- 2 Select **AspenTech | Process Modeling <version> | Aspen Plus | Aspen Plus <version>**.  
The **Start Page** appears in the Aspen Plus main window.
- 3 Click **New**.  
The **New** dialog box appears. Use this dialog box to specify the template for the new run.
- 4 Click **Create**.  
Aspen Plus will open a blank Simulation.


For this customized simulation, Aspen Plus will handle everything but the calculations that correspond to the ultrafiltration process itself. First, specify the required properties.

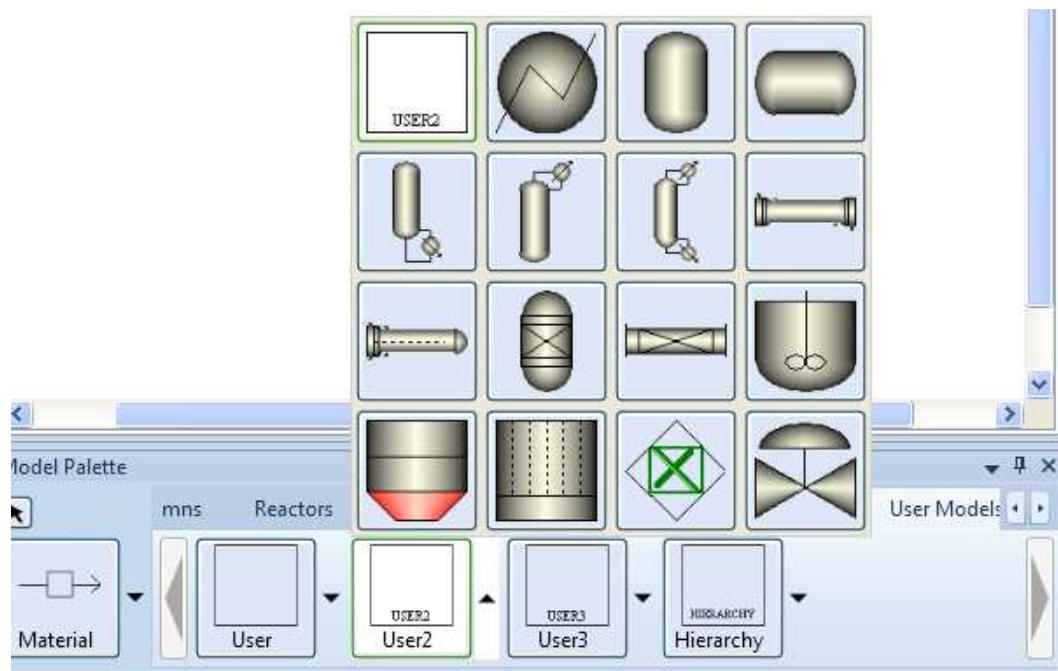
## Specifying the Properties

- 1 On the **Components - Specifications | Selection** sheet, type **WATER** and press **Enter** on the keyboard in the first cell under **Component ID**.  
The **Component name** and **Alias** are automatically filled in because Aspen Plus recognizes the ID **WATER**. Another row opens up in the table.
- 2 In the next **Component ID** field, type **PROTEIN** and press **Enter** on the keyboard.  
Protein is not in the Aspen Plus databank, so the **Component name** and **Alias** fields remain blank.
- 3 In the PROTEIN row, type **LYSINE** in the **Component name** field, and press **Enter** on the keyboard.  
Aspen Plus recognizes the amino acid lysine and fills in the **Alias** field. Lysine will stand in as a protein for the purposes of this illustration.
- 4 Go to the **Methods - Specifications | Global** sheet and select **IDEAL** in the **Base method** field.
- 5 On the **Home** tab of the ribbon, in **Units**, make sure that the **Unit Sets** is **METCBAR**.

Next, build the Flowsheet in the Simulation environment.

## Building the Process Flowsheet

- 1 Click . Select **Go to Simulation environment** and click **OK** to go to the Simulation environment.
- 2 From the Model Palette, click the **User Models** tab.
- 3 Click the arrow next to the **User 2** block icon to display all the User 2 model icons.



- 4 Move the mouse over the icons to see a name (or label) for each.

- 5 Locate the User 2 icon **FILTER** in the fourth row of icons, and select it.
- 6 Drag the **FILTER** icon to the **Main Flowsheet** window.
- 7 Create one input stream and two product streams.
- 8 Name the input stream **FEED** by selecting the stream or its label, right-clicking, and selecting **Rename Stream**.
- 9 Name the first product stream you created **RETENTAT**. Aspen Plus accepts eight letters and displays them as upper case.
- 10 Name the second product stream you created **PERMEATE**.
- 11 Name the block **MEMBRANE**.

**Note:** It matters which stream was created first because you will be passing Excel (and, in the next chapter, Fortran) data to an Aspen Plus array that contains product stream data. The first stream declared also comes first in the data array.

Your process flowsheet should look similar to Figure 1.1.

The next step is to enter a title, and feed stream characteristics.

## Entering Title, Components, Base Method, and Feed Specifications

- 1 Go to the **Setup - Specifications | Global** sheet and enter a title. For example, you might call it **Membrane1**.
- 2 Go to the **FEED (MATERIAL) - Input | Mixed** sheet.
- 3 Change the **Total flow basis** to **Volume**.
- 4 In the **Composition** area, change the type of units displayed to **Mass-Conc**.  
The **Solvent** field is now available.
- 5 Enter the temperature, pressure, total flow, solvent, and protein mass concentration as indicated below.

Specifications

Flash Type: **Temperature** **Pressure**

State variables

Temperature: **25** **C**

Pressure: **2.5** **bar**

Vapor fraction:

Total flow basis: **Volume**

Total flow rate: **17.5** **cum/hr**

Solvent: **WATER**

Composition

**Mass-Conc** **gm/l**

Component	Value
WATER	
PROTEIN	<b>3</b>

Total: **3**

- 6 Click the **Flash Options** tab to open the **FEED (MATERIAL) - Input | Flash Options** sheet.
- 7 In the **Valid phases** field, select **Liquid-Only**.

Next, specify the location of the Excel file that performs membrane calculations and enter the integer and real parameters that define the physical and calculational characteristics of the particular membrane being used for this simulation. The integer and real parameter user array will also hold results that do not correspond to standard Aspen Plus stream properties.

## Entering Excel Path and User Array Data

- 1 Go to the **MEMBRANE (User2) - Setup | Subroutines** sheet.
- 2 In the Excel file name area, enter the name of the future Excel file. For example, you might enter **MEMCALC.XLS**.

**Note:** This corresponds to an Excel file in the Aspen Plus working folder. If the file is located elsewhere, the full path must be specified (and is limited to 219 characters), but we recommend putting the Excel file in the working folder.

No other fields on this sheet need to be specified for this example. In the next chapter, a customized Fortran subroutine is used to interface with the Excel spreadsheet and the name of the subroutine is entered in the **Model** field in the User2 subroutines area. A customized Fortran subroutine allows you to control which data is passed between the Excel spreadsheet and Aspen Plus.

The screenshot shows the 'Subroutines' tab in the Aspen Plus interface. The 'User2 subroutines' section includes 'Model' and 'Report' text boxes. The 'Length of work arrays' section has 'Integer' and 'Real' spinners, both set to 1. The 'Excel file name' text box contains 'MEMCALC.XLS' and a 'Browse...' button. Below, the 'Additional user subroutines' section features a table with the header 'Subroutine name' and one empty row.

- 3 Click the **User Arrays** tab to open the **MEMBRANE (User2) - Setup | User Arrays** sheet.
- 4 In the **Number of parameters** area, enter **1** in the **Integer** field, **13** in the **Real** field, and **13** in the **Character** field.
- 5 Enter the parameters as indicated below. The character entries are optional here; they are just for convenience in this example. The last two real parameters are Excel-calculated results and will appear in the copy of the User Array table that is written to the **MEMBRANE (User2) - Results** sheet after the simulation run.

Subroutines User Arrays Configured Variables Calculation Options Stream

Number of parameters

Integer: 1 Real: 13 Character: 13

Values for parameters

	Integer	Real	Character
1	180	0.1	NUM TUBES AND DIAMETER CM
2		400	LENGTH CM
3		6e-07	DIFFUSIVITY CM2/S
4		450	GEL CONC G/L
5		0.9998	REJ COEFF
6		0.023	MASS XFER COEFF C1
7		0.8	C2
8		0.33	C3
9		0	C4
10		1	PERM PRES ATM
11		0.5	PRES DROP ATM
12			PROT PERM CONC (RESULT) G/L
13			PROT RETE CONC (RESULT) G/L

Next, make a product flash specification to cause Aspen Plus to evaluate the enthalpy and other product stream properties based on the temperature and pressure calculated by Excel.

## Setting up a Product Stream Flash

- 1 Click the **Stream Flash** tab to open the **MEMBRANE (User2) - Setup | Stream Flash** sheet.
- 2 In the **Stream** field, select **RETENTAT**.
- 3 In the **Flash type** field, select **Temperature & pressure**.
- 4 In the **Stream** field, select **PERMEATE**.
- 5 In the **Flash type** field, select **Temperature & pressure**.
- 6 Save your Aspen Plus file as **membrane1.apw**.

**Note:** It is recommended that you make a folder under the Aspen Plus working folder to use for these examples, and save your files in that folder. You will need to know the path to this folder in later chapters. For instance, if you create a folder named **membrane** under the default working folder on a typical installation, this path will be **C:\Documents and Settings\All Users\Application Data\AspenTech\Aspen Plus <version>\membrane**. To change the default working folder, click **File | Options**, then select **Files** from the pane on the left side of the **Options** dialog box, change the path of the **Default working directory** under **Locations**.



# Setting Up the Excel Model

An Excel template is available to expedite the specification process. For Excel 97 and later, use this template:

... \Engine\User\userxlTemplate.xls

Another template named userxlTemplate95.xls is available for use with Excel 95.

## Copying and Examining the Excel Template

- 1 Place one of the templates in the folder you specified on the Aspen Plus **MEMBRANE (User2) - Setup | Subroutines** sheet.
- 2 Open the template. When you open it, Excel may give you one of a few different warnings about macros. Excel 2007 or later may not even warn you macros are being disabled. This file contains macros used to communicate with Aspen Plus.
  - o If Excel offers you the option to **Enable Macros**, pick this option. In Excel 2007 or later, click the **Options** button in the security notice, select **Enable this content**, and click **OK**.
  - o If Excel says that macros are disabled, follow the directions given to lower your security setting in Excel to allow the macros to run. For example, in Excel 2003, go to **Tools | Options, Security** tab, click **Macro Security**, and set the security level to **Medium**. Then close the file and re-open it with the macros enabled.
  - o If using Excel 2007 or later and you do not get a warning about macros when opening this file, click the Office button in the upper left corner of the window, then click **Excel Options | Trust Center | Trust Center Settings** and set the option **For macros in documents not in a trusted location** to **Disable all macros with notification**. Then close the document and re-open it, and click **Options** in the security notice to enable the macros.
- 3 Save a copy as **MemCalc\_PreRun.xls** in your working folder.

**Note:** Later, you will save a copy of this file as **MemCalc.xls**. This second file will be changed by Aspen Plus and you may wish to compare the changed version to the original **MemCalc\_PreRun.xls**.
- 4 Use the tabs on the bottom of the screen to view the four named Excel sheets in the template that contain data. Model calculations will be performed on Sheet 1 (now empty).

	A	B	C	D	E	F	G	H	I	J
1	IntParams	1								
2		1	0							
3		2	0							
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										

The four data sheets and the data they will contain after the simulation is run are listed below:

- o **Aspen\_IntParams:** One integer parameter read from the Integer User Array (entered on the **MEMBRANE (User2) - Setup | User Arrays** Aspen Plus sheet).
- o **Aspen\_RealParams:** Eleven real parameters read from the Real User Array and two results (protein concentration in each product stream) calculated by Excel on Sheet 1.
- o **Aspen\_Input:** The mole flow rates (kmol/s) of each component of the feed stream and nine stream parameters.
- o **Aspen\_Output:** The mole flow rates (kmol/s) of each component of each product stream and the three product stream parameters that will be calculated by Excel. The other six product stream parameters are calculated by Aspen Plus.

**Note:** The flow rates and nine stream parameters are stored in Aspen Plus in two arrays: MSIN (input streams) and SOUT (output streams). Aspen Plus first transfers input data from MSIN to Excel and then transfers output results from Excel to SOUT. **All data read from MSIN or written to SOUT must be in SI units. Mole flow data must be in kmol/s.**

Next, modify the four Excel sheets to reflect the number of input and output streams and input and output parameters in this example. Define variables to reference data-containing cells. Use these variables in the model calculations performed on Sheet 1.

## Editing the Excel Sheets

- 1 Edit the **Aspen\_IntParams** sheet as shown below. Only the NTUBES entry in cell C2 is strictly necessary. The entries in the first two columns are dummies; Aspen Plus will fill in these cells when you run the simulation. Cell B2 will contain the integer parameter from the Aspen Plus **User Arrays** sheet.

	A	B	C	D	E	F
1	IntParams	labelB1	DEFINED AS			
2	labelA2	1	NTUBES			
3						
4						
5						
6						

- 2 Select **NTUBES** and click **Insert | Name | Define**. (In Excel 2007, right-click the cell and select **Name a Range**.)
- 3 Edit the **Refers to** field so that NTUBES represents the value in cell B2. In Excel 2003 this looks like:

The 'Define Name' dialog box in Excel 2003. The 'Names in workbook' list contains 'NTUBES'. The 'Refers to' field is set to '=Aspen\_IntParams!\$B\$2'. Buttons on the right include OK, Close, Add, and Delete.

In Excel 2007, it looks like:

The 'New Name' dialog box in Excel 2007. The 'Name' field is 'NTUBES', the 'Scope' is 'Workbook', and the 'Refers to' field is '=Aspen\_IntParams!\$B\$2'. Buttons at the bottom are OK and Cancel.

- 4 Edit the **Aspen\_RealParams** sheet as shown below. Again, the entries in the first and second column are dummies except for the equations in cells B13 and B14. Aspen Plus will fill in the first two columns but will not overwrite equations.

	A	B	C	D
1	RealParams	labelB1	DEFINED AS	units
2	labelA2	1	<b>DIAM</b>	cm
3	labelA3	1	<b>LEN</b>	cm
4	labelA4	1	<b>DIFF</b>	cm^2/s
5	labelA5	1	<b>GEL_CONC</b>	g/l
6	labelA6	1	<b>REJ_COEFF</b>	unitless
7	labelA7	1	<b>COEF1</b>	unitless
8	labelA8	1	<b>COEF2</b>	unitless
9	labelA9	1	<b>COEF3</b>	unitless
10	labelA10	1	<b>COEF4</b>	unitless
11	labelA11	1	<b>PRES_PERM</b>	atm
12	labelA12	1	<b>DELTA_P</b>	atm
13	labelA13	=CP	perm stream prot conc result from sheet 1	g/l
14	labelA14	=CR	ret stream prot conc result from sheet 1	g/l
15				
16				

**Note:** Since CP and CR have not been defined yet, Excel will not be able to determine values for cells B13 and B14. To display the equations (instead of an error message) click **Tools | Options | View** and select the **Formulas** check box. In Excel 2007, click the Office button at the far upper left, then click **Excel Options | Advanced** and select the **Show formulas in cells instead of their calculated results** check box.

- 5 Define the boldfaced variables (one at a time) in the third column to refer to the cells in the second column. For example, select **DIAM** and click **Insert | Name | Define** (right-click and select **Name a Range** on Excel 2007) and change **\$C\$2** to **\$B\$2** in the **Refers to** field.
- 6 Edit the Excel **Aspen\_Input** sheet as shown below. The Fortran interface subroutine reads the actual values of the feed stream properties from the Aspen Plus **MSIN** array and writes to column B the Excel sheet. The units column is optional.

	A	B	C	D	E
1	Input Stream Vector	FEED STREAM	units	defined as	
2	WATER		1 kmol/s	WATER_FEED	
3	PROTEIN		1 kmol/s	PROT_FEED	
4	TOTFLOW		1 kmol/s	TOT_FEED	
5	TEMP		1 K	TEMP_FEED	
6	PRES		1 N/m^2	PRES_FEED	
7	ENTHALPY		1 J/kg		
8	VAP FRAC		1 molar		
9	LIQ FRAC		1 molar		
10	ENTROPY		1 J/kg-K		
11	DENSITY		1 kg/m^3	DENS_FEED	
12	MOLE WT		1 kg/kmol	MW_FEED	
13					
14					
15					
16					

- 7 Redefine the variable **Aspen\_Input** to refer to the proper range of input variables. To do this, run the menu command **Insert | Name | Define (Formulas | Name Manager** in Excel 2007), then select the **Aspen\_Input** variable, and change the last letter and/or number in the definition. In this case **Aspen\_Input!\$B\$2:\$B\$12** is the correct definition.
- 8 Define the variables in the fourth column to refer to the cells in the second column.
- 9 Edit the **Aspen\_Output** sheet as shown below. Click **Tools | Options | View** and select **Formulas** to display the equations. (In Excel 2007, click the Office button at the far upper left | **Excel Options | Advanced** and select the **Show formulas in cells instead of their calculated results** check box.) The equations will not produce numbers yet because variables such as FP must be determined on **Sheet1**. The units column is optional.

	A	B	C	D
1	Output	RETENTATE STREAM	PERMEATE STREAM	units
2	WATER	=WATER_FEED - C2	=FP*RHO/3600/MWW	kmol/s
3	PROTEIN	=PROT_FEED - C3	=FP*CP/1000/3600/MWP	kmol/s
4	TOTFLOW	=TOT_FEED - C4	=C2+C3	kmol/s
5	TEMP	=TEMP_FEED	=TEMP_FEED	K
6	PRES	=PR	=PRES_PERM*101325	N/m^2
7	ENTHALPY	1	1	J/kg
8	VAP FRAC	1	1	molar
9	LIQ FRAC	1	1	molar
10	ENTROPY	1	1	J/kg-K
11	DENSITY	1	1	kg/m^3
12	MOLE WT	1	1	kg/kmol
13				
14				
15				
16				

The four sheets you just built hold data that the Fortran interface subroutine reads from Aspen Plus as well as results that are passed to Aspen Plus. The remaining Excel sheet, **Sheet1**, contains the unit operation model equations; it calculates the product stream properties using input from the first four sheets.

Create the model by entering equations and parameters on **Sheet1**.

**Note:** To facilitate construction of the Excel model, the template comes with a number of *aspen helper* functions built in. They are implemented as Visual Basic routines. To look at the code, select **Tools | Macro | Visual Basic Editor**; or in Excel 2007, select **View | Macros | View Macros**, type any name, and click **Create**. In the Visual Basic Editor, double-click the **AspenHooks** module. The **ahGetValue** function, for example, lets you refer to data using a label in the first column. Instead of using =DENS\_FEED/1000 in cell B9 on **Sheet1**, you could use =ahGetValue(Aspen\_Input, "DENSITY") / 1000. The first argument of this function is the Aspen\_Input variable mentioned previously and the second argument is a first-column string to search for within the specified range. The helper functions are described in the Aspen Plus *User Models* manual.

## Entering Membrane Model Equations and Parameters

- 1 Edit **Sheet1** as shown below. Turn on formulas. Define each variable in column A to refer to the cell to its right in column B. Variables that are used in other sheets are shown in boldface. Notes in column C are optional.



**Note:** Range names such as MT1 are no longer valid from Excel 2007. If you are using Excel files obtained from earlier versions of Aspen Plus examples (version V7.3 or earlier) and trying to run using Excel 2007 or later, you need to modify the range names of the terms MT1, MT2, MT3, MT4 to MT\_1, MT\_2, MT\_3, MT\_4, as well as update the formula for K as shown below.

	A	B	C	D
1	<i>Model Input</i>			
2	MWP	146.18944	kg/kmol prot mole wgt	
3	MWW	18.01528	kg/kmol water mole wgt	
4	MU	0.009	g/cm/s viscosity	
5	PMF	=PROT_FEED*MWP	prot mass flow feed stream	
6	TMF	=TOT_FEED*MW_FEED	total mass flow feed stream	
7	<i>Feed Stream</i>			
8	RHO	=DENS_FEED/1000	g/cm3 density	
9	FIN	=TOT_FEED*MW_FEED/RHO*3600	l/hr feed stream flow rate	
10	CIN	=PMF/TMF*RHO*1000	g/l feed stream prot conc	
11	PIN	=PRES_FEED/101325	atm feed stream pressure	
12	<i>Intermediate</i>			
13	UAVE	=FIN/(PI()/4*DIAM^2*NTUBES)*1000/3600	cm/s bulk average velocity	
14	RE	=DIAM*UAVE*RHO/MU	reynolds number (unitless)	
15	SC	=MU/RHO/DIFF	schmidt number (unitless)	
16	MT_1	=COEF1	Mult Term 1	
17	MT_2	=RE^COEF2	Mult Term 2	
18	MT_3	=SC^COEF3	Mult Term 3	
19	MT_4	=(DIAM/LEN)^COEF4	Mult Term 4	
20	<i>Results</i>			
21	CP	=GEL_CONC*(1-REJ_COEFF)	g/l perm stream prot conc	
22	K	=MT_1*MT_2*MT_3*MT_4*DIFF/DIAM/100	m/s mass xfer coefficient	
23	J	=K*LN((GEL_CONC-CP)/(CIN-CP))	m/s volumetric flux	
24	FP	=J*DIAM*LEN*PI()*NTUBES*100*3600/1000	l/hr perm stream flow rate	
25	PR	=(PIN-DELTA_P)*101325	N/m^2 ret stream pressure	
26	CR	=(CIN-FP/FIN*CP)/(1-FP/FIN)	g/l ret stream prot conc	
27				
28				
29				
30				

- Click **Tools | Options | View** and clear the **Formulas** check box. (In Excel 2007, click the Office button at the far upper left | **Excel Options | Advanced** and clear the **Show formulas in cells instead of their calculated results** check box.)
- Verify that your equations produce a number in each cell. Because there is no data yet, these numbers are meaningless.
- Turn off formulas in the **Aspen\_Output** and the **Aspen\_RealParams** sheets as well and check that there is a number in each cell.
- Optional:** Put sensible data in the **Aspen\_IntParams**, **Aspen\_RealParams**, and **Aspen\_Input** sheets.
- Once you are satisfied with the Excel ultrafiltration membrane model, save your **MemCalc\_PreRun.xls** file.
- Save a copy as **MemCalc.xls**. This file will be altered by Aspen Plus.

## Running the Excel Model

The Excel sheet can remain open when the simulation is run.

- 1 If necessary, open **MemCalc.xls** and then open **Membrane1.apw**.  
**Note:** Opening Excel before Aspen Plus saves startup time.
- 2 Go to the **MEMBRANE (User2) - Setup | Subroutines** sheet and verify that the path to the Excel file is correct.
- 3 Press **F5** to run the simulation.

## Examining Simulation Results

### Viewing the Results from the Excel Model

- 1 Examine the altered Excel file. Turn off formulas in each sheet and reformat numbers as needed using the Excel **Format | Cells** command (**Home | Format | Format Cells** in Excel 2007). The sheets are changed as follows:
  - o **Aspen\_IntParams:** Column A contains the row number because no labels are specified in the default interface routine. Cell B2 contains the number of tubes in the membrane model as specified on the Aspen Plus **User Arrays** sheet. Column C is not changed.
  - o **Aspen\_RealParams:** Column A contains the row number because no labels are specified in the default interface routine. Column B contains the eleven real parameters from the Aspen Plus **User Arrays** sheet that describe the membrane. Cells B12 and B13 contain equations and are therefore not directly changed by the interface routine.
  - o **Aspen\_Output:** Column A already has the correct labels – the interface routine overwrites with the same labels. Columns B and C are initialized to zero except for the cells containing equations which are not changed. The subroutine CalculateData invokes the Excel calculations and product stream results appear in cells B2:B6 and C2:C6. The interface routine reads the data in columns B and C and writes to the Aspen Plus SOUT array. Aspen Plus performs a product stream flash to replace the zeroes in SOUT with data so that, for example, enthalpy data is available to Aspen Plus even though it is not calculated by Excel.
  - o **Aspen\_Input:** Column A already has the correct labels – the interface routine overwrites with the same labels. Column B now contains the input parameters for the feed stream. These data are read from the Aspen Plus **MSIN** array.
  - o **Aspen\_Output\_MEMBRANE:** This is a new sheet created via a call to EndIteration which copies the contents of **Aspen\_Output** to this new sheet.
  - o **Sheet 1:** Using the feed stream data and real and integer parameters read from Aspen Plus arrays to the Excel sheets, the equations on **Sheet1** determine concentrations and flow rates for the two output streams. The interface routine does not read or write directly to or from this sheet.

For convenience, the data-filled Excel sheets and Aspen Plus results sheets are reproduced here.



### Aspen\_IntParams:

	A	B	C
1	INTPARAMS	1	DEFINED AS
2	1	180	NTUBES
3			
4			
5			

### Aspen\_RealParams:

	A	B	C	D	E
1	REALPARAMS	1	DEFINED AS	units	
2	1	0.1	DIAM	cm	
3	2	400	LEN	cm	
4	3	0.0000006	DIFF	cm^2/s	
5	4	450	GEL_CONC	g/l	
6	5	0.9998	REJ_COEFF	unitless	
7	6	0.023	COEF1	unitless	
8	7	0.8	COEF2	unitless	
9	8	0.33	COEF3	unitless	
10	9	0	COEF4	unitless	
11	10	1	PRES_PERM	atm	
12	11	0.5	DELTA_P	atm	
13	12	0.09	perm stream prot conc result from sheet 1	g/l	
14	13	4.615765004	ret stream prot conc result from sheet 1	g/l	

### Aspen\_Input:

	A	B	C	D
1	INPUT	FEED	units	defined as
2	WATER	0.267360092	kmol/s	WATER_FEED
3	PROTEIN	9.97564E-05	kmol/s	PROT_FEED
4	TOTFLOW	0.267459849	kmol/s	TOT_FEED
5	TEMP	298.15	K	TEMP_FEED
6	PRES	250000	N/m^2	PRES_FEED
7	ENTHALPY	-15821221.35	J/kg	
8	VAP FRAC	0	molar	
9	LIQ FRAC	1	molar	
10	ENTROPY	-9021.277647	J/kg-K	
11	DENSITY	993.8366156	kg/m^3	DENS_FEED
12	MOLE WT	18.06308603	kg/kmol	MW_FEED
13				
14				

### Aspen\_Output (and Aspen\_Output\_MEMBRANE):

	A	B	C	D
1	OUTPUT	RETENTAT	PERMEATE	units
2	WATER	0.171619584	0.095740508	kmol/s
3	PROTEIN	9.8688E-05	1.06844E-06	kmol/s
4	TOTFLOW	0.171718272	0.095741576	kmol/s
5	TEMP	298.15	298.15	K
6	PRES	199337.5	101325	N/m^2
7	ENTHALPY	0	0	J/kg
8	VAP FRAC	0	0	molar
9	LIQ FRAC	0	0	molar
10	ENTROPY	0	0	J/kg-K
11	DENSITY	0	0	kg/m^3
12	MOLE WT	0	0	kg/kmol

### Sheet1:

	A	B	C
1	Model Input		
2	MWP	146.18944	kg/kmol prot mole wgt
3	MWW	18.01528	kg/kmol water mole wgt
4	MU	0.009	g/cm/s viscosity
5	PMF	0.014583333	prot mass flow feed stream
6	TMF	4.831150254	total mass flow feed stream
7	Feed Stream		
8	RHO	0.993836616	g/cm3 density
9	FIN	17500.00014	l/hr feed stream flow rate
10	CIN	3	g/l feed stream prot conc
11	PIN	2.467308167	atm feed stream pressure
12	Intermediate		
13	UAVE	3438.532749	cm/s bulk average velocity
14	RE	37970.44166	reynolds number (unitless)
15	SC	15093.02411	schmidt number (unitless)
16	MT_1	0.023	Mult Term 1
17	MT_2	4608.467393	Mult Term 2
18	MT_3	23.93294637	Mult Term 3
19	MT_4	1	Mult Term 4
20	Results		
21	CP	0.09	g/l perm stream prot conc
22	K	0.000152206	m/s mass xfer coefficient
23	J	0.000767254	m/s volumetric flux
24	FP	6247.758725	l/hr perm stream flow rate
25	PR	199337.5	N/m^2 ret stream pressure
26	CR	4.615765004	g/l ret stream prot conc

## Viewing the Block Results

- 2 Go back to the Aspen Plus simulation environment.

- 3 Go to the **MEMBRANE (User2) - Stream Results | Material** sheet and the **MEMBRANE (User2) - Results | Summary** sheet to verify that Aspen Plus data and Excel data are the same. You find that the units are different.  
Aspen Plus did read the correct output values from Excel but it assumed (correctly) that they were in SI units and converted the data to METCBAR units before displaying it on the **Stream Results** sheet. For example, the component mole flow rates are reported in kmol/hr instead of kmol/sec.
- 4 On the **Home** tab of the ribbon, in the **Units** group, change the units to **SI**. Now the flow rates should be shown in kmol/s and match the ones from Excel.
- 5 Change the units back to **METCBAR**.
- 6 Go to the **MEMBRANE (User2) - Results | Summary** sheet to view the real and integer parameter data read from Excel. The last two real parameters displayed on this Aspen Plus sheet contain values. Aspen Plus read these values from the Excel **Aspen\_RealParams** sheet without regard to units.
- 7 Save the run. This Aspen Plus file, **Membrane1.apw** will be used in the next chapter.
- 8 Exit Aspen Plus.
- 9 Save the changed version of your Excel file, **MemCalc.xls**.
- 10 Exit Excel.

Aspen Plus **MEMBRANE (User2) - Stream Results | Material** sheet:

Material	Heat	Work	Vol. % Curves	Wt. % Curves	Petro. Curves	Poly. C
Display: Streams	Format: GEN_M	Stream Table				
	FEED	PERMEATE	RETENTAT			
Temperature C	25	25	25			
Pressure bar	2.5	1.013	1.993			
Vapor Frac	0	0	0			
Mole Flow kmol/hr	962.855	344.67	618.186			
Mass Flow kg/hr	17392.1	6209.81	11182.3			
Volume Flow cum/hr	17.5	6.248	11.252			
Enthalpy Gcal/hr	-65.724	-23.519	-42.205			
Mole Flow kmol/hr						
WATER	962.496	344.666	617.83			
PROTEIN	0.359	0.004	0.355			

Aspen Plus **MEMBRANE (User2) - Results | Summary** sheet:

Summary		Balance	Configured Variables	Status
User results				
		Integer	Real	Character
►	1	180	0.1	NUM TUBES AND DIAMETER CM
►	2		400	LENGTH CM
►	3		6e-07	DIFFUSIVITY CM2/S
►	4		450	GEL CONC G/L
►	5		0.9998	REJ COEFF
►	6		0.023	MASS XFER COEFF C1
►	7		0.8	C2
►	8		0.33	C3
►	9		0	C4
►	10		1	PERM PRES ATM
►	11		0.5	PRES DROP ATM
►	12		0.09	PROT PERM CONC (RESULT) G/L
►	13		4.61577	PROT RETE CONC (RESULT) G/L

## 2 Customizing the Excel/Aspen Plus Interface

Aspen Plus allows you to customize the Fortran interface subroutine that communicates with the Excel model. You need Intel Fortran 9.0 (or a later version) to compile your customized interface routine using the **aspcomp** command in Aspen Plus Simulation Engine Window. Compaq Visual Fortran 6.0 or later may also work.

The interface subroutine passes feed stream data from Aspen Plus to Excel and it passes product stream data from Excel to Aspen Plus. By customizing the subroutine, you can control which data is passed back and forth and can manipulate the data before or after it is processed in the Excel spreadsheet.

The Excel model in the previous chapter used Aspen Plus feed stream data in mole basis from the **Aspen\_Input** sheet, converted to mass basis in **Sheet1**, calculated the product stream flow rates, and then converted back to mole basis in **Aspen\_Output** so that the data could be transferred to Aspen Plus. This was necessary because the Aspen Plus **MSIN** and **SOUT** arrays expect flow rate data in mole basis.

In this session you will revise the interface routine to convert the input data to mass basis and the results data back to mole basis. The new interface routine will create custom labels for the **Aspen\_RealParams** Excel sheet. More importantly, it will obtain the viscosity of the feed stream by calling the appropriate Aspen Plus built-in subroutines. The new interface routine will then pass the viscosity to the Excel model so that you do not have to enter this parameter by hand (on **Sheet1**) as in the previous chapter.

Allow about 60 minutes for this session.

### Revising the Excel Model

Alter the Excel file from the previous chapter so that when the revised interface routine writes the feed stream viscosity to the Excel **Aspen\_Input** sheet, it is passed to **Sheet1**, converted to the appropriate units, and used in the model calculations.

Also, change **Aspen\_Input**, **Aspen\_Output**, and **Sheet1** so that they use mass basis.

- 1 Open **MemCalc\_PreRun.xls** and save a copy as **MemCalc\_Custom\_PreRun.xls** in your working folder.
- 2 Add **row 13** to the **Aspen\_Input** sheet as shown below. Change the units in **rows 2, 3, and 4** to indicate mass basis.

	A	B	C	D
1	Input Stream Vector	FEED STREAM	units	defined as
2	WATER		1 kg/s	<b>WATER_FEED</b>
3	PROTEIN		1 kg/s	<b>PROT_FEED</b>
4	TOTFLOW		1 kg/s	<b>TOT_FEED</b>
5	TEMP		1 K	<b>TEMP_FEED</b>
6	PRES		1 N/m^2	<b>PRES_FEED</b>
7	ENTHALPY		1 J/kg	
8	VAP FRAC		1 molar	
9	LIQ FRAC		1 molar	
10	ENTROPY		1 J/kg-K	
11	DENSITY		1 kg/m^3	<b>DENS_FEED</b>
12	MOLE WT		1 kg/kmol	<b>MW_FEED</b>
13	<b>ASPEN VISCOSITY</b>		<b>1 kg/m/s</b>	<b>MU_ASPEN</b>

- 3 Define **MU\_ASPEN** to be the viscosity that will be written to cell **B13** using the **Insert | Name | Define** (or right-click | **Name a Range** on Excel 2007) command.
- 4 Alter **Sheet1** to use the viscosity calculated by Aspen Plus. Convert the viscosity (Aspen Plus always uses SI units) to cgs units and change two equations in the Excel model to use mass basis. Two rows are no longer needed (**PMF** and **TMF**) and three cells (**B4**, **B7**, and **B8**) require minor changes. The three new equations are shown in boldface.



	A	B	C
1	Model Input		
2	MWP	146.18944	kg/kmol prot molec. wgt.
3	MWW	18.01528	kg/kmol water molec. wgt.
4	MU	=MU_ASPEN*10	g/cm/s viscosity
5	Feed Stream		
6	RHO	=DENS_FEED/1000	g/cm3 density
7	FIN	=TOT_FEED/DENS_FEED*3600*1000	l/hr feed stream flow rate
8	CIN	=PROT_FEED/FIN*3600*1000	g/l feed stream prot conc
9	PIN	=PRES_FEED/101325	atm feed stream pressure
10	Intermediate		
11	UAVE	=FIN/(PI()/4*DIAM^2*NTUBES)*1000/3600	cm/s bulk average velocity
12	RE	=DIAM*UAVE*RHO/4.4	reynolds number (unitless)
13	SC	=4.4/RHO/DIFF	schmidt number (unitless)
14	MT_1	=COEF1	Mult Term 1
15	MT_2	=RE^COEF2	Mult Term 2
16	MT_3	=SC^COEF3	Mult Term 3
17	MT_4	=(DIAM/LEN)^COEF4	Mult Term 4
18	Results		
19	CP	=GEL_CONC*(1 - REJ_COEF)	g/l perm stream prot conc
20	K	=MT_1*MT_2*MT_3*MT_4*DIFF/DIAM/100	m/s mass xfer coefficient
21	J	=K*LN((GEL_CONC - CP)/(CIN - CP))	m/s volumetric flux
22	FP	=J*PI()*DIAM*LEN*NTUBES*100*3600/1000	l/hr perm stream flow rate
23	PR	=(PIN - DELTA_P)*101325	N/m^2 ret stream pressure
24	CR	=(CIN - FP/FIN*CP)/(1 - FP/FIN)	g/l ret stream prot conc

- 5 Alter the Excel **Aspen\_Output** sheet to calculate component flow rates in mass basis. The new equations are shown in boldface. Change the units in **rows 2, 3, and 4** to indicate mass basis.

	A	B	C	D
1	Output Stream Vector	RETENTATE STREAM	PERMEATE STREAM	units
2	WATER	=WATER_FEED - C2	=FP*RHO/3600	kg/s
3	PROTEIN	=PROT_FEED - C3	=FP*CP/1000/3600	kg/s
4	TOTFLOW	=TOT_FEED - C4	=C2 + C3	kg/s
5	TEMP	=TEMP_FEED	=TEMP_FEED	K
6	PRES	=PR	=PRES_PERM * 101325	N/m^2
7	ENTHALPY	1	1	J/kg
8	VAP FRAC	1	1	molar
9	LIQ FRAC	1	1	molar
10	ENTROPY	1	1	J/kg-K
11	DENSITY	1	1	kg/m^3
12	MOLE WT	1	1	kg/kmol

- 6 Save your new Excel file, **MemCalc\_Custom\_PreRun.xls**.
- 7 Save a copy of the new Excel file as **MemCalc\_Custom.xls** so that you retain the original version after the run. Make sure this version is stored in the folder specified on the Aspen Plus **Subroutines** sheet.

# Revising the User Subroutine

The code for the default subroutine provides a template that will make creating the customized subroutine easier. The complete revised subroutine is listed below with changes marked.

- 1 Put a copy of \Engine\User\usrxls.f in a folder in your working folder. For example, copy this file to  
d:\Program Files\AspenTech\Working Folders\Aspen Plus\Membrane\Mem2.f.
- 2 Open **Mem2.f** and change the name of the subroutine to **Mem2**.
- 3 Edit the file as indicated in the code listing that follows. Some of the comments from the template (**usrxls.f**) have been shortened while others have been added. Numbers in parentheses following the boldface comments refer to the relevant part of the Code Description that follows the listing.

## Code Listing – Subroutine MEM2

```
C User2 Unit Operation Model Subroutine for Excel Models
C
C You can use any subroutine name up to six characters as long as
C you identify the name on the Blocks BLOCKNAME Setup Subroutines
C sheet in Aspen Plus. The argument list must not be lengthened or
C shortened.

      SUBROUTINE MEM2 (NMATI,  MSIN,  NINFI,  SINFI,  NMATO,
2          SOUT,  NINFO,  SINFO,  IDSMI,  IDSII,
3          IDSMO,  IDSIO,  NTOT,  NSUBS,  IDXSUB,
4          ITYPE,  NINT,  INTV,  NREAL,  REALV,
5          IDS,  NPO,  NBOPST,  NIWORK,  IWORK,
6          NWORK,  WORK,  NSIZE,  ESIZE,  INTSIZ,
7          LD  )

C
      IMPLICIT NONE

C
      Declare variables used in dimensioning first, then other variables.
C

      INTEGER NMATI, NINFI, NMATO, NINFO, NTOT,
+          NSUBS, NINT, NPO,  NIWORK, NWORK,
+          NSIZE, NREAL

      INTEGER IDSMI(2,NMATI), IDSII(2,NINFI), IDSMO(2,NMATO),
+          IDSIO(2,NINFO), IDXSUB(NSUBS),  ITYPE(NSUBS),
+          INTV(NINT),  IDS(2,3),  NBOPST(6,NPO),
+          IWORK(NIWORK),  INTSIZ(NSIZE),  LD

      REAL*8 MSIN(NTOT,NMATI), SINFI(NINFI), SOUT(NTOT,NMATO),
+          SINFO(NINFO),  WORK(NWORK),  ESIZE(NSIZE),
+          REALV(NREAL)
```



```

C
C----- Aspen Plus Common Definitions Start in Column 1. -----
C
#include "dms_errout.cmn"
#include "ppexec_user.cmn"
#include "dms_ncomp.cmn"
#include "dms_plex.cmn"
      REAL*8 B(1)
      EQUIVALENCE (B(1),IB(1))
C
C----- Declare Aspen Plus utility functions that will be used. -----
C
      INTEGER DMS_IRRCHK, DMS_IFCMNC
C
C----- Local Variable Declarations -----
C
      INTEGER OFFSET,      NUM_COLS,      NUM_ROWS,      RETCODE,      LID,
+      IBLANK(2), KREAL(3), KINT(3), KINPUT(2), KOUTPUT(2),
+      LDATA, I,J,K, LEN, KDIAG, IDX(10),
+      LIDSC, NCD, C_OFF, ERRNUMBER, NUM_LINES,
+      SOURCE(16), HOL_STRLEN, SOR_LENGTH, SSID(2), IPROG(2),
+      DESC_LENGTH,KOUTSOL(3), EXCEL_NAME_LEN,
+      EXCEL_NAME(64), KINPSOL(3),
+      DESCRIPTION(128)      !Moved LABELS and ROWNAMES declarations. (1)

      Integer LABELS(2,10), ROWNAMES(2,NCOMP_NCC+10)
                                          !Ten labels instead of nine. (1)
      Integer REALROWS(2,NTOT+1)          !Hollerith Realparam row ID's. (2)
      Integer LMW, NCP                     !Conversion and Viscosity. (3)
      Real*8 FLOW, XMW, X(10)               !Calc component mass flow. (4)
      Character*8 RealLabels(13)           !Realparam row ID's. (5)
      REAL*8 INSTREAM(NTOT*NMATI),OUTSTREAM(NTOT*NMATO)
C
C----- Data Section -----
C

      DATA KINPUT      /4HINPU, 4HT /
      DATA KOUTPUT     /4HOUTP, 4HUT /
      DATA KREAL       /4HREAL, 4HPARA, 4HMS /
      DATA KINT        /4HINTP, 4HARAM, 4HS /
      DATA KINPSOL     /4HINP_, 4H , 4H /
      DATA KOUTSOL     /4HOUT_, 4H , 4H /
      DATA IPROG       /4HUSRX, 4HLS /
      DATA IBLANK      /4HIMIS, 4HS /

      DATA LABELS /4HTOTF, 4HLOW ,
+      4HTEMP, 4H ,
+      4HPRES, 4H ,
+      4HENTH, 4HALPY,
+      4HVAP , 4HFRAC,
+      4HLIQ , 4HFRAC,
+      4HENTR, 4HOPY ,
+      4HDENS, 4HITY ,
+      4HMOLE, 4H WT ,
+      4HMUMX, 4H /      !MUMX is viscosity label. (6)

```

```

DATA RealLabels  /'x      ',      !Labels for Realparam table
+                'L      ',      !(each total eight chars). (7)
+                'Dab    ',
+                'Cg     ',
+                'R      ',
+                'C1     ',
+                'C2     ',
+                'C3     ',
+                'C4     ',
+                'Pperm  ',
+                'DeltaP ',
+                'Cp     ',
+                'Cr     ' /

C--- Establish Excel link and call StartIteration Workbook Hook Function ---
CALL USRUTL_GETEXCEL(EXCEL_NAME, EXCEL_NAME_LEN)
CALL StartIteration(RETCODE, EXCEL_NAME,256, IDS(1,1), 8)
IF(RETCODE .NE. 0) GOTO 1000

C----- Build Aspen_Input Data Table -----
C First get name of each of the NCOMP_NCC components for
C first NCOMP_NCC rows in column 1 of the table. Names are in
C two four letter blocks. Component names in column 1 label
C component flow rates in column 2.

      OFFSET = DMS_IFCMNC('IDSCC')
      DO J=1, NCOMP_NCC
        I = OFFSET + 2*(J-1) + 1
        ROWNAMES(1,J) = IB(I)
        ROWNAMES(2,J) = IB(I+1)
      END DO

C After the component names, list the nine standard stream properties plus
C any extras (in this case, there is one extra).

      DO K=1, 10
        J = K + NCOMP_NCC
        ROWNAMES(1,J) = LABELS(1,K)
        ROWNAMES(2,J) = LABELS(2,K)
      END DO
      NUM_ROWS = NCOMP_NCC + 10

C Now fill in column 2 (using the INSTREAM array) with Aspen Plus feed
C stream data (from the MSIN array). Convert to mass basis using molecular
C weights. Make a column for each feed stream. Also have Aspen Plus
C calculate viscosity of feed stream.

      LMW = DMS_IFCMNC('MW')  !Will need molecular weights of individual
                              !components to do flow rate conversions. (9)
      NUM_COLS = NMATI

      DO J=1, NUM_COLS
        OFFSET = (J-1)*(NCOMP_NCC+10)
        DO I=1, NUM_ROWS
          IF (I .EQ. NUM_ROWS) THEN
            Call SHS_CPACK(MSIN(1,J), NCP, IDX, X, FLOW) !"pack" data
          
```

```

        KDIAG = 4                                !then get
        K = NCOMP_NCC                            !viscosity. (11)
        Call PPMON_VISCL(MSIN(K+2,J), MSIN(K+3,J), X, NCP,
+                               IDX, NBOPST, KDIAG,
+                               INSTREAM(OFFSET+I), RETCODE)

    ELSE
        XMW = 1                                !Only convert flow rates.
        IF(I .LE. NCOMP_NCC) XMW = B(LMW + I) !If doing flow rates,
                                           !make XMW molecular wgt. (12)
        INSTREAM(OFFSET + I) = MSIN(I,J)*XMW
                                           !Convert and fill INSTREAM.
    END IF
    IF (I .EQ. NCOMP_NCC+1) INSTREAM(OFFSET+I) =
+       INSTREAM(OFFSET+I)*MSIN(I+8,J) !Convert total mole flow
        ! to total mass flow using molec wgt of stream. (13)
    END DO
END DO

C----- Send Aspen_Input table to Excel. -----
    LDATA = 8*NUM_ROWS*NUM_COLS
    CALL WRITETABLE(RETCODE , KINPUT, 8      , NUM_ROWS,
+       ROWNAMES, 8      , NUM_COLS, IDSMI  ,
+       8      , 2      , INSTREAM, LDATA  )
    IF (RETCODE .NE. 0) GO TO 1000

C----- Send Aspen_IntParams table to Excel. -----
    LDATA = 8*NINT
    CALL WRITETABLE(RETCODE, KINT, 12 , NINT ,
+       IBLANK , 0      , 1 , IBLANK,
+       0      , 1      , INTV, LDATA )
    IF (RETCODE .NE. 0) GO TO 1000

C----- Send Aspen_RealParams table to Excel. -----
    DO I = 1, 13                                !Make the 13 labels into Holleriths. (14)
        CALL USRUTL_CHARTOH(RealLabels(I), RealRows(1,I), 8)
    END DO

    LDATA = 8*NREAL
    CALL WRITETABLE(RETCODE, KREAL, 12 , NREAL , !Send Labels,
+       RealRows, 8      , 1 , IBLANK, !Length to Excel.
+       0      , 2      , REALV, LDATA )
    IF (RETCODE .NE. 0) GO TO 1000

C----- Initialize the Aspen_Output Excel table. -----
    NUM_ROWS = NCOMP_NCC + 9                    !No viscosity for output streams. (15)
    DO J=1, NMATO
        OFFSET = (J-1)*(NCOMP_NCC+9)          !Still 9 here. (15)
        DO I=1, NUM_ROWS
            OUTSTREAM(OFFSET+I) = 0.D0
        END DO
    END DO

    NUM_COLS = NMATO
    LDATA = 8*NUM_ROWS*NUM_COLS
    CALL WRITETABLE(RETCODE , KOUTPUT, 8      , NUM_ROWS,
+       ROWNAMES, 8      , NUM_COLS , IDSMO  ,
+       8      , 2      , OUTSTREAM, LDATA  )

```

```

      IF (RETCODE .NE. 0) GO TO 1000

C
C----- Solid Substream Section -----
C
C   When a simulation contains either conventional or nonconventional
C   solids, another Excel sheet will be created as the interface with
C   a separate sheet being created for each substream. The name of
C   substream will be appended onto the sheet name. For example, a
C   sheet called Aspen_INP_CISOLID will be created for the CISOLID
C   substream. The component names and stream properties for this
C   stream will be prepared similarly to Aspen_Input. A separate
C   column will be used for each feed. NSUBS will be greater than 1
C   if any solid substreams exist.

      IF (NSUBS .GT. 1) Then

C   Build the input tale for each solid substream
      DO K=2,NSUBS

          IF (ITYPE(K) .EQ. 3) THEN
C               Nonconventional solid
                  LIDSC = DMS_IFCMNC('IDSNCC')
                  NCD = NCOMP_NNCC
          ELSE
C               Conventional solid
                  LIDSC = DMS_IFCMNC('IDSCC')
                  NCD = NCOMP_NCC
          ENDIF

          DO J=1, NCD
              LID = LIDSC + 2*(J-1)
              ROWNAMES(1,J) = IB(LID+1)
              ROWNAMES(2,J) = IB(LID+2)
          END DO

C   Place stream property labels in column 1
          DO J=1, 9
              I=NCD+J
              ROWNAMES(1,I) = LABELS(1,J)
              ROWNAMES(2,I) = LABELS(2,J)
          END DO
          NUM_ROWS = NCD+9

C   Append the substream name onto the end of the input
C   and output Excel sheet name.
          CALL SHS_PID(LD, K, SSID)
          KINPSOL(2) = SSID(1)
          KINPSOL(3) = SSID(2)
          KOUTSOL(2) = SSID(1)
          KOUTSOL(3) = SSID(2)

C   Register the stream data
          NUM_COLS = NMATI
          DO J=1, NUM_COLS
              OFFSET = (J-1)*(NCD+9)
              DO I=1, NCD+9

```

```

                INSTREAM(OFFSET+I) = MSIN(IDXSUB(K)+I-1,J)
            END DO
        END DO
C
C  Send interface table to Excel
        LDATA = 8*NUM_ROWS*NUM_COLS
        CALL WriteTable(RETCODE , KINPSOL, 12      , NUM_ROWS,
+                      ROWNAMES, 8      , NUM_COLS, IDSMI   ,
+                      8      , 2      , INSTREAM, LDATA   )
        IF (RETCODE .NE. 0) GO TO 1000
C
C      Initialize the Excel output interface for the substream
        NUM_COLS = NMATO
        DO J=1, NUM_COLS
            OFFSET = (J-1)*(NCD+9)
            DO I=1, NCD+9
                OUTSTREAM(OFFSET+I) = 0.D0
            END DO
        END DO
C
C  Send interface table to Excel
C
        NUM_ROWS = NCD+9
        LDATA = 8*NUM_ROWS*NUM_COLS
        CALL WriteTable(RETCODE , KOUTSOL, 12      , NUM_ROWS,
+                      ROWNAMES, 8      , NUM_COLS , IDSMO   ,
+                      8      , 2      , OUTSTREAM, LDATA   )
        IF (RETCODE .NE. 0) GO TO 1000

        END DO

    End If

C----- Invoke Excel Workbook Calculations -----
        CALL CalculateData(RETCODE)
        IF (RETCODE .NE. 0) GOTO 1000

C----- Read Excel values into Aspen Plus Arrays -----

C----- Read back integer parameter table in case it has results.
        LDATA = 8*NINT
        CALL READTABLE(RETCODE, KINT, 12      , NINT ,
+                      1      , 1      , INTV, LDATA)
        IF (RETCODE .NE. 0) GOTO 1000

C ---- Read back real parameter table in case it has results.
        LDATA = 8*NREAL
        CALL READTABLE(RETCODE, KREAL, 12      , NREAL,
+                      1      , 2      , REALV, LDATA)
        IF (RETCODE .NE. 0) GOTO 1000

C
C  Read product stream data from Excel Aspen_Output sheet to OUTSTREAM.
C  Only have component flow rates, total flow rate, temperature and pressure.
C  First column of data (the retentate stream data) comes first in the
C  1 dimensional OUTSTREAM array.
C

```

```

      NUM_ROWS = NCOMP_NCC + 3      ! Change 9→3; Two flow rates + tot flow,
      NUM_COLS = NMATO              ! temp, pres. (16)
      LDATA = 8*NUM_ROWS*NUM_COLS
      CALL READTABLE(RETCODE , KOUTPUT, 8      , NUM_ROWS,
+                NUM_COLS, 2      , OUTSTREAM, LDATA )
      IF (RETCODE .NE. 0) GOTO 1000
C
C Put results from two product streams (OUTSTREAM array) in Aspen Plus
C product stream array (2 dimensional SOUT).
C The second dimension of SOUT is the number of streams; the first stream
C placed by the user on the Aspen Plus process flowsheet comes first in
C SOUT. Retentate stream was placed first and it must also be listed first
C in the Excel Aspen_Output sheet. SOUT expects flow rate data in mole basis
C so convert back to mole basis using molecular weights. This time
C get total mole flow by just adding up component mole flows.
C
      DO J = 1, NMATO
        FLOW = 0                      ! Holds the total mole flow.
        OFFSET = (J-1)*NUM_ROWS      ! NUM_ROWS, not NCOMP_NCC+9
        DO I = 1, NUM_ROWS
          IF(I .LE. NCOMP_NCC) THEN
            SOUT(I, J) = OUTSTREAM(OFFSET+I)/B(LMW+I) !Convert flow
              !rates to mole basis and fill Aspen Plus array. (17)
            FLOW = FLOW + SOUT(I,J) !Store total. (17)
          ELSE IF(I .EQ. NCOMP_NCC+1) THEN
            SOUT(I,J) = FLOW          !Total flow to SOUT (3rd row). (17)
          ELSE
            SOUT(I,J) = OUTSTREAM(OFFSET + I) !For other rows, just fill array.
          END IF
        END DO
      END DO
C-----
C For solid substreams, obtain product info from Excel interface sheets
C
      IF (NSUBS .GT. 1) Then
C
C Retrieve data for each substream
      DO K = 2, NSUBS
        IF (ITYPE(K) .EQ. 3) THEN
          NCD = NCOMP_NNCC
        ELSE
          NCD = NCOMP_NCC
        ENDIF

        CALL SHS_PID(LD, K, SSID)
        KOUTSOL(2) = SSID(1)
        KOUTSOL(3) = SSID(2)
C
C Read Excel interface product sheet for substream
      NUM_ROWS = NCD+9
      NUM_COLS = NMATO
      LDATA = 8*NUM_ROWS*NUM_COLS

      CALL ReadTable(RETCODE , KOUTSOL, 12      , NUM_ROWS,
+                NUM_COLS, 2      , OUTSTREAM, LDATA )
      IF(RETCODE .NE. 0) GO TO 1000

```

```

C
C Store data into outlet stream vector
      DO J=1, NMATO
        OFFSET= (J-1)*(NCD+9)
        DO I=1, NCD+9
          SOUT(IDXSUB(K)+I-1, J) = OUTSTREAM(OFFSET+I)
        END DO
      END DO
END DO

      End If
C----- End of Current Iteration ----- !No more changes.

      CALL EndIteration(RETCODE)
      RETURN

C----- Error Handling Section -----
1000 CONTINUE
C
C ALL ERRORS COME HERE TO REPORT ERRORS
C
C Call GetLastTableDataErr()
C
C HOL_STRLEN SETS THE PADDING FOR ERROR HANDLING STRING RETURNED
C FROM THE EXCEL INTERFACE.
C
      HOL_STRLEN = 64
      DESC_LENGTH= 512
      SOR_LENGTH = 64
      CALL GetLastTableDataErr(RETCODE,      ERRNUMBER,  DESCRIPTION,
+                                     DESC_LENGTH, HOL_STRLEN, SOURCE,
+                                     SOR_LENGTH)
C REGISTER SEVERE SIMULATION ERROR FROM USER2 WITH ENGINE
      USER_ICONVG = -3
      IF (DMS_IRRCHK(IPROG, 1, 4, 4, USER_IUMISS,
+               0, 0, 2) .NE. 0) THEN
C DETERMINE NUMBER OF LINES OF LENGTH HOL_STRLEN TO PRINT
C MAX IS 10 SO WE CAN USE 8 FOR DESCRIPTION OF ERROR
      NUM_LINES = ( (DESC_LENGTH+HOL_STRLEN-1) /HOL_STRLEN)
      IF (NUM_LINES .GT. 8 ) NUM_LINES = 8
C MOVE STRINGS TO ERROUT_IEROUT ARRAY OF STRINGS. WILL BE PRINTED
C TO HISTRY FILE BY ERRPRT(). FIRST WRITE "MS Excel" THEN SOURCE
C STRING RETURNED BY API AS ERROR HEADING.
      WRITE(ERROUT_IEROUT(1), 10)
      WRITE(ERROUT_IEROUT(2), 11)(SOURCE(I), I=1,(SOR_LENGTH+3)/4)
C WRITE ERROR MESSAGE RETURNED FROM EXCEL API TO THE HISTORY FILE
      DO 111 J=1,NUM_LINES
        C_OFF = (J-1) * HOL_STRLEN/4
        WRITE(ERROUT_IEROUT(J+2), 11)
+          ( DESCRIPTION(I+C_OFF), I=1,(HOL_STRLEN+3)/4 )
111      CONTINUE
C PRINT ERROR MESSAGES TO HISTORY FILE USING THE FIRST NUMLINES+2
C LINES OF THE ERROUT_IEROUT ARRAY OF STRINGS.
      CALL DMS_ERRPRT(NUM_LINES+2)
      ENDIF
C END ERROR REPORTING TO HISTORY FILE
2000 CONTINUE

```

```

C  Call EndIteration()
C  EndIteration IS CALLED EVEN WHEN A SEVERE ERROR OCCURS ABOVE.
    CALL EndIteration(RETCODE)
999  CONTINUE
C    FORMAT STATEMENTS
10   FORMAT('          MS EXCEL Interface Reports:')
11   FORMAT('          ',16A4)

END

```

## Code Description – Subroutine MEM2

- 1 The MUMX (viscosity) addition to the list of stream properties means that there will be 10 properties read from Aspen Plus instead of the usual 9 so LABELS and ROWNAMES must be redimensioned.
- 2 We want to add labels (stored in REALROWS) to the Aspen\_RealParams table.
- 3 LMW will hold the *offset* to the location of the molecular weight data for each component. NCP will hold the actual number of components present in the stream.
- 4 FLOW will hold the total mole flow for the output streams. XMW will hold the molecular weight data for a given component. X will contain the mole fraction of each component actually present in the stream and is sent to the Physical Property Monitor subroutine PPMON\_VISCL which calculates the viscosity of the stream.
- 5 RealLabels is a character array for the labels of the Aspen\_RealParams Excel sheet that will be converted to Hollerith format.
- 6 MUMX is stored in Hollerith format and will be written to the Aspen\_Input and Aspen\_Output Excel sheets along with the standard nine stream property labels.
- 7 Labels for the Aspen\_RealParam table are entered explicitly as a character array.
- 8 There are now 10 rows in the Aspen\_Input and Aspen\_Output tables in addition to the NCOMP\_NCC = 2 rows that correspond to the component flow rates so DO loops have to be lengthened.
- 9 DMS\_IFCMNC is an integer function that determines the location of the data such as molecular weight and critical temperature that corresponds to each component. The user subroutine includes the following statements:

```

#include "dms_plex.cmn"
      REAL*8 B(1)
      EQUIVALENCE (B(1),IB(1))

```

These statements allow the subroutine to access component data. The character strings to feed to the DMS\_IFCMN function are listed in the **Universal Constant Names and Definitions** table in Chapter 6 of the *User Models* manual. The molecular weight of the first component will be found at B(LMW + 1) where LMW is the offset returned by the function.



- 10 The subroutine SHS\_CPACK *packs* the stream array by determining which components are actually present (have a non-zero flow rate) and putting their mole fractions in the X array. The PPMON\_VISCL subroutine needs a packed stream array when it calculates stream viscosity.
- 11 The subroutine PPMON\_VISCL returns the viscosity of the stream. In the subroutine in this example, the viscosity is placed in the INSTREAM array.
- 12 The **B** array is used to get the molecular weight which is used to convert component flow rates. The integer variable NCOMP\_NCC is equal to the number of conventional components (in this case 2) declared in the **Components Specifications Selection** sheet and is passed through the common DMS\_NCOMP also included in this subroutine.
- 13 The total mole flow is the third (NCOMP\_NCC + 1) entry in the Aspen Plus stream input array called MSIN. The first two (NCOMP\_NCC) are the component mole flows. The next eight stream properties are temperature, pressure, enthalpy, vapor fraction, liquid fraction, entropy, density, and molecular weight. These are listed in Appendix C of the *User Models* manual and these are the data that appear in the Aspen\_Input and Aspen\_Output Excel sheets.
- 14 The subroutine USRUTS\_CHARTOH converts the explicitly entered character labels into Hollerith format so that they may be sent to the WRITETABLE subroutine.
- 15 Only the viscosity of the feed stream is calculated so the output table still has the usual (NCOMP\_NCC + 9) rows.
- 16 Since the Excel model only calculates flow rates, temperature, and pressure, there is no point sending it a bunch of zeroes for the other properties. The properties that Excel does calculate are placed in OUTSTREAM when READTABLE is called.
- 17 The array SOUT contains the same NCOMP\_NCC component mole flows, total flow, and 8 other properties as MSIN (see item 13 above). When the SOUT array is filled with the OUTSTREAM values, these data will be available in Aspen Plus results sheets. The MSIN and SOUT arrays contain component and total flow rates in mole basis (kmol/s) and eight other properties in standard SI units.

## Compiling and Linking the Subroutine

The aspcomp command delivered in the Simulation Engine DOS prompt window will create the .OBJ file and ensure consistent compiler options. Two DLOPT (dynamic linking option) files control the linking process: one creates a DLL from specified .OBJ files and another tells Aspen Plus where to find the DLL being used for a particular run.

### Compiling the Excel Interface Subroutine

- 1 Save your customized Fortran file as **Mem2.f**.
- 2 If you have not already done so, set your compiler option by running **Start | Programs | AspenTech | Process Modeling <version> |**

**Aspen Plus | Select Compiler for Aspen Plus.** Choose the option which represents the combination of Fortran compiler and linker you want to use.

- 3 Launch the Aspen Plus Simulation Engine Window. From the **Start** menu, click **Programs | AspenTech | Process Modeling <version> | Aspen Plus | Aspen Plus Simulation Engine**.

A DOS window appears with the working directory as the default directory.

- 4 Use the DOS **cd** command to set the default directory to the location of **Mem2.f**. For example, type **cd membrane** if your Fortran file is in a subfolder called **membrane** in your working directory.

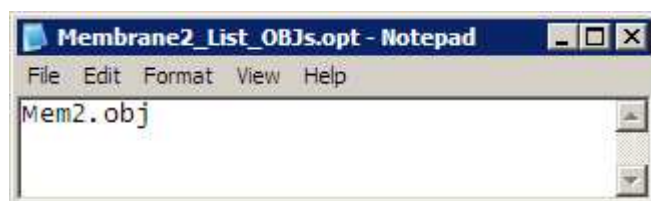
- 5 Type **aspcomp Mem2** to compile the subroutine.

An object file (**Mem2.obj**) is created in the same directory as **Mem2.f**.

Keep the DOS window available for the linking step and use a text editor to create a DLOPT file to control the creation of a shared library (also known as a dynamic linking library).

## Creating a Shared Library

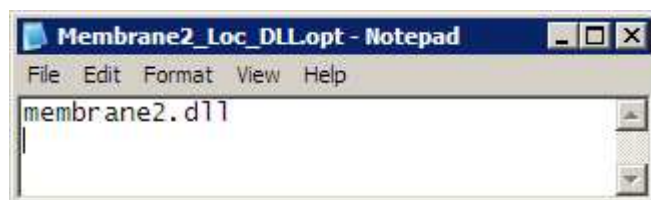
- 1 Use a text editor such as Notepad to create a text file called **Membrane2\_List\_OBjs.opt** in your working folder.
- 2 Type **Mem2.obj** in the top line of the text file.



- 3 Save the file and exit the text editor.
- 4 In the DOS window of the Aspen Plus Simulation Engine, type **asplink dlopt membrane2\_list\_objs.opt membrane2**.  
A file called **membrane2.dll** is created. This is a Fortran shared library. Using a shared library avoids the need for a linking step when Aspen Plus runs. Once you have the shared library, it can be used with Aspen Plus even if you do not have a Fortran compiler available.

**Note:** If you simply type **asplink membrane2**, Aspen Plus will create **membrane2.dll** using all the object files in the default directory.

- 5 Create another DLOPT text file called **Membrane2\_Loc\_DLL.opt**.
- 6 In the top line of this file type **membrane2.dll**.



- 7 Save the text file and exit the text editor.

- 8 Put a copy of the **Membrane2\_Loc\_DLL.opt** in your Aspen Plus Working Directory. At run time, you will tell Aspen Plus the name of this file so that the shared library will be available.

**Note:** If you use the pre-supplied DLOPT files, be sure to check that the path specified matches your working folder and update it (or remove the path).

## Running the Simulation

Now that you have your revised Excel file and have created a shared library containing your customized Fortran subroutine, you can run the simulation using **membrane1.apw** from the previous chapter.

- 1 Open **membrane1.apw**.
- 2 Click **H** (or press **Shift+F5**) and click **OK** twice to clear data from the previous run.

**Note:** If you have both the GUI and the Simulation Engine Window open at the same time, an **asplink** command will fail unless you reinitialize the open run.

- 3 Go to the **Setup - Specifications | Global** sheet, change the title to **membrane2**.
- 4 On the **Home** tab of the ribbon, in the **Units** group, make sure the units are set to **METCBAR**.
- 5 Go to the **MEMBRANE (User2) - Setup | Subroutines** sheet and type **MEM2** in the **Model** field in the User2 subroutines area so that Aspen Plus knows what your new subroutine is called.
- 6 On the **Subroutines** sheet, change the name of the Excel file to **MemCalc\_Custom.xls**.

The screenshot shows the 'Subroutines' tab in the Aspen Plus interface. The 'User2 subroutines' section has 'Model' set to 'MEM2' and 'Report' is empty. The 'Length of work arrays' section has 'Integer' and 'Real' both set to '1'. The 'Excel file name' field is 'MEMCALC\_CUSTOM.XLS' with a 'Browse...' button. The 'Additional user subroutines' table is empty.

User2 subroutines	
Model:	MEM2
Report:	

Length of work arrays	
Integer:	1
Real:	1

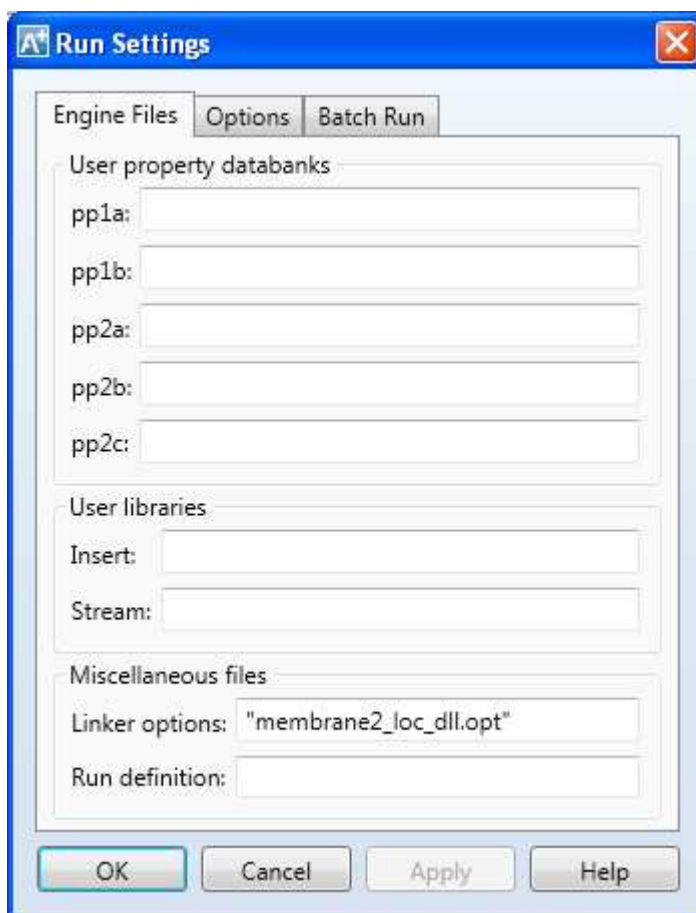
Excel file name: MEMCALC\_CUSTOM.XLS [Browse...]

Additional user subroutines	
	Subroutine name

- 7 From the ribbon, click the arrow at the bottom right corner of the **Run** group.



The **Run Settings** dialog box appears.



- 8 In the **Linker options** field in the **Miscellaneous files** area, type **membrane2\_loc\_dll.opt** so that Aspen Plus knows the name of the file containing the location of the library that contains your customized subroutine. This DLOPT file must be in your Aspen Plus Working Folder.

**Note:** You can change the default Aspen Plus working directory in the **Aspen Plus Options | Files | Locations** area. The **membrane2\_loc\_dll.opt** file must be saved in the same directory as the simulation file.

- 9 Click **OK**.
- 10 Run the simulation.
- 11 Compare your Aspen Plus and Excel Results to those displayed below.

**12** Save the run as **membrane2.apw** and exit Aspen Plus.

## Results from the Custom Excel Model

For convenience Excel and Aspen Plus sheets are shown here.

The Excel **Aspen\_Input** sheet appears below. It now contains viscosity data as calculated by Aspen Plus. The original label **Aspen Viscosity** was overwritten with the new label **MUMX** specified in the interface routine.

	A	B	C	D
1	INPUT	FEED	units	defined as
2	WATER	4.816566882	kg/s	WATER_FEED
3	PROTEIN	0.014583333	kg/s	PROT_FEED
4	TOTFLOW	4.831150215	kg/s	TOT_FEED
5	TEMP	298.15	K	TEMP_FEED
6	PRES	250000	N/m^2	PRES_FEED
7	ENTHALPY	-15821221.35	J/kg	
8	VAP FRAC	0	molar	
9	LIQ FRAC	1	molar	
10	ENTROPY	-9021.277647	J/kg-K	
11	DENSITY	993.8366156	kg/m^3	DENS_FEED
12	MOLE WT	18.06308603	kg/kmol	MW_FEED
13	MUMX	0.000913878	kg/m/s	MU_ASPEN

The Excel **Aspen\_RealParams** sheet appears below. It now contains labels in column A specified in the interface routine.

	A	B	C	D
1	REALPARAMS	1	DEFINED AS	units
2	x	0.1	DIAM	cm
3	L	400	LEN	cm
4	Dab	0.0000006	DIFF	cm^2/s
5	Cg	450	GEL_CONC	g/l
6	R	0.9998	REJ_COEF	unitless
7	C1	0.023	COEF1	unitless
8	C2	0.8	COEF2	unitless
9	C3	0.33	COEF3	unitless
10	C4	0	COEF4	unitless
11	Pperm	1	PRES_PERM	atm
12	DeltaP	0.5	DELTA_P	atm
13	Cp	0.09	perm stream prot conc result from sheet1	g/l
14	Cr	4.60	ret stream prot conc result from sheet1	g/l

The value of **Cr** in cell **B14** is slightly different from that in the previous chapter because the viscosity used in this example is more accurate.

The Excel **Aspen\_Output** sheet appears below. It contains flow rates in mass basis.

	A	B	C	D
1	OUTPUT	RETENTAT	PERMEATE	units
2	WATER	3.104	1.712	kg/s
3	PROTEIN	0.014	0.00016	kg/s
4	TOTFLOW	3.119	1.713	kg/s
5	TEMP	298.15	298.15	K
6	PRES	199337.5	101325	N/m^2
7	ENTHALPY	0	0	J/kg
8	VAP FRAC	0	0	molar
9	LIQ FRAC	0	0	molar
10	ENTROPY	0	0	J/kg-K
11	DENSITY	0	0	kg/m^3
12	MOLE WT	0	0	kg/kmol

The Aspen Plus **MEMBRANE (User2) - Stream Results | Material** sheet:

Material	Heat	Work	Vol.% Curves	Wt. % Curves	Petro. Curves	Poly. C
Display: Streams Format: GEN_M Stream Table						
	FEED	RETENTAT	PERMEATE			
Temperature C	25	25	25			
Pressure bar	2.5	1.993	1.013			
Vapor Frac	0	0	0			
Mole Flow kmol/hr	962.855	620.656	342.2			
Mass Flow kg/hr	17392.1	11226.8	6165.31			
Volume Flow cum/hr	17.5	11.297	6.203			
Enthalpy Gcal/hr	-65.724	-42.374	-23.35			
Mole Flow kmol/hr						
WATER	962.496	620.301	342.196			
PROTEIN	0.359	0.355	0.004			

The Aspen Plus **MEMBRANE (User2) - Results | Summary** sheet:

Summary		Balance	Configured Variables	Status
User results				
		Integer	Real	Character
▶ 1		180	0.1	DIAM CM
▶ 2			400	LEN CM
▶ 3			6e-07	CM2/S DIFF
▶ 4			450	G/L GEL CONC
▶ 5			0.9998	G/L REJ COEF
▶ 6			0.023	C1
▶ 7			0.8	C2
▶ 8			0.33	C3
▶ 9			0	C4
▶ 10			1	PERM PRES
▶ 11			0.5	DELTA P
▶ 12			0.09	PROT PERM CONC (RESULT)
▶ 13			4.59783	PROT RET CONC (RESULT)





# 3 Customizing the Aspen Plus Model Library

Your customized Excel model can have its own unit operation model stored in an Aspen Plus Library file (.apm extension) and displayed in the Model Palette. In the previous chapter you entered a set of real and integer parameters for the physical characteristics and calculational coefficients of the model using the Aspen Plus **User Arrays** data sheet. With a customized unit operation model, you can enter these parameters once and they will be automatically included whenever the customized block is placed on the Flowsheet.

In addition, the name of the user Fortran subroutine and Excel file can be associated with the customized model. The icon itself can be custom drawn.

Finally, a **Configured Variables** sheet is available in which the set of real and integer parameters are associated with character strings and identified as input or output. This data can be accessed in the user Fortran subroutine by referring to the variable names, thereby simplifying the code.

In this session, create a new Aspen Plus Library, **Ultrafiltration.apm**, containing a customized model for the ultrafiltration membrane. Use the same model parameters as in the previous chapter to specify the membrane characteristics. Create a custom icon for your model; then replace the User 2 block in **membrane2.apw** with your custom block and run the simulation again.

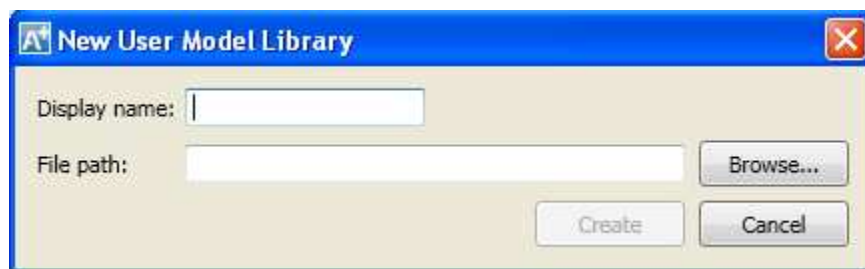
Allow about 30 minutes for this session.

## Creating a Model Library

A model library has three levels: the **library** itself consists of a set of **categories** each identified by a tab in the Model Palette. Each category consists of a set of **models**. Each model is represented by an icon (multiple icons may be created and used for strictly aesthetic reasons) and has default parameters associated with it. The present hollow fiber ultrafiltration membrane has 180 tubes but another might have only 90. They could be represented as separate models.

## Creating an Empty Custom Model Library

- 1 Create a sub-folder in your Aspen Plus Working Folder called, for example, **lib**, to contain library files.
- 2 Open an Aspen Plus blank simulation. Click the **Simulation** bar on the Navigation Pane to enter the Simulation environment.
- 3 Click the **Customize** tab of the ribbon. In **Model Library**, click **Manage Libraries**.  
The **Manage Libraries** dialog box appears.
- 4 Click **New**.  
The **New User Model Library** dialog box appears.



- 5 In the **Display name** field, type **Ultrafiltration**.
- 6 Click **Browse**. The **Save New User Model Library As** dialog box appears. Navigate to the directory where you wish to store library files. Click **Save**.
- 7 Click **Create**.  
You have created an empty library. It contains no categories and no models. The next step is to create a template that contains default parameters for your customized model.
- 8 Close the **Manage Libraries** dialog box.

## Creating a Template for Your Custom Model

- 1 From the Model Palette, place a **User Models | User2 | FILTER** block on the process flowsheet. It gets the default ID **B1**.
- 2 From the Navigation Pane, go to the **Blocks | B1 | Setup** form.
- 3 In the **Model** field, type the name of the Fortran subroutine that will be used with the customized block. For this example, type **MEM2**.
- 4 In the Excel file name area, type the name of the Excel file that contains the equations for the model. For example, **MemCalc\_Custom.xls**.
- 5 Click the **User Arrays** tab and enter the same real and integer parameters as in the previous chapter including two empty real parameters for results data.

Subroutines User Arrays Configured Variables Calculation Options Stream Flash Streams Info

Number of parameters  
Integer: 1 Real: 13 Character: 1

Values for parameters

	Integer	Real	Character
1	180	0.1	
2		400	
3		6e-07	
4		450	
5		0.9998	
6		0.023	
7		0.8	
8		0.33	
9		0	
10		1	
11		0.5	
12			
13			

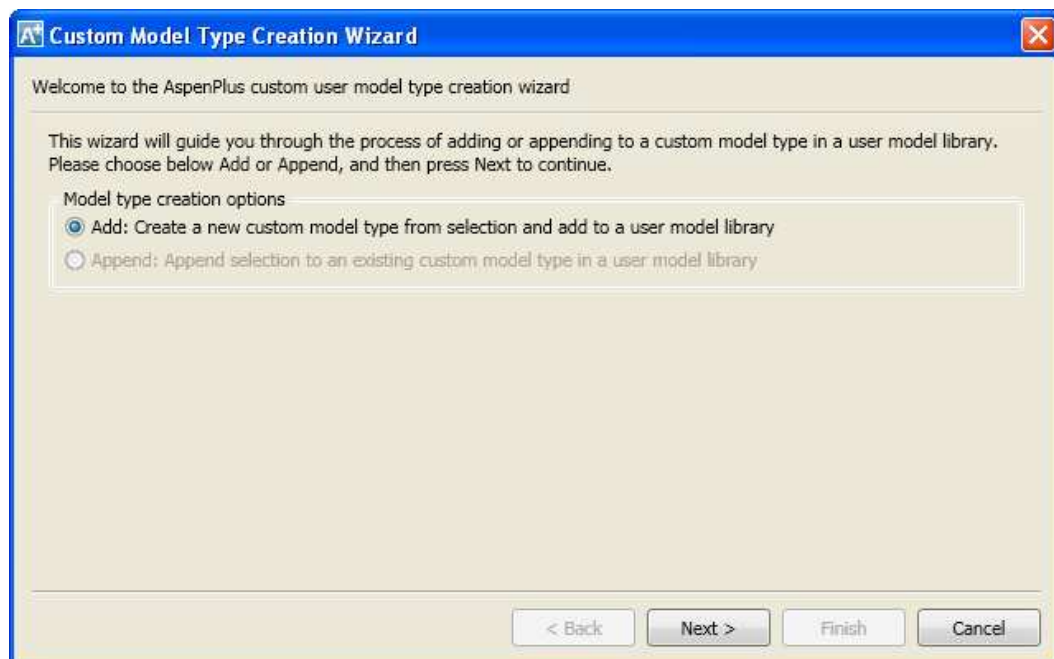
Now your template is prepared and you are ready to create the customized model.

## Creating the Custom Model from the Template

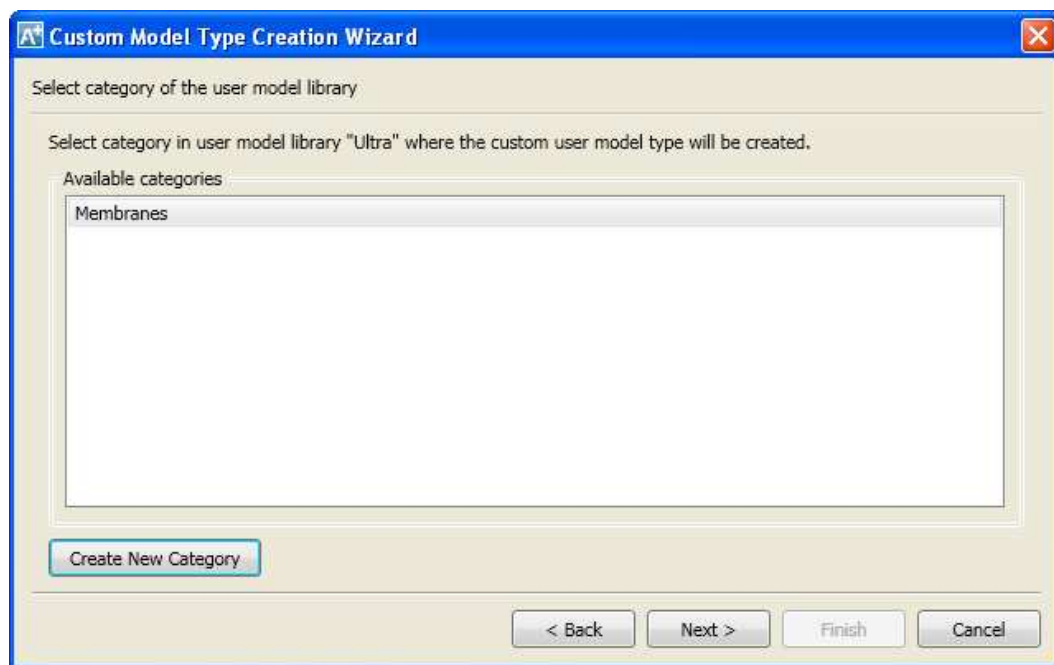
- 1 Go back to **Main Flowsheet**.
- 2 Select your template block B1, right click, and select **Add to Model Library**.

**Note:** If you start Aspen Plus in the future and find the **Add to Model Library** option unavailable, click **Manage Libraries** on the **Customize** tab of the ribbon, and select your library (**Ultrafiltration**).

The **Custom Model Type Creation Wizard** dialog box appears.



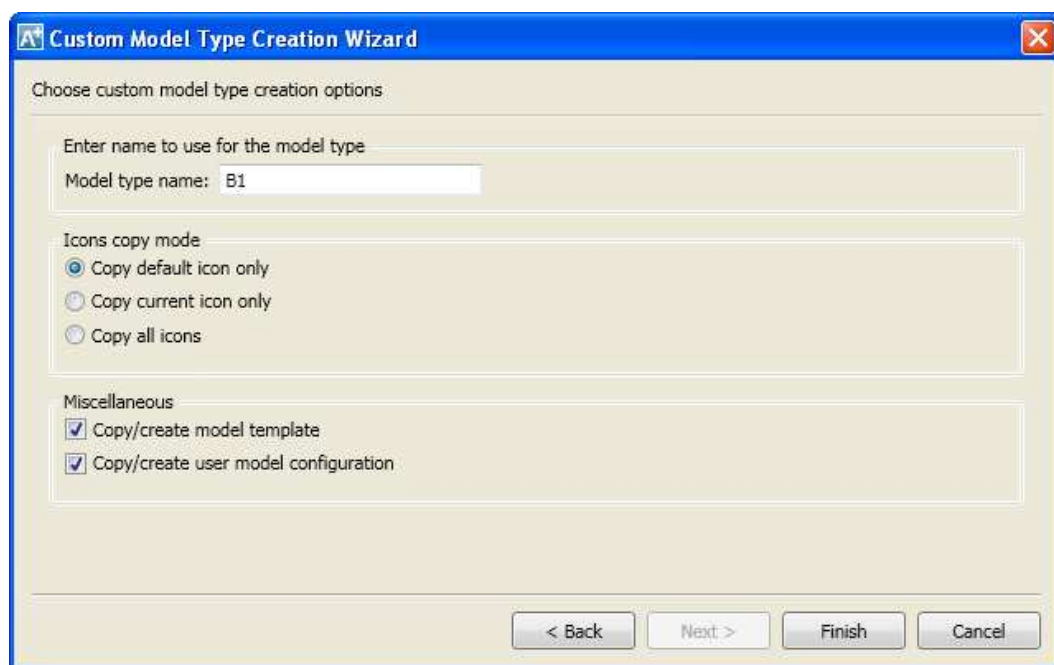
- 3 Select **Add: Create a new ...**, and click **Next**.
- 4 Click **Next** again for step **Select user model library**.
- 5 In the **Select category of the user model library** step, click **Create New Category**.
- 6 Type **Membranes** and click **OK**.  
The **Membranes** category appears.



- 7 Select **Membranes** and click **Next**.  
The **Choose single block or multi record custom model type** step appears.

- 8 Select **Single block custom user model** and click **Next**.

The **Choose custom model type creation options** step appears with the block ID **B1** entered automatically.



- 9 Replace **B1** with **HollowF**.

- 10 Choose any icon option. Later, you will delete the icon and create a custom icon.

- 11 Make sure the **Copy/create model template** check box is selected. Aspen Plus will copy the data you entered on the **Subroutines** sheet and on the **User Arrays** sheet into the new model.

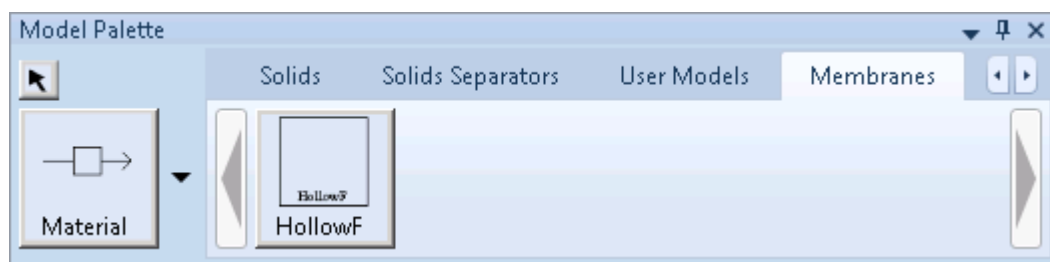
- 12 Make sure the **Copy/create user model configuration** check box is selected. Aspen Plus will make the **Configured Variables** sheet available to the new model so that you can associate variable names with your real and integer parameters.

- 13 Click **Finish**.

A tab for the **Membranes** category of the **Ultrafiltration** library now appears on the Model Palette alongside the tabs of the **Built-in** library.

- 14 In the Model Library click the **Membranes** tab.

The model you created is visible.



- 15 Delete the block on the Flowsheet that you used as a template.

- 16 From the Model Palette, drag a **HollowF** unit operation model onto the Flowsheet.
- 17 From the Navigation Pane, go to the **Blocks | B2 | Setup** form and check that the default data appears correctly on the **Subroutines** sheet and on the **User Arrays** sheet.
- 18 Exit Aspen Plus. There is no need to save an apw file since the library is already saved as **Ultrafiltration.apw**.  
Now, whenever you start Aspen Plus, click **Manage Libraries** on the **Customize** tab of the ribbon, and then select the **Ultrafiltration** check box, you will have access to the **HollowF** unit operation model with default user subroutine name, Excel file path, and real and integer parameters.

**Note:** Any Aspen Plus model from the Model Palette can be moved into the new library by dragging the icon into the Model Library Editor (click **Manage Libraries | Ultrafiltration | Edit** to open the Editor). The **Custom Model Type Creation Wizard** appears automatically.

## Editing the Custom Model

When you use a model from a custom library, you can overwrite the default entries to the **BLOCKNAME (Block Model Type) - Setup | Subroutines** and **User Array** sheets. For example, in the next chapter, you will place a **HollowF** block, delete the path to the Excel file and use a revised Fortran subroutine to perform the model calculations. This change only affects the block to which it is applied. The defaults for the **HollowF** model remain intact.

The **HollowF** model can be edited, however, in the following ways: you can delete it, rename it, redraw the icon, and add named variables that refer to the real and integer parameters on the **User Arrays** sheet. These changes are stored in the library.

Open the simulation from the previous chapter and edit the **HollowF** model. First customize the icon, then add named references to the real and integer parameters.

## Creating a Custom Icon

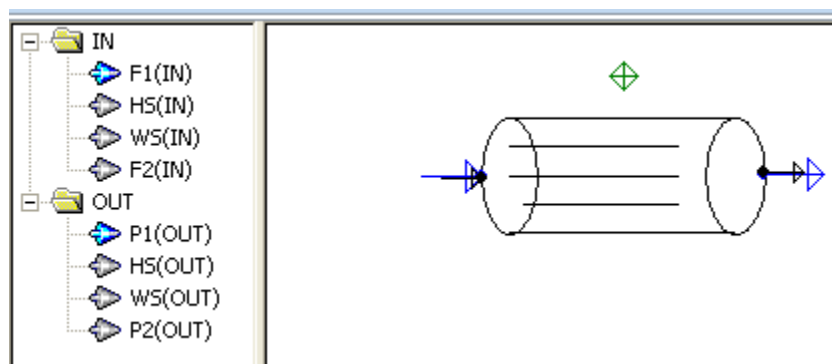
- 1 Open **Membrane2.apw**. Click the **Simulation** bar on the Navigation Pane to enter the Simulation environment.
- 2 On the **Customize** tab of the ribbon, click **Manage Libraries** and select the check box next to **Ultrafiltration** to make your library available.
- 3 Click **Edit** beside Ultrafiltration.
- 4 In the **Ultrafiltration – Model Library Editor** window, select the **HollowF** model, right click, and select **Edit Icon**.

The **BLOCK - Icon Editor** appears in the main window. A new tab, **Icon Tools** appears on the ribbon.

The small enclosed crosshair indicates the automatic position of the block ID. The **call Label.at(0,0.1)** statement in the lower pane indicates that it

will be placed very close to the spot where you click on the process flowsheet. (If necessary, drag the divider between the pane with the icon and the pane below it upward and scroll to find this statement.) If the coordinates are far from zero, block placement could be unpredictable. Keep your custom drawing reasonably well centered.

- 5 Click and hold and drag across the entire icon to enclose it in a dashed box. Click the right mouse button and select **Delete**.  
The icon disappears except for two arrows that represent the universal input and output ports.
- 6 Use the graphics tools to create a custom icon for the hollow fiber ultrafiltration membrane.
- 7 Move the universal port arrows so they are attached to the icon.
- 8 Drag an F1(IN) port from the left-hand pane and place it directly over the universal input arrow.
- 9 Drag a P1(OUT) port from the left-hand pane and place it directly over the universal output arrow.
- 10 Select **Save Icon** to update the code in the lower pane.



- 11 Exit the **Icon Editor** by closing the window.

**Note:** Existing AutoCAD DXF files can also be imported into the Icon Editor. When the Icon Editor is active, right click in the Icon Editor window and select **Import DXF** from the shortcut menu.

Next, add named references to the real and integer parameter arrays. These names will be used in the Fortran code in the next chapter. For the simulation in this chapter, the two parameters designated as output data will appear in a new results sheet. Otherwise this run will be identical to the run in the previous chapter.

## Creating References to the Real and Integer Parameters

- 1 In the **Manage Libraries** dialog box, click **Ultrafiltration | Edit**, right click the icon HollowF, and select **Edit User Configuration**.  
The **HollowF - User Model Configuration Editor** dialog box appears.
- 2 Create names for the 1 integer and 13 real parameters in the order that appear on the **MEMBRANE (User2) - Setup | User Arrays** sheet.

**Note:** Aspen Plus will correctly identify the integer no matter where you put it.

- 3 **Real** is the default type. For **NTUBES**, click in the **Type** field and select **Integer** from the list.
- 4 Parameters are designated as **Input** by default. For the last two real parameters, click in the **Input/Output** field and select **Output only**.

**Note:** To delete a row, select it, right click, select **Delete Row**.

The completed table is shown below.

HollowF - User Model Configuration Editor

User Configuration Design

	Variable Name	Type	Dimension	Depth	Input/Output
<input checked="" type="checkbox"/>	NTUBES	Integer	Scalar	0	Input Only
<input type="checkbox"/>	DIAM	Real	Scalar	0	Input Only
<input type="checkbox"/>	LEN	Real	Scalar	0	Input Only
<input type="checkbox"/>	DIFF	Real	Scalar	0	Input Only
<input type="checkbox"/>	GEL_CONC	Real	Scalar	0	Input Only
<input type="checkbox"/>	REJ_COEF	Real	Scalar	0	Input Only
<input type="checkbox"/>	COEF1	Real	Scalar	0	Input Only
<input type="checkbox"/>	COEF2	Real	Scalar	0	Input Only
<input type="checkbox"/>	COEF3	Real	Scalar	0	Input Only
<input type="checkbox"/>	COEF4	Real	Scalar	0	Input Only
<input type="checkbox"/>	PERM_PRES	Real	Scalar	0	Input Only
<input type="checkbox"/>	DELTA_P	Real	Scalar	0	Input Only
<input type="checkbox"/>	CP_PROT	Real	Scalar	0	Output Only
<input type="checkbox"/>	CR_PROT	Real	Scalar	0	Output Only

Import Export Save Close

**Note:** It is also possible to create a structure called **coefficients** by choosing **structure** in the **type** field and then making **coef1**, **coef2**, **coef3**, and **coef4** real parameters with a **depth** of 1. This allows you to access the parameters using the structure name and an index value and then place the real parameter in an array. This will be discussed further in the next chapter.

- 5 Click **Save** and then **Close** to exit the **Configuration Editor** window.
- 6 Exit the **Model Library Editor** by closing the window.

The new icon and the variable names will now appear whenever you use the **HollowF** model.



# Inserting the New Model and Running the Simulation

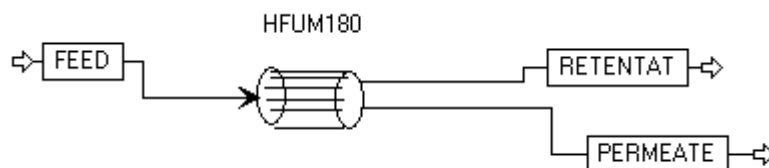
Delete the old model and replace it with your new **HollowF** model, reconnecting the streams, and respecifying the product stream flash.

The named references on the **Configured Variables** sheet must be supported by a short Fortran subroutine that you must create, compile, and link along with the user model subroutine **Mem2.f**. The **Model Library Editor** will write this new subroutine for you.

Insert the **HollowF** model, create a new dll file containing the new subroutine, then run the simulation and examine the results.

## Inserting the New Model

- 1 Open **Membrane2.apw**.
- 2 Press **Shift+F5** and click **OK** twice to reinitialize the simulation.
- 3 Click **Manage Libraries** on the **Customize** tab of the ribbon, and select the check box in front of Ultrafiltration.  
Your Ultrafiltration library has one category called Membranes and its tab appears in the Model Palette.
- 4 On the Main Flowsheet, delete the **MEMBRANE** block.
- 5 Place a **HollowF** block on the Flowsheet.
- 6 Select the **FEED** stream, right-click, select **Reconnect Destination**.
- 7 Move the cursor over the **Feed** port and click once to reconnect the stream.
- 8 Reconnect the **RETENTAT** stream to the **Product** port using the **Reconnect Source** option. If you accidentally select the **PERMEATE** stream first, press **Esc** on the keyboard to cancel.
- 9 Reconnect the **PERMEATE** stream. The order matters because the first stream connected comes first in the output stream data array (**SOUT**). Once both streams are connected they can be separated for clarity.
- 10 Rename the block, a Hollow Fiber Ultrafiltration Membrane with 180 tubes, **HFUM180**.



Now that the Process Flowsheet is complete, check the data sheets for the HFUM180 block. Make sure all the data included with the model by default is correct and add the product stream flash specification.

## Checking and Editing Setup Sheets

- 1 Go to the **HFUM180 (HollowF) - Setup** form.

- 2 Check the **Subroutines** sheet and make sure the subroutine name and path to the Excel file are correct.
- 3 Check the **User Arrays** and **Configured Variables** sheets. The two results parameters that you defined for the **HollowF** model in the **Configuration Editor** do not appear on the **Configured Variables** sheet because they were specified as output parameters; they will appear on a results sheet.

**Note:** The data in the **User Arrays** sheet and **Configured Variables** sheet are the same. If you change any of the numbers on one sheet, the same change will automatically appear on the other sheet. You can alter the numerical data if you wish and it will be changed for the particular block that you placed. For future block placements, the defaults will remain intact.

- 4 Go to the **Stream Flash** sheet and specify a **Temperature & pressure** Flash for each product stream. No data is needed in the Flash specifications area.

Now create the Fortran file to handle the new variable names, compile it, create a shared library including your user subroutine and this new subroutine, and run the simulation.

## Compiling, Linking, and Running

- 1 Open the **Manage Libraries** dialog box, click **Edit** beside Ultrafiltration to open the **Ultrafiltration - Model Library Editor**.
- 2 Select the **HollowF** model, right-click, and select **Edit User Configuration** to open the **HollowF - User Model Configuration Editor**.
- 3 Click **Export** and save the Fortran file as **Mem2Vars.f** to the folder that contains **Mem2.f** (\Working Folders\Aspen Plus\Membrane.)
- 4 Exit the Configuration Editor by closing the window.
- 5 Exit the Model Library Editor and the **Manage Libraries** dialog box.
- 6 Save your run as **Membrane2\_CM.apw** and then exit Aspen Plus so that you can run asplink in a moment.

- 7 From the **Membrane** sub-folder, open your DLOPT file, **Membrane2\_List\_OBJs.opt** and add a line indicating **Mem2Vars.obj**.
- 8 Save the new DLOPT file and exit the text editor.



**Note:** If you use the pre-supplied DLOPT files, be sure to check that the path specified matches your working folder and update it (or remove the path).

- 9 Start the Aspen Plus Simulation Engine to get the DOS window, type **cd Membrane** and press **Enter** to set the directory to the location of your Fortran files.
- 10 Type **aspcomp Mem2Vars** and press **Enter** to compile the new Fortran subroutine. The **Mem2** subroutine was compiled in the previous chapter; so the obj file should already exist.
- 11 Type **asplink DLOPT Membrane2\_List\_OBJs.opt Membrane2** and press **Enter** to create a new shared library from the two obj files.



A new version of **Membrane2.dll** is placed in your **Membrane** sub-folder.

- 12 Close the Aspen Plus Simulation Engine window.
- 13 Check that your other DLOPT file, **Membrane2\_Loc\_dll.opt** is still in the Aspen Plus Working Folder and contains **Membrane2.dll**.
- 14 Start Aspen Plus, and open **Membrane2\_CM.apw**.
- 15 Open the **Run Settings** dialog box. If necessary, type **Membrane2\_Loc\_dll.opt** in the **Linker Options** field to tell Aspen Plus where to find the shared library.
- 16 Click **OK**.
- 17 Run the simulation.
- 18 Go to the **HFUM180 (HollowF) - Results | Summary** sheet to see the entire real and integer parameter lists including the two real parameters that were specified as output only.
- 19 Go to the **Configured Variables** sheet to see just the two output parameters.

Summary		Balance		Configured Variables		Status	
Configured variables							
		Label				Value	
▶	CP_PROT					0.09	
▶	CR_PROT					4.59579	

**20** Save the simulation as **Membrane2\_CM.apw** and exit Aspen Plus.

**Note:** This simulation file uses the same **Mem2** subroutine as **Membrane2.apw**. It uses an updated version of **Membrane2.dll**, an updated version of **Membrane2\_List\_OBJs.opt**, and the same **Membrane2\_Loc\_dll.opt** file as in the previous chapter and produces the same results. It has been named **Membrane2\_CM.apw** to emphasize this connection to the previous simulation.

## 4 Creating an Aspen Plus Fortran Model

Excel is useful for testing and troubleshooting a new model because it is self-contained – during the testing phase, it is not necessary to run Aspen Plus simulations. However, once the model is developed, you may wish to perform the model calculations in the user Fortran subroutine and transfer the results directly to Aspen Plus.

In this chapter, build a User2 Fortran model to replace the Excel model that you developed previously. Use Aspen Plus utility functions that read from and write to the real and integer array stored with the HollowF customized model. Use the named references (such as "DIAM") you specified when you created the model.

The absence of Excel interface functions and the presence of the Aspen Plus utility functions make for compact code in the new subroutine. A complete listing is provided here.

Allow about 30 minutes for this session.

### Putting the Model Equations in the User Subroutine

A Fortran template is available that will make creating a subroutine to perform the calculations for the HollowF model easier. The complete subroutine is listed below.

- 1 Put a copy of \Engine\User\Urus2.f in the folder you are using for your Fortran subroutines. For example, copy this file to Aspen Plus working directory.
- 2 Edit Urus2.f and change the name of the subroutine to HFUM.

**Note:** Any filename with the .f extension is allowed and any subroutine name up to six characters is allowed.

- 3 Rewrite the subroutine following the complete Code Listing below. Numbers in parentheses after the boldface comments refer to the relevant part of the Code Description that follows the listing.

# Code Listing – Subroutine HFUM

```

SUBROUTINE HFUM (NMATI, MSIN, NINFI, SINFI, NMATO,
2          SOUT, NINFO, SINFO, IDSMI, IDSII,
3          IDSMO, IDSIO, NTOT, NSUBS, IDXSUB,
4          ITYPE, NINT, INT, NREAL, REAL,
5          IDS, NPO, NBOPST, NIWORK, IWORK,
6          NWORK, WORK, NSIZE, SIZE, INTSIZ,
7          LD )

      IMPLICIT NONE
#include "ppexec_user.cmn"
#include "dms_plex.cmn"
      Real*8 B(1)
      Equivalence (B(1),IB(1))
#include "dms_ncomp.cmn"

C   Include files pass additional variables via COMMONs.
C   Ppexec_user.cmn passes USER_NHSTRY. (1) Dms_plex.cmn passes
C   arrays containing component data such as molecular weight. (2)
C   Dms_ncomp.cmn passes NCOMP_NCC. (3)

C ----- Declare arguments -----

      INTEGER NMATI, NINFI, NMATO, NINFO, NTOT,
+          NSUBS, NINT, NPO, NIWORK, NWORK,
+          NSIZE, NREAL

      INTEGER IDSMI(2,NMATI), IDSII(2,NINFI),
+          IDSMO(2,NMATO), IDSIO(2,NINFO),
+          IDXSUB(NSUBS), ITYPE(NSUBS), INT(NINT),
+          IDS(2,3), NBOPST(6,NPO),
+          IWORK(NIWORK), INTSIZ(NSIZE), LD

      REAL*8 MSIN(NTOT,NMATI), SINFI(NINFI),
+          SOUT(NTOT,NMATO), SINFO(NINFO),
+          WORK(NWORK), SIZE(NSIZE), REAL(NREAL)

C ----- Declare Local Variables -----

      INTEGER OFFSET, IERR, LDATA, KDIAG, IDX(10), NCP, I, J, INDEX,
+          LMW, NTUBES, IPERM, IRET,IFAIL

      REAL*8 DIAM, LEN, DIFF, CG, REJ_COEF, C1, C2, C3, C4, P_PERM,
+          DELTA_P, RHO, MU, FIN, CIN, PIN, UAVE, RE, SC,
+          CP, CR, KM, JM, FP, PRET, XMW, X(10), FLOW

C ----- Declare Functions -----

      INTEGER USRUTL_GET_REAL_PARAM, ! These functions allow access to real
+          USRUTL_GET_INT_PARAM,    ! and integer parameters using named
+          USRUTL_SET_REAL_PARAM    ! references, and to write results data

```

```

                                ! to Aspen Plus. (4)

INTEGER DMS_IFCMNC !Determines offset to universal constant data. (5)

REAL*8  DLOG                                !Standard Fortran function.

C ----- Begin Executable Code -----
C ----- Get configured REAL variables from Aspen Plus. -----

IFAIL = 0
INDEX = 0                                !Used for passing a structure. (6)
IERR = USRUTL_GET_REAL_PARAM('DIAM', INDEX, DIAM) !Put Configured
                                ! Variable called "DIAM" in local variable DIAM. (7)

IF (IERR .NE. 0) THEN                    !Write to History file if error. (8)
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING HYDRAULIC DIAMETER'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('LEN', INDEX, LEN)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING LENGTH'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('DIFF', INDEX, DIFF)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING DIFFUSIVITY'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('GEL_CONC', INDEX, CG) !Local variable
IF (IERR .NE. 0) THEN                            !can have any name.
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING GEL CONCENTRATION'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('REJ_COEF', INDEX, REJ_COEF)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING REJECTION COEFFICIENT'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('COEF1', INDEX, C1)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING COEF1'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('COEF2', INDEX, C2)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING COEF2'
    IFAIL = 1
END IF

```

```

IERR = USRUTL_GET_REAL_PARAM('COEF3', INDEX, C3)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING COEF3'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('COEF4', INDEX, C4)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING COEF4'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('PERM_PRES', INDEX, P_PERM)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING PERMEATE PRESSURE'
    IFAIL = 1
END IF

IERR = USRUTL_GET_REAL_PARAM('DELTA_P', INDEX, DELTA_P)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING PRESSURE DROP'
    IFAIL = 1
END IF

C ----- Get configured INTEGER variables from Aspen Plus -----

IERR = USRUTL_GET_INT_PARAM('NTUBES', INDEX, NTUBES)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR FETCHING NUMBER OF TUBES'
    IFAIL = 1
END IF

C ----- Calculate viscosity -----

CALL SHS_CPACK(MSIN(1,1), NCP, IDX, X, FLOW)      !Pack stream data. (9)
KDIAG = 4
CALL PPMON_VISCL(MSIN(NCOMP_NCC+2,1), MSIN(NCOMP_NCC+3,1), X, NCP,
+               IDX, NBOPST, KDIAG, MU, IERR)      !Calculate viscosity,
                                                    !put result in MU. (9)

IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY, *) ' ERROR EVALUATING VISCOSITY FOR FEED'
    IFAIL = 1
END IF

IF (IFAIL .EQ. 1) RETURN

C ----- Get location of molecular weight data.-----

LMW = DMS_IFCMNC('MW') !Offset to component molecular weight data. (10)

C ----- Model Equations -----

RHO = MSIN(NCOMP_NCC+8,1) / 1000.      !Density of feed stream
                                      !in g/cm^3. [= kg/l] (11)
MU = MU * 10.                          !Viscosity of feed stream in g/cm/s.
FIN = MSIN(NCOMP_NCC+1,1) * MSIN(NCOMP_NCC+9,1) / RHO * 3600.

```



```

                                !Feed flow in l/hr. (11)
CIN = MSIN(2,1) * B(LMW+2) / FIN * 3600000. !Protein conc.
                                !in feed stream in g/l. (11)
PIN = MSIN(NCOMP_NCC+3,1) / 101325. ! Feed stream pressure
                                !in atmospheres. (11)
UAVE = FIN / (DIAM**2*3.14*NTUBES/4.) * 1000. / 3600.
                                !Bulk average velocity in cm/s.
RE = DIAM * UAVE * RHO / MU      !Reynolds number.
SC = MU / (RHO * DIFF)          !Schmidt number.

CP = CG * (1 - REJ_COEF)         !Permeate stream protein conc. in g/l.
KM = C1 * RE**C2 * SC**C3 * (DIAM/LEN)**C4 * DIFF/DIAM/100.
                                !Mass xfer coef in m/s.
JM = KM * DLOG((CG - CP) / (CIN - CP)) !Volumetric flux in m/s.
FP = JM * DIAM * LEN * 3.14 * NTUBES*100*3600/1000.
                                !Permeate stream flow rate in l/hr
CR = (CIN - FP/FIN*CP) / (1 - FP/FIN) !Retentate stream protein conc.
                                ! in g/l.
PRET = PIN - DELTA_P             !Retentate stream pressure in atm.

C ----- Assume PERMEATE stream is first, switch if not. -----

    IPERM = 1
    IRET = 2
    IF (IDSMO(1,1) .EQ. 'RETE') THEN      !IDSMO is an argument passed to the
        IPERM = 2                          !subroutine. It contains the outlet
        IRET = 1                          !stream ID's. (12)
    END IF

C ----- Fill SOUT array for PERMEATE stream. -----

    SOUT(1,IPERM) = FP * RHO / 3600. / B(LMW+1) !Approximate water flow
                                                !in kmol/s. (13)
    SOUT(2,IPERM) = FP * CP / 1000. / 3600. / B(LMW+2)
                                                !Protein flow in kmol/s. (13)
    SOUT(3,IPERM) = SOUT(1,IPERM) + SOUT(2,IPERM)
                                                !Total flow in kmol/s. (13)
    SOUT(4,IPERM) = MSIN(4,1)                !Temp. unchanged from feed (K). (13)
    SOUT(5,IPERM) = P_PERM * 101325.         !Pressure in N/m^2. (13)

C -- Fill SOUT array for RETENTAT stream using values from PERMEATE stream.

    SOUT(1,IRET) = MSIN(1,1) - SOUT(1,IPERM) !Water flow in kmol/s. (13)
    SOUT(2,IRET) = MSIN(2,1) - SOUT(2,IPERM) !Protein flow in kmol/s. (13)
    SOUT(3,IRET) = SOUT(1,IRET) + SOUT(2,IRET) !Total flow in kmol/s. (13)
    SOUT(4,IRET) = MSIN(4,1)                !Temp. unchanged from feed (K). (13)
    SOUT(5,IRET) = PRET * 101325.           !Pressure in N/m^2. (13)

C -- Now set values of the two variables designated as output parameters. --

    IERR = USRUTL_SET_REAL_PARAM('CP_PROT', INDEX, CP)      !The HollowF
        ! model was designed to accept 2 output parameters. (14)
    IF (IERR .NE. 0) THEN
        WRITE(USER_NHSTRY,*) ' ERROR STORING PERMEATE PROT CONC'
        IFAIL = 1
    END IF

```

```

IERR = USRUTL_SET_REAL_PARAM('CR_PROT', INDEX, CR)
IF (IERR .NE. 0) THEN
    WRITE(USER_NHSTRY,*) ' ERROR STORING RETENTATE PROT CONC'
    IFAIL = 1
END IF

RETURN
END

```

## Code Description – Subroutine HFUM

- 1 USER\_NHSTRY is the Fortran unit number for the History File. From Aspen Plus, you can access this file by clicking **History** in **Summary** on the **Home** tab. This command will open the history file with a text editor and it can be saved using the text editor. Aspen Plus will also save a History File if you save the simulation as an apw file.
- 2 The common DMS\_PLEX contains data for all components including the alias, ID, and physical property parameters such as molecular weight. The data is accessed by using a function to find the offset to the location of the data in the B or IB array. For more discussion of the DMS\_PLEX common and examples see the *User Models* manual and steps 5 and 10 in this list.
- 3 NCOMP\_NCC is the number of conventional components in the simulation. In this example, there are two, water and protein.
- 4 The three USRUTL functions return an integer which will have the value zero if there were no errors. These functions allow the user to pass values between the Fortran subroutine and Aspen Plus according to the names defined using the **User Model Configuration Editor**.
- 5 The integer function DMS\_IFCMCN takes a character string as input (for example, 'MW' for molecular weight) and returns the offset to an area of the "Plex" data array. For more information and examples, see the *User Models* manual and step 10 in this list.
- 6 INDEX is an array of indices used in the USRUTL functions when handling a structure. Its value is irrelevant in the present example since none of the data are structures. However, if you created a coefficients structure with four elements you could access them using the following code:

```

INTEGER INDEX(1), Ierr, USRUTL_GET_REAL_PARAM, i
REAL*8 C(4)
.
.
DO i = 1,4
    INDEX(1) = i
    Ierr = USRUTL_GET_REAL_PARAM('coefficients',INDEX,C(i))
END DO

```

The 1-based INDEX array contains only as many elements as there are levels of structure. Each element must be set once per loop. For example, suppose you have nested structures with str1 containing str2a, str2b, and str2c. If str2b has five scalar elements and you wish to access the third, set INDEX(1) = 2 and INDEX(2) = 3. A scalar element within a structure

(for example, coefficients.coef2) may also be referenced directly by its lowest level name ('coef2' in this example) as if it were not in a structure at all: `USRUTL_GET_REAL_PARAM('coef2', 0, C2)`.

- 7 On the Aspen Plus **HFUM180 (HollowF) - Setup | Configured Variables** sheet, you can see that one of the real parameters is called **DIAM**. The function `USRUTL_GET_REAL_PARAM` retrieves the value of this parameter and puts it into a local Fortran variable that, in this example, is named `DIAM`. Of course any declared `Real*8` variable is allowed to hold the value of this parameter.
- 8 This History File, also discussed in Step 1 above, is a useful place for error messages and troubleshooting output statements. For example, a statement like

```
WRITE (user_nhstry,*) 'diam = ', diam
```

might be useful if your calculations are not coming out properly and you want zero in on the problem.

- 9 You can call physical property monitors such as `PPMON_VISCL` to access the Aspen Plus Physical Property system. The `PPMON` subroutines require the stream information in "packed" form (that is, a reduced list of component mole fractions with the zeroes removed) produced by a packing utility such as `SHS_CPACK`. For more information about accessing physical properties see Chapters 3 and 4 of the *User Models* manual.
- 10 For a list of the names to feed to `DMS_IFCMNC` in order to retrieve component data from Aspen Plus, see Chapter 6 of the *User Models* manual. In this example, the offset (`LMW`) to the molecular weight data is obtained using the string '`MW.`'. The molecular weight of the first component listed on the Aspen Plus **Components - Specifications | Selection** sheet (water) is stored in `B(LMW + 1)`. The molecular weight of protein (really lysine) is stored in `B(LMW + 2)`.
- 11 The input (feed) stream array `MSIN` contains `NCOMP_NCC+9` elements for each feed stream. In this example there is only one feed stream so the second index is always 1. The first index may be 1 through 11 since there are two components. The first two elements are the mole flows in `kmol/s` for water and protein. The following nine elements are total mole flow, temperature, pressure, enthalpy, vapor fraction, liquid fraction, entropy, density, and molecular weight of the stream. They are listed in Appendix C of the *User Models* manual and are in SI units. Since the model equations in this example use other units, conversions must be performed.
- 12 The argument list for User 2 models is documented in Chapter 5 of the *User Models* manual. `IDSMO` is an array with dimension (2, `NMATO`) where `NMATO` is the number of material outlet streams. For each stream, there are 2 four-character labels. In this example, the streams are distinguished by the first four letters in the label of each (`RETE` and `PERM`). Alternatively, the programmer can instruct users to always connect the Retentate stream first.
- 13 The `SOUT` array has the same structure as the `MSIN` array. It contains the mole flow, temperature, pressure, and other data for the two product streams. The program calculates flow rates for each component, the total flow rates, set the product stream temperatures equal to the feed stream temperature, and calculates the pressure of each stream. These data are

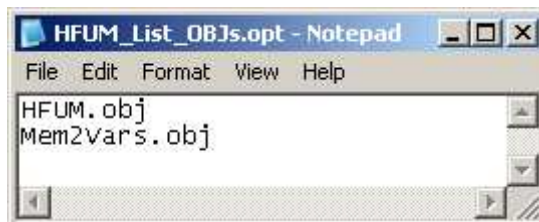
sent to Aspen Plus by filling the first five elements of the SOUT array for each stream.

- 14 Other data not included in SOUT will appear on the **HFUM180 (HollowF) - Results | Configured Variables** sheet because two variables were designated output only when you set up the configured variables for the HollowF model. The protein concentration in g/l for both product streams appears on the results sheet. Since this is user-specified output data, Aspen Plus makes no assumptions about units.

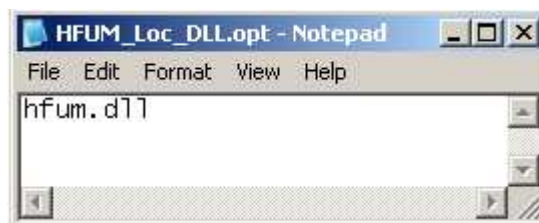
## Compiling and Linking the Subroutine

Use the **aspcomp** command to compile the subroutine. Then create a new DLOPT file that contains the name of the new subroutine. Create the DLL file and place a DLOPT file in the Aspen Plus Working Folder that contains the path to this new DLL.

- 1 Save your Fortran code as **HFUM.f** and exit the development environment.
- 2 Start the Aspen Plus Simulation Engine Window. From the **Start** menu click **Programs | AspenTech | Process Modeling <version> | Aspen Plus | Aspen Plus Simulation Engine**.
- 3 In the DOS prompt window type **cd membrane** if your Fortran files are in a sub-folder of your Aspen Plus Working Directory called **membrane**.
- 4 Type **aspcomp HFUM** to compile the subroutine to create **HFUM.obj**.
- 5 Open **Membrane2\_List\_OBJs.opt** and replace **Mem2.obj** with **HFUM.obj**.
- 6 Save the file as **HFUM\_List\_OBJs.opt** and exit the text editor.



- 7 In the DOS window of the Aspen Plus Simulation Engine, type **asplink dlopt HFUM\_List\_OBJs.opt HFUM** to create **HFUM.dll**.
- 8 From the Aspen Plus Working Folder, open **Membrane2\_Loc\_dll.opt** and replace **Membrane2.dll** with **HFUM.dll**.
- 9 Save the file as **HFUM\_Loc\_DLL.opt** and exit the text editor.

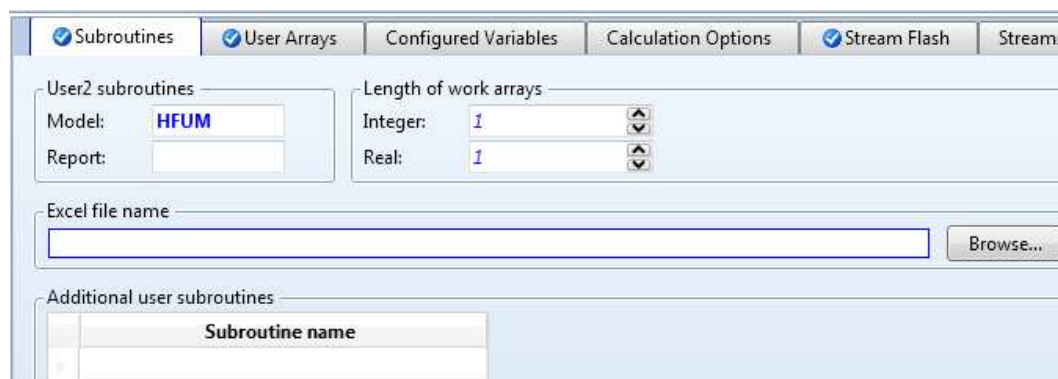


**Note:** If you use the pre-supplied DLOPT files, be sure to check that the path specified matches your working folder and update it (or remove the path).

## Running the Simulation

Use the **membrane2\_cm.apw** simulation from the previous chapter. Since your model calculations will be performed by the Fortran subroutine, remove the reference to the Excel file. Then enter the name of the DLOPT file containing the location of the dll file and run the simulation.

- 1 Open **membrane2\_cm.apw**.
- 2 Select **K**, or press **Shift+F5**, and click **OK** twice.
- 3 Go to the **Setup - Specifications | Global** sheet and change the title to **HFUM**.
- 4 Go to the **HFUM180 (HollowF) - Setup | Subroutines** sheet, change the name of the Fortran subroutine to **HFUM** and delete the reference to the Excel file.



- 5 Open the **Run Settings** dialog box, type **HFUM\_Loc\_DLL.opt** in the **Linker Options** field, and click **OK**.
- 6 Run the simulation.
- 7 Compare your results to those obtained using the Excel model or to the results displayed below. You may wish to set the units set to **SI** for comparison with the Excel spreadsheet.
- 8 Save the run as **HFUM.apw** and exit Aspen Plus.

## Results from the Custom Fortran Model

**HFUM180 (HollowF) - Stream Results | Material** sheet (in SI units):

Material	Heat	Work	Vol. % Curves	Wt. % Curves	Petro. Curves	Poly. C
Display: <span>Streams</span> Format: <span>GEN_M</span> <span>Stream Table</span>						
	FEED	RETENTAT	PERMEATE			
▶ Temperature K	298.1	298.1	298.1			
▶ Pressure N/sqm	250000	199338	101325			
▶ Vapor Frac	0	0	0			
▶ Mole Flow kmol/sec	0.267	0.172	0.095			
▶ Mass Flow kg/sec	4.831	3.119	1.712			
▶ Volume Flow cum/sec	0.005	0.003	0.002			
▶ Enthalpy Gcal/hr	-65.722	-42.375	-23.347			
▶ Mole Flow kmol/sec						
▶ WATER	0.267	0.172	0.095			
▶ PROTEIN	< 0.001	< 0.001	trace			

#### HFUM180 (HollowF) - Results | Configured Variables sheet:

Summary	Balance	Configured Variables	Status
Configured variables			
	Label	Value	
▶	CP_PROT	0.09	
▶	CR_PROT	4.59758	