Aspen Plus®

STEADY STATE SIMULATION

Version 10

Customizing Unit Operation Models

GETTING STARTED

AspenTech®

Use of Aspen Plus and This Manual

This manual is intended as a guide to using Aspen Plus process modeling 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 Aspen Plus and the application of the results obtained.

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

Contents

About *Getting Started Customizing Unit Operation Models*

1 Using Proprietary Unit Operation Models

2 Creating an Aspen Plus Excel® Model

3 New Integration Features for Proprietary Unit Operation Models

4 Creating an Aspen Plus Fortran Model

5 Creating GUI Forms for Aspen Plus Proprietary Models

A Connecting to the Aspen Plus Simulation Engine

About *Getting Started Customizing Unit Operation Models*

Aspen Plus Version 10.1 offers new interfaces for custom or proprietary models in Aspen Plus simulations. In this tutorial, you will learn key features of this interface by building a simulation centered about an external membrane model. The hands-on sessions in this book are described in the following table:

Follow the steps in Chapter	To learn how to
1. Using Proprietary Unit Operation Models	Specify a simulation that uses an external model
2. Creating an Aspen Plus Excel® Model	Develop an external unit operation model in Excel
3. New Integration Features for Proprietary Unit Operation Models	Create: An entry into the Aspen Plus Model menu Custom icons to better represent your models User-defined simulation input and output variables
4. Creating an Aspen Plus Fortran Model	How to access user-defined simulation variables from within an external Fortran routine
5. Creating GUI Forms for Aspen Plus Proprietary Models	Create your own GUI forms for proprietary models

Getting Started Customizing Unit Operation Models assumes that you have an installed copy of the Aspen Plus and User Interface software, as well as a working knowledge and access to either Microsoft Excel 97 or Microsoft Excel 95 and Release 5.0C or higher of Digital’s Visual Fortran. The session in Chapter 5 requires that you have a working knowledge and access to Microsoft Visual Basic Version 5.

Aspen Plus Excel Unit Operations

You can now 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 the model within Excel without performing iterative compile and link steps that are required when writing a Fortran model. You can utilize the full power of Excel including all of its built-in functions and specialized Visual Basic routines within the model.

New Integration Features for Custom Unit Operation Models

Version 10.1 also offers four powerful new features that improve the usability of Aspen Plus simulations containing proprietary models. You can now create:

- Model Libraries containing categorized groups of models for the Aspen Plus Model Menu,
- User-defined variables that become part of the Aspen Plus data structure as an alternative to Real and Integer Arrays that were used in previous versions,
- Custom icons to better represent the equipment that your models represent, and
- Standard and default model input templates.

Aspen Plus Fortran Models

A new facility is available in Version 10.1 that provides read and write access to user-defined variables from within the Fortran user model routine. This simplifies user-defined parameter mapping into external user routines.

It is now possible to develop rich data structures for user-defined unit operation models involving integer, real and character data. The parameters can either be scalar or 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.

Creating GUI forms for Proprietary Models

It is now possible to create your own user interface forms for proprietary or custom models using the same environment used to create built-in Aspen Plus forms. User-defined variables can be easily linked to these forms. The forms can have a free format layout to suit the needs of your model.

The ability to create forms combined with the templating capabilities outlined in Chapter 3 is a very powerful customization capability. It is now possible to create forms for an existing plant unit that hides design parameters for users interested in performing rating only calculations for example. These forms are easily inserted into Aspen Plus by registering the OCX file created by Visual Basic using the Aspen Plus *ApwnSetup* utility.

Visual Basic template files are distributed with Aspen Plus to help reduce the development effort.

Supporting Files

A zip file containing the Aspen Plus backup, Excel, Fortran and Visual Basic files described in this book are available from our internet-based Technical Support Center. From our World Wide Web Home Page, select Technical Support Center then Aspen Plus. See the *Technical Support* section of this preface for details on registering for internet-based technical support.

For More Information

Online Help Aspen Plus has a complete system of online help and context-sensitive prompts. The help system contains both context-sensitive help and reference information. For more information about using Aspen Plus help, see the *Aspen Plus User Guide*, Chapter 3.

Aspen Plus Getting Started Building and Running a Process Model This tutorial includes several hands-on sessions to familiarize you with Aspen Plus. The guide takes you step-by-step to learn the full power and scope of Aspen Plus.

Aspen Plus Getting Started Modeling Processes with Electrolytes This tutorial includes several hands-on sessions to familiarize you with simulating electrolyte systems with Aspen Plus.

Aspen Plus Getting Started Modeling Petroleum Processes This tutorial includes several hands-on sessions to familiarize you with simulating petroleum processes with Aspen Plus.

Aspen Plus Getting Started Customizing Unit Operation Models This tutorial includes several hands-on sessions to familiarize you with the customization of unit operation models with Aspen Plus.

Aspen Plus User Guide The three-volume *Aspen Plus User Guide* provides step-by-step procedures for developing and using an Aspen Plus process simulation model. The guide is task-oriented to help you accomplish the engineering work you need to do, using the powerful capabilities of Aspen Plus.

Aspen Plus reference manual series Aspen Plus reference manuals provide detailed technical reference information. These manuals include background information about the unit operation models and the physical properties methods and models available in Aspen Plus, tables of Aspen Plus databank parameters, group contribution method functional groups, and a wide range of other reference information. The set comprises:

- *Unit Operation Models*
- *Physical Property Methods and Models*
- *Physical Property Data*
- *User Models*
- *System Management*
- *System Administration*
- *Summary File Toolkit*

Aspen Plus application examples A suite of sample online Aspen Plus simulations illustrating specific processes is delivered with Aspen Plus.

Aspen Plus Installation Guides These guides provide instructions on platform and network installation of Aspen Plus. The set comprises:

- *Aspen Plus Installation Guide* for Windows
- *Aspen Plus Installation Guide* for OpenVMS
- *Aspen Plus Installation Guide* for UNIX

The Aspen Plus manuals are delivered in Adobe portable document format (PDF) on the Aspen Plus Documentation CD.

Technical Support

World Wide Web For additional information about AspenTech products and services, check the AspenTech World Wide Web home page on the Internet at:

http://www.aspentech.com/

Technical resources To obtain in-depth technical support information on the Internet, visit the Technical Support homepage. Register at:

http://www.aspentech.com/ts/

Approximately three days after registering, you will receive a confirmation e-mail and you will then be able to access this information.

The most current Hotline contact information is listed. Other information includes:

- Frequently asked questions
- Product training courses
- Technical tips

AspenTech Hotline If you need help from an AspenTech Customer Support engineer, contact our Hotline for any of the following locations:

If you are located in:	Phone Number	Fax Number	E-Mail Address
North America & the Caribbean	+1-617/949-1021 +1-888/996-7001 (toll free)	+1-617/949-1724	support@aspentech.com
South America			
(Argentina office)	+54-11/4393-5308	+54-11/4394-8621	tecnoba@aspentech.com
(Brazil office)	+55-11/5506-0756	+55-11/5506-0567	tecnosp@aspentech.com
Europe, Gulf Region, & Africa			atesupport@aspentech.com
(Brussels office)	+32-2/724-0100	+32-2/705-4034	
(UK office)	+44-1223/312220	+44-1223/366980	
Japan	+81-3/3262-1743	+81-3/3262-1744	atjsupport@aspentech.com
Asia & Australia			atasupport@aspentech.com
(Hong Kong office)	+85-2/2838-6077	+85-2/2833-5642	
(Korea office)	+82-2/761-5800	+82-2/761-5803	

❖ ❖ ❖ ❖

1 Using Proprietary Unit Operation Models

Version 10.1 offers new interfaces for including custom or proprietary models in Aspen Plus simulations. Key features of this interface are illustrated by building a simulation centered about an external membrane model. The required steps to specify a Microsoft Excel model for the membrane are outlined in Chapter 2. The process of creating an entry into the Aspen Plus Model Menu, how to create a custom icon to graphically represent the membrane and custom simulation variables will be illustrated in Chapter 3.

In this session you will build the external membrane model simulation. This session assumes that you have a working knowledge of building Aspen Plus simulations in Version 10, and have completed *Aspen Plus Getting Started Building and Running a Process Model*.

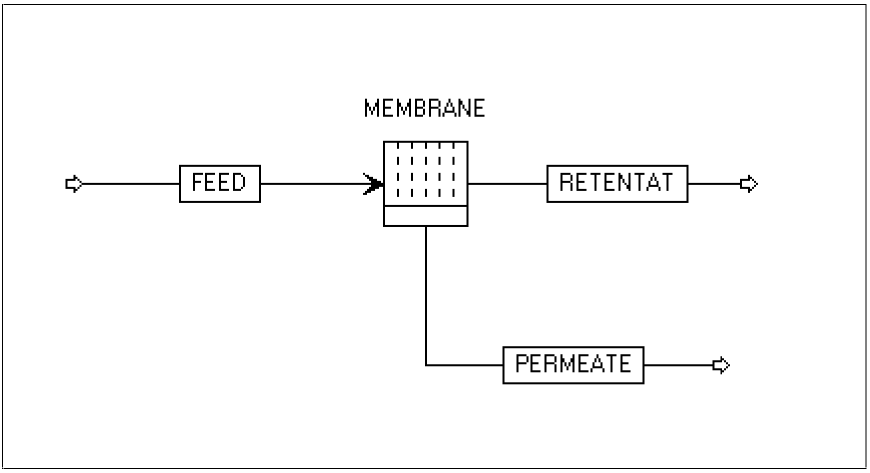
Allow about 30 minutes for this session.

Example Membrane Separation

You will construct an Ultrafiltration Membrane model in this session. 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.

Figure 1.1 Ultrafiltration Membrane Mode



- ➤ You must enter all pertinent data onto the Setup, Component, Property, Stream and Block forms.

Starting Aspen Plus

To start Aspen Plus:

- ➤ From your desktop, click Start and then select Programs.
- ➤ Select AspenTech, then Aspen Plus 10.1-0, then Aspen Plus User Interface.

Note If the Connect to Engine dialog box appears, see Appendix A.

The Aspen Plus Startup dialog box appears. Aspen Plus displays a dialog box whenever you must enter information or make a selection before proceeding. In this simulation, you will use an Aspen Plus template.

➤ To select the template option, click the Template option button.

➤ Click OK.

The New dialog box appears.

You use the New dialog box to specify the Application Type and the Run Type for the new run. Aspen Plus uses the Application Type you choose to automatically set various defaults appropriate to your application.

To specify the Application Type and Run Type for the new run:

➤ Select the General with Metric Units template.

➤ The default Run Type, Flowsheet, is appropriate for this session. Click OK to apply these options.

It will take a few seconds for Aspen Plus to apply these options.

The Aspen Plus Main window is now active.

Building the Flowsheet

In the Ultrafiltration Membrane Model shown in Figure 1.1, there is one feed stream (water-protein feed), one unit operation model (an ultrafiltration membrane), and two product streams (permeate and retentate).

You will now define the process flowsheet.

First, select a unit operation model to simulate the user defined membrane model:

➤ From the Aspen Plus main window, click the User Models tab on the Model Library.

➤ The list of available user blocks is displayed. Move the mouse over any model in the Model Library and read the description in the lower left of the window.

➤ Point to the User2 block.

The User2 model can be represented on the flowsheet by a number of pictorial icons. You can select the default displayed in the Model Library or select a different icon from the icon list:

➤ To choose a different icon, click the down arrow to the right of User2 in the Model Library.

➤ Move the mouse pointer over the displayed icons to view the label for each icon.

In this session, you will use the icon labeled FILTER for the User2 block.

➤ Select the User2 icon that you want to place in your process flowsheet.

➤ Click the icon and drag it to the Process Flowsheet window.

The mouse pointer is in the shape of a box with an arrow, which indicates that only one block will be placed.

➤ In the Process Flowsheet window, release the mouse button where you want to place the block. The icon that you selected appears on the flowsheet.

Next, place the streams for the User2 block by doing the following:

➤ Click the Material Stream icon on the left side of the Model Library.

➤ Move the cursor to the Process Flowsheet window.

For each block in the Process Flowsheet window, all ports that are compatible with the stream type you selected are highlighted.

To connect the feed stream:

➤ Point to the User2 Feed (Required; one or more) port on the User2 icon in the process flowsheet window. Click once to select it.

➤ Move the mouse pointer to a blank part of the process flowsheet window where you want the feed stream to originate in your graphical flowsheet and click once.

A feed stream connecting the selected inlet port is created. Aspen Plus automatically assigns an ID of 1 to this stream.

To connect the retentate product stream:

- Move the mouse pointer to the User2 Product port. Click once to select it.

- Move the mouse pointer to a blank part of the process flowsheet window where you want the product stream to terminate in your graphical flowsheet, and click.

  Product stream 2 is created, connected to the selected outlet port.

Note The Retentate stream must be the first product stream declared since the Retentate will be the first product in our Excel model that will be developed in Chapter 2.

To connect the permeate product stream:

- Move the mouse pointer to the User2 Product port. Click once to select it.

- Move the mouse pointer to a blank part of the Process Flowsheet window and click to create Stream 3.

- When you have finished placing streams, click the Select Mode button ↖ in the upper left corner of the Model Library.

Your graphical simulation flowsheet is now complete. The status indicator in the bottom right of the main window says "Required Input Incomplete" indicating that further input specifications are required for the simulation.

- To rename the streams:

- Move the mouse pointer to Stream 1. Click once to select it.

- Click the right mouse button and select Rename stream.

- Type Feed in the dialog box to rename Stream 1 to Stream Feed.

- Rename Stream 2 to Retent and Stream 3 to Permeate.

Adding Data to the Process Model

Now that you have defined your flowsheet graphically, use the Data Browser input forms to enter the remaining required information.

➤ Click the Next button in the main window toolbar.

➤ Click OK to display the first required input form.

Aspen Plus displays a Data Browser window containing the Setup Specifications forms. The Data Browser window title bar contains the form name.

➤ Click the Next button on the Data Browser window toolbar, to move to the next required input form.

The Components Specifications form appears.

The components for the process in this example are water and protein. For each component, you must enter a unique component ID.

➤ In the first Component ID box, type WATER and press Enter.

➤ Click the next blank Component ID box below WATER. Type PROTEIN and press Enter.

Because Protein is not in the Aspen Plus databank, the component name and formula remain blank. You will use the component data for the amino acid lysine to represent the protein.

➤ Click the blank Component Name box. Type the name LYSINE and press Enter.

➤ Click the Next button on the Data Browser window toolbar.

➤ The Properties Specifications Global sheet appears.

For this simulation, use the IDEAL model to estimate the properties.

➤ Click the arrow to the right of the Base Method box to display the available thermodynamic property methods in Aspen Plus.

➤ Scroll down the list to the IDEAL property method. Click to select this property method.

➤ Click the Next button on the Data Browser window toolbar.

The Required Properties Input Complete dialog box appears. This dialog box allows you to select additional input forms for physical property parameters and estimation.

➤ Click OK to close the dialog box.

The Stream Input Specifications sheet for Stream FEED appears, as indicated in the Data Browser window title bar.

➤ Enter the following state variable and component flow specifications for the feed stream:

Temperature	**25**	**C**
Pressure	**2.5**	**Atm**
Total flow value	**17,500**	**L/hr**
Protein mass-conc value	**3**	**gm/L**
Solvent	**water**	

➤ Click Next on the Data Browser window toolbar.

The Flash Options subform for the feed appears.

➤ Set the Valid Phases to Liquid-Only.

➤ Click Next on the Data Browser window toolbar.

The User2 Setup form for block B1 appears.

➤ Click Next on the Data Browser window toolbar.

➤ On the User2 Setup Subroutines sheet, enter the name of the User2 subroutines Model as USRXLS.

USRXLS is the name of the Excel Fortran Interface file that you will customize in Chapter 2. This step would not be required in many applications when all required input to the Excel model can be passed through the feed streams and parameter tables. You will modify this file to also pass the feed stream viscosity to the Excel model.

- Enter the full path where the Excel file will be located. For example, c:\gsg\excel\memmodel2.xls.

You must specify real and integer parameters for the model. In Chapter 3, you will create special variables for these parameters to improve the usability of the model.

There is one integer parameter, the number of tubes. To specify:

- On the User Arrays sheet, enter **one** into the Number of Parameters field for Integers.

- Enter **180** for the first integer value.

There are thirteen real parameters, they are:

1	0.1	cm	X	Channel Hydraulic Diameter
2	400	cm	L	Channel Length
3	6E-07	cm2/s	Dab	Diffusivity
4	450	g/l	Cg	Gel Concentration at Membrane Surface
5	0.9998	g/l	R	Rejection Coefficient of Solute
6	0.023		C1	k=C1*(Re**C2)*(Sc**C3)*((x/L)**C4)*(D/x)
7	0.8		C2	
8	0.33		C3	
9	0.		C4	
10	1		Pperm	Permeate Pressure
11	0.5	atm	DeltaP	Pressure Drop
12		g/l	Cp	Protein permeate concentration
13		g/l	Cr	Protein retentate concentration

- On the User Arrays sheet, enter **13** into the Number of Parameters field for Reals.

- In the Real column, type **0.1**, **400**, **6e-7**, **450**, **0.9998**, **0.023**, **0.8**, **0.33**, **0.**, **1**, **0.5** for the real input values. The last two parameters are results that will be calculated by the model.

You should use Aspen Plus to flash the product streams to keep the flowsheet consistent. The unit operation model that you will develop in the next chapter will set the pressure and temperature of the outlet streams. You will make a product flash specification to cause Aspen Plus to evaluate the enthalpy and other stream properties based on this temperature and pressure. To do this:

- On the Stream Flash sheet, select stream PERMEATE.

- ➤ Select a flash type of Temperature and Pressure to specify a TP flash.
- ➤ On the Stream Flash sheet, select stream RETENT and also select a flash type of Temperature and Pressure.

Save Simulation File

The simulation can be saved in either backup or document format. You will use this simulation in the following chapters . To save the simulation:

- ➤ From the File menu, select Save As and assign a name for the bkp file.
- ➤ Exit the User Interface.

❖ ❖ ❖ ❖

2 Creating an Aspen Plus Excel® Model

Using an Excel Workbook as an Aspen Plus user-defined unit operation model is one of the big Release 10.1 benefits. Once the input variables and physical properties used by Excel have been identified, the model can be developed within Excel without performing iterative compile and link steps that are required when writing a Fortran model. The full power of Excel including all of its built-in functions and specialized Visual Basic routines can be used within the model.

In this session, you will learn how to implement the Membrane Model from Chapter 1 as an Excel Aspen Plus model.

Allow about 2 hours for this session.

This session assumes that you have a working knowledge and access to either Microsoft Excel 97 or Microsoft Excel 95 and Release 5.0C or higher of Digital's Visual Fortran.

Specifying the Excel Model

An Excel template is available to simplify the specification process. This template contains built-in datasheets to serve as the input and output interface to Aspen Plus. Visual Basic Helper functions are also provided to aid model development. The templates are named UserTemplate97.xls and UserTemplate95.xls for the 97 and 95 versions of Excel, respectively.

➤ In the Windows Explorer, move to the Engine User subfolder and open the template file. Select the *Enable Macros* button to complete the open operation.

Excel Input Variables

When the model is called by Aspen Plus, the Excel interface copies input variables to Excel input sheets in table format. By default, the complete stream vector for each feed stream, and the complete integer and real work variable tables are automatically copied to the Aspen_Input, Aspen_IntParams, and Aspen_RealParams Excel sheets. No modifications to the Excel Interface file are necessary when no additional input is needed.

In this example, you will modify the Excel Interface file to add viscosity to the bottom of the Aspen Plus stream vector and change the flow basis from mole to mass. For more information on supporting changes to this file, see Tailoring the Excel Interface File, later in this chapter.

Optionally, you can add text comments to Aspen_Input, Aspen_RealParams and Aspen_IntParams starting in column 3 as shown in Table 2.2. These comments will help prevent reference errors when developing the Excel model.

➤ Enter the input data shown for the Aspen_Input, Aspen_RealParams, and Aspen_IntParams Excel Sheets from Tables 2.1, 2.2, and 2.3.

These data will be updated by the Aspen Plus Excel Interface at run time. The data added now will only be used for model development purposes. Zeros can be entered for the enthalpy and entropy since these data are not used in the model.

➤ Assign the names shown for the first 11 variables in column 1 of Aspen_RealParams to the cells containing that data. For example, name cell B$2, *x,* so that you can refer to this data later as the variable *x*. In Excel 97, select the Name command from the Insert menu. The last two variables are results that will be calculated on Sheet1.

Table 2.1 Aspen_Input Excel Sheet

INPUT	FEED	
WATER	4.830	
PROTEIN	0.015	
TOTFLOW	4.845	(Kg/s)
TEMP	298.	(K)
PRESSURE	25300.	(Pa)
ENTHALPY	0	(J/Kg)
VAP FRAC	0	
LIQ FRAC	1	
ENTROPY	0	(J/Kg-K)
DENSITY	996	(Kg/m3)
MOLE WT	18.1	
MUMX	0.0009	(N/sec-m2)

Table 2.2 Aspen_RealParams Excel Sheet

REALPARAMS	1		
X	0.1	cm	Channel Hydraulic Diameter
L	400	cm	Channel Length
Dab	6E-07	cm2/s	Diffusivity
Cg	450	g/l	Gel Concentration at Membrane Surface
Rej	0.9998	g/l	Rejection Coefficient of Solute
Ca	0.023		k=Ca*(Re**Cb)*(Sc**Cc)*((x/L)**Cd)*(D/x)
Cb	0.8		
Cc	0.33		
Cd	0		
Pperm	1		Permeate Pressure
DeltaP	0.5	atm	Pressure Drop
Cp	0.09	g/l	Protein Permeate concentration
Cr	3.32	g/l	Protein Retentate concentration

Table 2.3 Aspen_IntParams Excel Sheet

INTPARAMS	1	
1	180	Number of Parallel Membranes

Aspen Plus Helper Functions

The Aspen Plus Excel template defines several helper functions to support references to input and output tabular data by row and column name. All helper functions are prefixed with "ah" to denote Aspen Helper. All helper functions are implemented as Visual Basic routines that you can modify using Excel's built-in Visual Basic Editor.

Tip Access the Visual Basic Editor in Windows98 by selecting Macro from the Tools menu, then select Visual Basic Editor.

The *ahGetValue* helper function enables data reference based on Excel row and column names. For example, you can access the feed temperature copied to Aspen_Input by the interface routine with the function: *ahGetValue(Aspen_Input, "TEMP").*

You can also specify the column name if more than one feed stream is present. For example, if two streams FEED1 and FEED2 are present, you could use *ahGetValue(Aspen_Input,"TEMP","FEED2")* to retrieve the temperature for FEED2.

Note The *ahGetValue* function does an "Equals To" type match. It locates the row containing the specified text in the first column of that row. The text match is case sensitive.

Another useful helper function is *ahSetValue.* This function copies a result to a cell that is defined by its row and column identifier. For example, *ahSetValue(Aspen_Output,"TEMP","RETENTAT")* would set the Retentate temperature on the Aspen_Output sheet that will be later read back by the Excel Interface Routine.

Other useful functions are *ahNumStreams, ahNumParams, ahNumComps, ahCompName, and ahStreamName.* The first three functions return the number of streams, number of parameters and number of components that are present on one of the input sheets. For example, *ahNumStreams(Aspen_Input)* will return the number of feed streams that are present on the Aspen_Input sheet.

Note If you modify the Interface Routine so that additional data is appended to the standard stream vector, the number of properties in the ahNumComps statement, *ahNumComps= r.rows.Count – 9,* should be updated with the correct number. This is necessary since the stream vector contains both stream properties and component flow rates. The number of components is equal to the number of component flow rates minus the default nine standard stream properties.

The *ahCompName and ahStreamName* functions return the index that correspond to the order of the component or stream. For example, the function *ahCompName(Aspen_Input,"PROTEIN")* will return a value of 2 since PROTEIN is the second component on the Aspen_Input sheet. The *ahStreamName* function returns the column number for the column containing the referenced stream.

Tip The range name such as Aspen_Input used in the helper functions are updated by the Excel Interface Routine. In cases where the stream vector or real and integer parameter tables are expanded beyond their typical sizes, the ranges of these tables can be changed manually to test the model before running Aspen Plus. In Excel97, update the range boundaries by selecting Name from the Insert menu.

For further information on Helper Functions, see Chapter 5 of the *Aspen Plus User Models Reference Manual*.

Aspen Plus Hook Functions

Other functions are provided to control model execution, initialization strategies and clean-up operations at the end of model execution. Three functions, *AspenStartIteration, AspenCalculate,* and *AspenEndIteration,* are called by the Aspen Plus interface routine. *AspenEndRun* is called by Aspen Plus after the simulation has converged. These routines can be modified to execute any valid Excel command before the input data is copied to Excel, the workbook calculations are invoked or after the results are read from Excel.

For example, the Visual Basic command *ThisWorkbook.Save* can be entered into the AspenEndRun hook function to cause the contents of the Workbook to be automatically saved at the end of the simulation. The use of this command is highly recommended.

For further information on Hook Functions, see Chapter 5 of the *Aspen Plus User Models Reference* Manual.

Implementing the Membrane Model

You will enter the governing equations for the Membrane Model on Sheet1. An overview of the model variables is shown in Table 2.5.

➤ Enter the variable names and units into columns 1 and 3 as shown in Table 2.5. The formulas in Column 2 for each variable will be specified in the next section.

Table 2.5 Excel Sheet1UltraFiltration Sample Membrane Model

Adjusted Aspen Plus Input		
Rho	0.993	g/cm3
Mu	0.009	g/cm/s
Fin	17500.	l/hr
Cin	3.000	g/l
Pin	2.5	atm

Intermediate Calculations		
Uave	3.44E+03	cm/sec
Re	3.74E+04	
Sc	1.53E+04	

Output		
Cp	0.09	g/l
Cb	4.60	g/l
k	1.51E-04	m/s
J	0.00076	m/s
Fp	6.20E+03	l/hr
Pret	202650	N/m2

Adjusted Aspen Plus Input

To read the Aspen Plus stream vector variables that were copied into the Aspen_Input sheet and perform units conversion:

➤ Enter the formulas for the variables Rho (total density), Mu (total viscosity), Fin (volumetric flow rate), Cin (Protein feed concentration), and Pin (feed pressure).

```
Rho:    =ahGetValue(Aspen_Input,”DENSITY”)/1000.
Mu:     =ahGetValue(Aspen_Input,“MUMX”)*10.
Fin:    =ahGetValue(Aspen_Input,“TOTFLOW”)/ ahGetValue(Aspen_Input,
                                             “DENSITY”)*3600000.
Cin:    =ahGetValue(Aspen_Input,“PROTEIN”)/Fin*3600000.
Pin:    =ahGetValue(Aspen_Input,“PRESSURE”)/101325.
```

Each cell was named by variable name to enable the use of these names in the model equations. For example, assign *Rho* as the name for the cell holding the total density formula.

Intermediate Calculations

General quantities used throughout the model are defined in this section.

➤ Define the bulk velocity, Uave, and the dimensionless Reynolds and Schmidt numbers.

```
Uave: = Fin/(x^2.*3.14*n/4.)*1000/3600
Re:      =x*Uave*rho/mu
Sc:     =mu/(rho*Dab)
```

The *Dab* and *n* parameters were copied from the Aspen Plus user-defined real and integer tables. Names assigned by Excel to these input fields can be used in place of the *ahGetValue* functions. An advantage of the helper functions is their dynamic behavior. The correct value will be returned if the location where the Dab variable is written changes. This can happen if more variables are added or removed from the Real Parameter list. You could have entered the statement *ahGetValue(Aspen_Input, “Dab”)* instead of using the Dab Excel name.

Output

Aspen Plus calculates the variables on the Aspen_Output sheet. To perform the membrane calculations:

➤ Enter the model equations:

```
Cp:     =Cg*(1-Rej)
Cb:     =(Cin -Fp/Fin*Cp)/(1-Fp/Fin)
k:      =Ca*(Re^Cb)*(Sc^Cc)*((x/L)^Cd)*(Dab/x)/100.
J:      =k*LN((Cg-Cp)/(Cin-Cp))
Fp:     =J*(x*L*3.14*n)*360
Prem:   =(Pin-deltaP)*101325
```

The variables Dab, Rej, x, L, n, deltaP, Ca, Cb, Cc, Cd are Excel named variables that were assigned to the Integer and Real parameters tables.

➤ Assign the names Cp and Cr to the cells containing the formulas for Cp and Cb. Cr is chosen for the retentate concentration since Cb was already used to name the second coefficient in the real parameter table. Enter the formulas on the Aspen_RealParams sheet *=Cp* for the variable Cp and *=Cr* for the variable Cb. These results will be copied back to Aspen Plus and displayed in the Real Parameter table as a result.

Excel Output Variables

After the worksheet calculations are complete, Aspen Plus reads the component mole flows, total mole flow, stream temperature and pressure that are written to Aspen_Output. In this section, we will specify the cell formulas to produce the output table shown in Table 2.4. You will modify the Excel interface routine later to enable Aspen Plus to accept mass flow instead of mole flow.

Table 2.4 Aspen_Output Excel Sheet

Retentate	Permeate
3.10	1.71
1.44E-2	1.55E-4
3.12	1.71
298.15	298.15
101325	202650

The component flow rates are entered in the same order as the Aspen_Input sheet. The top-down row order are water and protein mass flow, total mass flow, temperature and pressure. SI units are used.

The permeate stream is calculated first. The retentate stream is then calculated from mass balance.

➤ Define the output stream data for Aspen Plus. The remaining stream vector data will be determined by Aspen Plus from temperature, pressure and composition.

Permeate Stream

Water Flow:	=Fp*Rho/3600.
Protein Flow:	=Fp*Cp/1000/3600.
Total Flow:	=FlowWater+FlowProt
Temperature:	=ahGetValue(Aspen_Input,"TEMP")
Pressure:	=ahGetValue(Aspen_RealParams, "Pperm")*101325

Retentate Stream

Water Flow:	=ahGetValue(Aspen_Input,"WATER")-FlowWater
Protein Flow:	=ahGetValue(Aspen_Input,"PROTEIN")-FlowProt
Total Flow:	=ahGetValue(Aspen_Input,"TOTFLOW")-FlowPerm
Temperature:	=ahGetValue(Aspen_Input,"TEMP")
Pressure:	=Pret

The permeate water and protein flows were assigned the Excel names FlowWater and FlowProt. FlowPerm was assigned to the total Permeate mass flow. Pperm was a real input variable specified in the Aspen Plus simulation. Pret is an Excel named variable that was defined on Sheet1.

Save Worksheet in Work Directory

Save the Excel sheet in your Aspen Plus work directory using the Save As option from the Excel File menu. You should use the same filename and directory path specified in the User2 block in Chapter 1.

Linking the Excel Model with Aspen Plus

Aspen Plus automatically calls an interface routine to build the Excel input data tables and to copy back tabular results. When additional information is needed beyond these default tables, the functionality can be easily added by tailoring the interface file. This interface is stored as Usrxls.f in the User subdirectory of the Engine folder. You will be editing this file for illustrative purposes to demonstrate how an additional property can be passed to Excel that is outside the default Aspen Plus stream vector and cannot be passed via the Real or Integer Parameter table.

Note All Aspen Plus Fortran template routines are delivered in a User folder located beneath the Aspen Plus Engine folder.

Tailoring the Excel Interface File

To implement the membrane model, the viscosity of the feed stream must be passed to the Excel Aspen_Input table. To simplify the Excel calculations you will also pass the component flow rates on a mass basis.

The interface routine calls StartIteration which in turn calls the *AspenStartIteration* hook function to initialize the Excel model. The input tables for the Aspen_Input, Aspen_IntParams and Aspen_RealParams sheets are built next. WriteTable is called to send these tables to Excel. The CalculateData routine which calls the *AspenCalculate* hook function is called to invoke the Excel calculations. The ReadTable routine is called to retrieve the product stream results. Special care must be taken to ensure that the Excel stream order is the same as the order in which the Aspen Plus streams were declared. For example, the Retentate stream resides in the first column of the Aspen_Output sheet. It was also the first stream that was declared and will have an index of 1 in the outlet vector SOUT. Finally, EndIteration is called to close the current Excel link.

You will edit the interface file in this section to implement these changes:

- ➤ The template for the Excel Interface file is stored with the User Fortran templates. Copy the template file Usrxls.f from the Aspen Plus Engine\User subfolder to your work directory.

- ➤ Microsoft's Developer Studio can be used to edit this file. Invoke it by double clicking the fortran file.

- ➤ Both the declarations and body of the routine are shown in Figures 2.1 and 2.2. The altered functionality is highlighted. The SHS_CPACK and PPMON_VISCL are Aspen Plus callable routines that are documented in chapters 3 and 4 of the *Aspen Plus User Models Reference Manual*. These routines are used to obtain the viscosity of the feed stream. USRUTL_CHARTOH is an Aspen Plus utility that converts character data to hollerith format.

Figure 2.1 Excel Interface Routine Declarations

```
C     Aspen Plus Interface for Excel Proprietary Unit Operation Models
C     The Input and Output data tables for the Excel model are
C     built by default from the Mixed Stream Vector for the feed
C     and product streams.  This template can be tailored to include
C     more information.
C
      SUBROUTINE USRXLS (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     )

      IMPLICIT NONE
C
C     DECLARE DIMENSIONED VARIABLE LENGTHS
C
      INTEGER NMATI, NINFI, NMATO, NINFO, NTOT,
     +        NSUBS, NINT,  NPO,   NIWORK,NWORK,
     +        NSIZE, NREAL
C
C     DECLARE ARGUMENTS
C
      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 Commons -----------------------
#include "dms_plex.cmn"
#include "dms_ncomp.cmn"
#include "ppexec_user.cmn"
      REAL*8 B(1)
      EQUIVALENCE (B(1),IB(1))
C
C----------------------- Declare Local Variables ----------------------
C
      INTEGER INDEX,    OFFSET,   NUM_COLS,   NUM_ROWS, IERR,
     +        BLANK(2), KREAL(3), KINT(3),    INPUT(2), OUTPUT(2),
     +        LDATA,    I,J,K,    LEN,        KDIAG,    IDX(10),
     +        EXCEL_NAME(16),     DMS_IFCMNC, LMW,      NCP,
     +        ROWNAMES(2,NTOT+1), REALROWS(2,NTOT+1)

      REAL*8  STREAMDATA((NTOT+1)*NMATI), XMW, X(10), FLOW
      CHARACTER*8 VarNames(10)
      CHARACTER*8 RealLabels(13)
C
C--------------------------- Data section -----------------------------
C
      DATA BLANK       /4H    , 4H    /
      DATA INPUT       /4HINPU,4HT   /
      DATA OUTPUT      /4HOUTP, 4HUT  /
      DATA KREAL       /4HREAL, 4HPARA, 4HMS  /
      DATA KINT        /4HINTP, 4HARAM, 4HS   /
C     Variable labels that will be copied to the Excel input tables
      DATA VarNames   /’TOTFLOW ’,’TEMP    ’,
     +                 ’PRESSURE’,’ENTHALPY’,
     +                 ’VAP FRAC’,’LIQ FRAC’,
     +                 ’ENTROPY ’,’DENSITY ’,
     +                 ’MOLE WT’,’MUMX  ’ /
      DATA RealLabels /’x        ’,’L       ’,’Dab     ’,’Cg      ’,
     +                 ’R        ’,’Ca     ’,’Cb      ’,’Cc      ’,
     +                 'Cd       ','Pperm  ','DeltaP   ' ‘Cp      ‘,
     +                  ‘Cr      ‘/
```

Figure 2.2 Excel Interface Routine Body

```
C----- Execute StartIteration Aspen Plus Excel Hook Function -------

       CALL USRUTL_GETEXCEL(EXCEL_NAME, LEN)
       CALL StartIteration(IERR, EXCEL_NAME, 64, IDS(1,1), 8)
       IF(IERR .NE. 0) GO TO 1000

C------------------ Build Aspen_Input Data Table ---------------------

C     Component flow rates and stream variables will be entered as row data
C     A separate column will be entered for each material stream
C     The number of rows equals the 9 stream vector variables plus viscosity
C     The number of columns equals the number of material inlet streams

       NUM_ROWS= NCOMP_NCC+10
       NUM_COLS= NMATI

C     The component ids will be retrieved from the Plex data structure.
C     Each integer variable stores 4 characters of text.
C     Add 8 blank trailing characters for property labels.

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

C     Convert character variable names to hollerith table row names
      DO J=1, 10
        CALL USRUTL_CHARTOH(VarNames(J), RowNames(1,Ncomp_Ncc+J), 8)
      END DO

C     Copy in the stream vector for each material feed
      LMW= DMS_IFCMNC('MW')
      DO J=1, NUM_COLS
        INDEX= (J-1)*NUM_ROWS
          DO I=1, NUM_ROWS
          IF(I .EQ. NUM_ROWS) THEN
C           Insert viscosity onto end of list
C           Call Aspen Plus utility to form composition vector, x

            CALL SHS_CPACK(MSIN(1,J), NCP, IDX, X, FLOW)
            KDIAG=4
             K=NCOMP_NCC
            CALL PPMON_VISCL(MSIN(K+2,J), MSIN(K+3,J), X, NCP, IDX,
     +                       NBOPST, KDIAG, STREAMDATA(INDEX+I), IERR)
          ELSE
C           Copy over mass flows instead of mole flows
            XMW=1
            IF(I .LE. NCOMP_NCC) XMW= B(LMW+I)
               STREAMDATA(INDEX+I)= MSIN(I,J)*XMW
          ENDIF
C         Change total mole flow, stored at NCC+1,to total mass flow
          IF(I .EQ. NCOMP_NCC+1)
     +       STREAMDATA(INDEX+I)= STREAMDATA(INDEX+I)*MSIN(I+8,J)
          END DO
      END DO

      LDATA= NUM_COLS*NUM_ROWS*8
      CALL WriteTable (IERR,     INPUT,    8,
     +                 NUM_ROWS, ROWNAMES, 8,
     +                 NUM_COLS, IDSMI,    8,  2, STREAMDATA, LDATA)
      IF(IERR .NE. 0) GO TO 1000
```

Figure 2.2 (continued) Excel Interface Routine Body

```
C---------------- Build Aspen_IntParams Data Table --------------------

      LDATA= NINT*8
      CALL WriteTable (IERR,    KINT,    12,
     +                 NINT,    BLANK,    0,
     +                    1,    BLANK,       0,  1, INTV, LDATA)
      IF(IERR .NE. 0) GO TO 1000


C---------------- Build Aspen_RealParams Data Table -------------------

C     Convert character variable names to hollerith table row names

      DO J=1, 11
        CALL USRUTL_CHARTOH(RealLabels(J), RealRows(1,j), 8)
      END DO
      LDATA=NREAL
      CALL WriteTable (IERR,    KREAL,   12,
     +                 NREAL,   REALROWS, 8,
     +                     1,   BLANK,    0,  2, REALV, LDATA)
      IF(IERR .NE. 0) GO TO 1000

C------ Execute CalculateData Aspen Plus Excel Hook Function ---------

      CALL CalculateData(IERR)
      IF(IERR .NE. 0) GO TO 1000

C------------------ Retrieve Aspen_Output Data Table ------------------

      NUM_ROWS= NCOMP_NCC + 3
      NUM_COLS= NMATO
      LDATA=NUM_ROWS*NUM_COLS*8
      CALL ReadTable (IERR,     OUTPUT,  8,  NUM_ROWS,
     +                NUM_COLS, 2,       STREAMDATA,   LDATA)
      IF(IERR .NE. 0) GO TO 1000

C     Copy back stream vector for each product stream
C     Must ensure consistent ordering of Aspen Plus and Excel streams

      DO J=1, NUM_COLS
        FLOW= 0.
        INDEX= (J-1)*NUM_ROWS
           DO I=1, NUM_ROWS
              IF(I .LE. NCOMP_NCC) THEN
                FLOW= STREAMDATA(INDEX+I)+FLOW
               SOUT(I,J)= STREAMDATA(INDEX+I)/B(LMW+I)
             ELSE IF(I .EQ. NCOMP_NCC+1) THEN
               SOUT(I,J)= FLOW
             ELSE
               SOUT(I,J)= STREAMDATA(INDEX+I)
             END IF
           END DO
      END DO

C------------------------ Close Excel Link -------------------------------
      CALL EndIteration(IERR)
      IF(IERR .NE. 0) GO TO 1000
      RETURN

C---------------------- Process Error Messages ---------------------------

 1000 Continue
```

Compiling the Excel Interface File

You must compile the interface file to activate the tailored routine. This is an optional step. If the routine had not been changed, this step would not be necessary. To compile the Excel Interface:

- ➤ After making the changes outlined by Figures 2.1 and 2.2, select Save from the File menu.
- ➤ Exit the editor.
- ➤ Open the Aspen Plus Simulation Engine Window. From the Start Menu, select Programs, then Aspen Tech, Aspen Plus 10.1-0 then Simulation Engine. This will activate a special DOS Prompt Window.
- ➤ Move to the default directory using the DOS *cd* command.
- ➤ Compile the routine by typing: *aspcomp usrxls*
- ➤ If the routine is located in your work directory, the object file will be automatically included when the simulation is run.

Running the Excel Model

The simulation created in Chapter 1 can be used to run the Excel model. The Excel sheet can remain open when the simulation is run.

➤ Open the Excel workbook then open the Aspen Plus simulation.

Tip Excel may require a much longer time to open if the workbook is opened after the Aspen Plus simulation.

➤ Verify the name of the Excel workbook including the full path to the workbook on the User2 Subroutines form. The name specified in both the User2 Subroutines Model field and the subroutine statement for the interface routine (USRXLS) must be the same.

➤ Run the simulation and observe changes made to the Excel Spreadsheet

❖ ❖ ❖ ❖

3 New Integration Features for Proprietary Unit Operation Models

In addition to our new Excel Interface for User2 models, Release 10.1 also offers four powerful new features that improve the usability of Aspen Plus simulations containing proprietary models. It is now possible to create:

- Model Libraries containing categorized groups of models that can automatically be displayed with the Aspen Plus Model Library Menu,
- User-defined variables for proprietary models integrated with Aspen Plus instead of working with Real and Integer Arrays that were used in Chapter 1,
- Custom icons to better represent the equipment that the proprietary model represents, and
- Templates of standard and default input for proprietary models.

These features will be illustrated in this chapter by enhancing the Membrane Model that was developed in chapters 1 and 2. This session assumes that you have a working knowledge of building Aspen Plus simulations in Release 10, and have completed the *Aspen Plus Getting Started Building and Running a Process Model* Guide.

Allow 20 minutes for this session.

Creating a Model Library

Creating a model library allows work groups to create Model Menu tabs within the Aspen Plus User Interface for proprietary or third party Unit Operation Models. You can insert these models into an Aspen Plus simulation by dragging a user-designed graphical representation of that model from a user-defined Model Menu tab. You can create and display model variables such as Channel Hydraulic Diameter on the Input form and Permeate Protein Concentration on the Results form. Default input values for these variables can be stored in the library with the model.

You can configure a single user model differently, for example, by specifying different combinations of input and making each configuration available as separate icons. You can also insert each configuration into a simulation by dragging the appropriate icon into the simulation diagram.

To create a custom model library for the Membrane Model:

- ➤ Open the Membrane Simulation that you developed in Chapter 1.
- ➤ Select New from the Library menu to create a new user-defined model library.
- ➤ Name the library Membranes by entering **Membranes** into the Display Name field.
- ➤ Specify a filename with full path to store the library definitions. Select the Create button.
- ➤ From the Library menu, select Membranes then Edit.
- ➤ From the Categories menu, select New. Enter UltraFiltration as the category name. An UltraFiltration tab will be automatically inserted into the library.
- ➤ Select the UltraFiltration tab. Click the membrane model in the simulation diagram, right click and select **Add to Model Library** from the pop-up menu.
- ➤ Enter **HollowFiber** as the Model name, select the **Copy current icon only** radio button and select **Copy/create user model configuration**. Select OK.

Note Any Aspen Plus model from the Model Menu can be moved into the new library by selecting the icon and dragging the icon by holding down the left mouse button. Once the left mouse button is released over the new library, the Add to Library form is displayed. It is possible to select *Copy all icons* to insert all the icons associated with that model. You can modify these icons later.

➤ Update the library file by selecting **Membranes** from the Library menu then **Save**. Aspen Plus now creates the UltraFiltration tab in the Model Library Menu along with the other Aspen Plus models.

Creating and Configuring Variables

Once the model is part of a user-defined library, custom variables can be created to replace the real and integer parameters. These variables will be automatically displayed on the user-model configured variable form.

To create custom variables for the Membrane Model:

➤ Select the HollowFiber model in the Model Library Editor by clicking the icon associated with the model.

➤ From the Model menu, select **Edit User Configuration**. Edit User Configuration could also have been selected from the pop-up menu after right clicking the model.

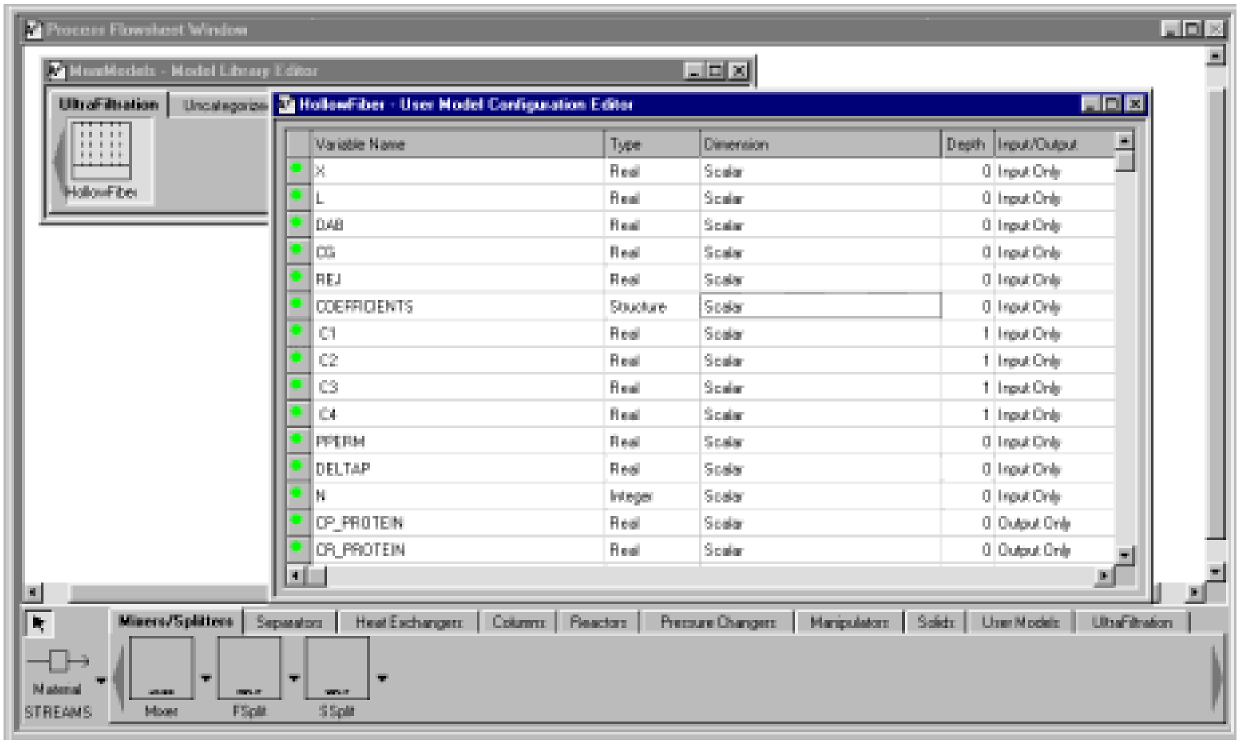
➤ Create the membrane variables for the real and integer parameters shown in Tables 2.2 and 2.3 using the Configuration Editor. The completed table should look like Figure 3.1 when complete.

➤ Change the type for the Coefficients variable to *Structure* to create an advanced object-like data construct. Change the depth for the C1, C2, C3, and C4 variables immediately below Coefficients to 1. This will cause these parameters to belong to the Coefficients structure. These parameters will be displayed on the User Model Configured Variable form as *Coefficients.C1, Coefficients.C2*, etc. Structures may be assigned to other structures by increasing the depth of subordinate structures.

Note The Configuration Variable Editor also enables the use of arrays based on nine common dimensions such as the *Number of Components* or *Number of Inlet Streams*, or can be dimensioned dynamically based on a defined integer variable. These dimensions can be selected from the Dimension drop-down list. If, for example, the variable *Ntubes* was previously defined in the configured variable list, another variable can be given a dimension of *Ntubes* by selecting *Ntubes* from the dimension drop-down list.

Variables can also be defined as combinations of Input and Output types. If a variable is marked as Input, it will only be displayed on the input form. Similarly, result type variables are only displayed on the results form. In this example, all variables are input only except the last two variables that will show the protein concentration in the permeate and retentate.

Figure 3.1 Configuration Variable Editor



	Variable Name	Type	Dimension	Depth	Input/Output
●	X	Real	Scalar	0	Input Only
●	L	Real	Scalar	0	Input Only
●	DAB	Real	Scalar	0	Input Only
●	CG	Real	Scalar	0	Input Only
●	REJ	Real	Scalar	0	Input Only
●	COEFFICIENTS	Structure	Scalar	0	Input Only
●	C1	Real	Scalar	1	Input Only
●	C2	Real	Scalar	1	Input Only
●	C3	Real	Scalar	1	Input Only
●	C4	Real	Scalar	1	Input Only
●	PPERM	Real	Scalar	0	Input Only
●	DELTAP	Real	Scalar	0	Input Only
●	N	Integer	Scalar	0	Input Only
●	CP_PROTEIN	Real	Scalar	0	Output Only
●	CR_PROTEIN	Real	Scalar	0	Output Only

- ➤ Before closing the Configuration Editor, save the newly created variables by selecting **Membranes** from the Library menu. Select the **Save** option. From the Fortran pull-down menu which is only active when the Configuration Editor is active, select the **Export** command from the Fortran menu. Export the definition file into your work directory.

- ➤ Start a simulation engine window from the Windows Start menu. Select Programs, AspenTech, Aspen Plus 10.1-0 then Aspen Plus Simulation Engine. Move to your work directory using the *cd* command. Compile the Fortran file that you just exported using the aspcomp command. If you called this file MemVars.f, type *aspcomp MemVars.*

➤ The configured variables will be displayed on the interface forms for any new blocks based on this model.

➤ Close the Model Library Editor and drag out a HollowFiber block.

The Configured Variable subform for the model is now active. This form displays the values for the configured variables that were previously entered as real and integer arrays. The protein concentration variables, Cp and Cr, will be displayed with the block results.

Creating a Custom Icon

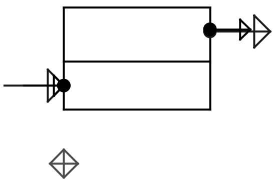
You can change or add new icons to better represent the equipment that the proprietary model represents. To change the Membrane Model icon:

➤ Open the Model Library Editor for the Membrane model by selecting **Membranes** from the Library menu then **Edit**.

➤ Select the HollowFiber model. Click the right mouse button, then select **Edit Current Icon**.

➤ Once in the editor, select the icon and press the delete key to remove.

➤ Draw the icon as shown below. Move the feed and product ports shown. Place the F1(IN) port directly over the blue universal feed port. Similarly, place the P1(OUT) port directly over the blue universal product port.



Note The small enclosed cross indicates the position where the block id will be placed.

➤ Exit the Icon Editor by closing the window. Select **Yes** to save new icon.

➤ Save the icon as part of the library by selecting Membranes from the Library then Save.

Tip Existing AutoCAD DXF files can also be imported into the Icon Editor. When the Icon Editor is active, a main Icon menu becomes available on the menu bar for the main window. Select *Import DXF* from the Icon menu.

Replacing the User2 Model

In this section, you will replace the User2 model with the new HollowFiber model to illustrate the use of the configured variables and custom icon. To do this:

➤ Close the Model Library Editor.

➤ Click the User2 model on the simulation diagram to select the icon then select delete to remove.

➤ Drag the new HollowFiber membrane from the UltraFiltration Model Library tab.

➤ Reconnect the feed and product streams reconnecting the Retentate stream before the Permeate stream.

➤ Specify the temperature and pressure flash option for the product streams.

Note All previously specified input is available. Observe that the Configured Variable form is now active and contains the Real and Integer array data that you specified.

➤ Reinitialize the simulation and run.

Save Simulation File

The simulation can be saved in either backup or document format. This simulation will be used in the next chapter.

➤ From the File menu, save and assign a name to the simulation using the Save As option.

Building a Custom Template

Adding a model to a library with previously specified parameters as you did at the start of this section is a new and powerful way to create model templates. Any model can be copied to a user-defined library with previously specified data for key or standard input. This capability can be used by workgroups to default input according to common engineering standards or previously configure advanced models to support different applications. For example, it is possible to apply a large dryer model in either a rotary kiln or tray type configuration. The model could be stored in the library twice with different values of configured variables to define each mode of operation. Each configuration can also have a separate icon enabling you to visually recognize the model according to its configuration.

The Version 10.1 framework permits hiding the built-in Aspen Plus Model Library so that only the custom libraries are visible. This enables one to completely replace the standard Aspen Plus library with Aspen Plus model templates containing default parameters and custom icons. For example, it is possible to store our RateFrac model twice, once as a Stripper and again as an Absorber. Each configuration would contain different values of key parameters.

❖ ❖ ❖ ❖

4 Creating an Aspen Plus Fortran Model

A new facility is available to provide read and write access to user-configured variables from Release 10.1 User Models. This facility was added to the existing Release 10.0 structure to simplify the mapping of user-defined parameters into Aspen Plus User routines. It is now possible to develop rich data structures for user-defined unit operation models involving integer, real and character data. The parameters can either be scalar or automatically dimensioned based on nine standard lengths such as *Number of Components* or by *Number of Inlet Streams*, or can be dynamically dimensioned based on a defined integer variable. For example, if the variable *Ntubes* was previously defined in the configured variable list, another variable can be given a dimension of *Ntubes*. Data structures can also have different depths. In this example, a structure was created called *Coefficients*. Variables C1 through C4 were assigned a depth of 1 since these variables belong to the Coefficients structure. Configured variables greatly enhance the usability of user-defined models particularly for large parameter sets.

In this session you will build a User2 model to replace the Excel model that you developed in Chapter 2. You will learn how to use the Aspen Plus Fortran Helper Functions to read and write the configured variables for the model. Reference will be made by variable name, instead of an index into the real or integer table to help prevent coding errors.

Allow about 40 minutes for this session.

This session assumes that you have a working knowledge of Fortran and have access to Digital's Visual Fortran Release 5.0C or higher.

Specifying the Fortran Model

A Fortran template, Usrus2.f, is available to simplify the specification process. The template contains the subroutine argument list, variable definitions and frequently used references to Aspen Plus labeled commons. Usrus2.f will be located in the Engine User subfolder with the other Fortran templates. The declaration section for the Usrus2.f file is shown in Figure 4.1.

Figure 4.1 User2 Model Declaration Section Template

```
C     User Unit Operation Model (or Report) Subroutine for USER2
C
      SUBROUTINE USRUS2 (NMATI,  SIN,    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   )
C
      IMPLICIT NONE
C
C     DECLARE VARIABLES USED IN DIMENSIONING
C
      INTEGER NMATI, NINFI, NMATO, NINFO, NTOT,
     +        NSUBS, NINT,  NPO,   NIWORK,NWORK,
     +        NSIZE
C
#include "ppexec_user.cmn"
      EQUIVALENCE (RMISS, USER_RUMISS)
      EQUIVALENCE (IMISS, USER_IUMISS)
C
#include "dms_ncomp.cmn"
C
C
C     DECLARE ARGUMENTS
C
      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),  NREAL, LD

      REAL*8 SIN(NTOT,NMATI), SINFI(NINFI),   SOUT(NTOT,NMATO),
     +       SINFO(NINFO),    WORK(NWORK),    SIZE(NSIZE)
C
C     DECLARE LOCAL VARIABLES
C
      INTEGER IMISS,       DMS_KFORMC
      REAL*8  REAL(NREAL), RMISS
```

For further information on the argument list variables, see the *Aspen Plus User Models Reference Manual. The ppexec_user.cmn* and *dms_ncomp.cmn* include statements support the use of key Aspen Plus variables in User routines such as *USER_NHSTRY* and *NCOMP_NCC. USER_NHSTRY* contains the Fortran output channel to the simulation history file and *NCOMP_NCC* holds the number of conventional components. For further information on other common variables, see the *Aspen Plus User Models Reference Manual.*

New Utility Functions

New utility functions are available to retrieve the values of user-defined configured variables created by the Configuration Editor. You will use the USRUTL_GET_REAL_PARAM utility to read real configured variable types.

In the membrane model, the specified diffusivity will be retrieved using the syntax *Ierr = Usrutl_Get_Real_Param('DAB', index, Dab).* Ierr is used as an error indicator. It will be nonzero if the variable could not be read. The first parameter *DAB* is the name assigned to the variable by the Configuration Editor. The last parameter *Dab* is the local Fortran variable that the value will be written into. The middle variable, *index*, defines the element that will be read for user-defined arrays.

For nested arrays, *index* is dimensioned by the configured variable's depth. For example, if an array has a depth of 1 preceded by another array in the Configuration Editor, the index will have the dimension *index(2).* If the dimensions for the first and second arrays are N and J, respectively, each element of the first array will have J elements of the second array. Index(1) will contain the index of the first dimension, and Index(2) will contain the index of the second dimension to completely define the element.

Only the variable name corresponding to the lowest level should be specified in the Helper function. For example, you created a Structure called Coefficients that contained the variables C1, C2, C3, and C4 each having a depth of 1. The variables are displayed on the Configured Variable form as Coefficients.C1, Coefficients.C2, etc. However, you would use the following syntax to retrieve the value of the C2 coefficient:

Ierr = Usrutl_Get_Real_Param('C2', index, C2).

Other useful functions are:

- *Usrutl_Set_Real_Param* to assign an Aspen Plus configured variable to the value of a local Fortran variable
- *Usrutl_CharToH* to convert a character string to hollerith format
- *Usrutl_Get_Int_Param* and *Usrutl_Set_Int_Param* to read and write integer parameters

Character variables can be read and written using *Usrutl_Get_Char_Param* and *Usrutl_Set_Char_Param*.

User2 Model

The User2 model is similar to the Excel interface file that you developed in Chapter 2. Instead of calling Excel from the interface routine, you will specify the governing equations directly. The routine could alternatively be developed using the User2 template file. To do this:

- ➤ Either start with the Excel Interface routine that was developed in Chapter 2 or copy the User2 template file, Usrus2.f, stored in the Engine User folder to your work directory.
- ➤ You can use Microsoft's Developer Studio to edit these files. Invoke it by double clicking the Fortran file.
- ➤ Append the local variable declaration section as shown in Figure 4.2 onto the template declaration section.

Figure 4.2 Additional User2 Declaration for Membrane

```
C     DECLARE LOCAL VARIABLES
C
      INTEGER OFFSET, IERR,  LDATA, KDIAG,  IDX(10), NCP,
     +        I,J,K,  LEN,   LMW,   Ntubes, IPerm,   IRet, ifail

      REAL*8  Hdia,  L,      Dab,  Cg,    Rej,  C1,  C2,  C3,  C4,
     +        Pperm, DeltaP, Rho,  Mu,    Fin,  Cin, Pin,
     +        Uave,  Re,     Sc,   Cp,    Cb,   Km,  Jm,
     +        Fp,    Pret,   Xmw,  X(10), Flow
C
C     DECLARE ASPEN PLUS HELPER AND FORTRAN FUNCTIONS THAT WILL BE USED
C
      INTEGER USRUTL_GET_REAL_PARAM, USRUTL_GET_INT_PARAM, DMS_IFCMNC
      REAL*8  DLOG
```

- ➤ Insert the Aspen Plus Helper function and Aspen Plus utility calls as shown in Figure 4.3. A status check is shown after each Helper function call. Errors should not occur once the call syntax is correct and the Configured Variable Definition file has been compiled. The DMS_IFCMNC function is used to find the location of the molecular weight table in the Aspen Plus Plex data structure. The PPMON_VISCL routine is called to calculate the liquid viscosity of the feed. All errors are processed before returning from the routine.

Figure 4.3 Helper Functions and Utilities

```
C     Obtain offset to molecular weight table
      LMW= DMS_IFCMNC('MW')
      ifail=0

C     Read in User Defined Configured Variables using Helper Functions
      Ierr= Usrutl_Get_Real_Param('X', K, Hdia)
             if(ierr .ne. 0) then
               Write(USER_NHSTRY,*)' Error fetching Hydraulic Diameter'
                ifail=1
              endif

      Ierr= Usrutl_Get_Real_Param('L', K, L)
             if(ierr .ne. 0) then
               Write(USER_NHSTRY,*) ' Error fetching Length '
                ifail=1
              endif

      Ierr= Usrutl_Get_Real_Param('DAB', K, Dab)
               if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ' Error fetching Diffusivity '
                 ifail=1
              endif

      Ierr= Usrutl_Get_Real_Param('CG', K, Cg)
              if(ierr .ne. 0) then
               Write(USER_NHSTRY,*) ' Error fetching Cg '
                ifail=1
              endif

      Ierr= Usrutl_Get_Real_Param('R', K, Rej)
               if(ierr .ne. 0) then
               Write(USER_NHSTRY,*) ' Error fetching Rejection Factor '
                ifail=1
               endif

      Ierr= Usrutl_Get_Real_Param('C1', K, C1)
               if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ' Error fetching C1 '
                 ifail=1
               endif

      Ierr= Usrutl_Get_Real_Param('C2', K, C2)
               if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ' Error fetching C2 '
                 ifail=1
               endif

      Ierr= Usrutl_Get_Real_Param('C3', K, C3)
               if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ' Error fetching C3 '
                 ifail=1
               endif

      Ierr= Usrutl_Get_Real_Param('C4', K, C4)
               if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ' Error fetching C4 '
                 ifail=1
               endif

      Ierr= Usrutl_Get_Real_Param('PPERM', K, Pperm)
               if(ierr .ne. 0) then
                Write(USER_NHSTRY,*)' Error fetching Permeation Press'
                 ifail=1
               endif
```

Figure 4.3 (continued) Helper Functions and Utilities

```
      Ierr= Usrutl_Get_Real_Param('DELTAP', K, DeltaP)
               if(ierr .ne. 0) then
                 Write(USER_NHSTRY,*) ' Error fetching delta P '
                  ifail=1
              endif

       Ierr= Usrutl_Get_Int_Param('N', K, Ntubes)
               if(ierr .ne. 0) then
                 Write(USER_NHSTRY,*) ' Error fetching Number of Tubes '
                  ifail=1
              endif

C     Determine viscosity for stream
      CALL SHS_CPACK(MSIN(1,1), NCP, IDX, X, FLOW)
      KDIAG=4
      CALL PPMON_VISCL(MSIN(NCOMP_NCC+2,1),MSIN(NCOMP_NCC+3,1),X,NCP,
     +                 IDX,NBOPST,KDIAG,Mu,IERR)
           if(ierr .ne. 0) then
             Write(USER_NHSTRY,*) ' Error Evaluating Viscosity for Feed'
              ifail=1
           endif

      if(ifail .eq. 1) return
```

- ➤ Now, insert the governing equations as shown in Figure 4.4. Aspen Plus copies the results back into the material product outlet vector SOUT on a molar basis using SI units. Before saving the results into SOUT, the Iperm and Iret indices are defined for the Permeate and Retentate streams.

- ➤ The two result scalars to display the protein permeate and retentate concentrations, are passed back to Aspen Plus using the Usrutl_Set_Real_Param utility function.

Figure 4.4 Governing Equations

```
C     Calculate adjusted input variables

      Rho =     MSIN(K+8,1)/1000.
      Mu =      Mu*98.
      Fin =     MSIN(K+1,1)*MSIN(K+9,1)/Rho*3600.
      Cin =     MSIN(2,1)*B(LMW+2)/Fin*3600000.
      Pin =     MSIN(K+3, 1)/101325.

C     Calculate Intermediate Variables

      Uave =    Fin/(Hdia*L*3.14*Ntubes)*1000/3600.
      Re =      Hdia*Uave*rho/Mu
      Sc =      Mu/(Rho*Dab)

C     Mass Transfer Equations

      Cp =      Cg*(1-Rej)
      Cb =      (Cin - Fp/Fin*Cp)/(1- Fp/Fin)
      Km =      C1*(Re**C2) * Sc**C3 * (Hdia/L)**C4 * Dab/HDia
      Jm =      km* DLOG((Cg - Cp)/(Cin - Cp))
      Fp =      Jm*HDia*L*3.14*Ntubes*360.
      Pret =    (Pin - deltaP)*101325

C     Identify which product stream is listed first in outlet vector
C     Assume Permeate stream is first, switch if not.

      IPerm = 1
      IRet =  2
```

Figure 4.4 (continued) Governing Equations

```
      If(IDSMO(1,1) .eq. ’RETE’) then
         IPerm= 2
         IRet=  1
      EndIf

C     Copy results into outlet stream vectors
C     User block should be specified to perform P and T flash

      SOUT(1,IRet)= Fp*Rho/3600./B(LMW+1)
      SOUT(2,IRet)= Fp*Cp/1000./3600./B(LMW+2)
      SOUT(K+1,IRet)= SOUT(1,IRet) + SOUT(2,IRet)
      SOUT(K+2,IRet)= MSIN(K+2,1)
      SOUT(K+3, IRet)= Pret

      SOUT(1,IPerm)= MSIN(1,1) - SOUT(1,IRet)
      SOUT(2,IPerm)= MSIN(2,1) - SOUT(2,IRet)
      SOUT(K+1,IPerm)= SOUT(1,IPerm) + SOUT(2,IPerm)
      SOUT(K+2,IPerm)= MSIN(K+2,1)
      SOUT(K+3,IPerm)= Pperm*101325

      Ierr= Usrutl_Set_Real_Param(’CP_PROTEIN’, K, Cp)
        if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ’ Error storing permeate concentration ’
                 ifail=1
       endif

      Ierr= Usrutl_Set_Real_Param(’CR_PROTEIN’, K, Cr)
        if(ierr .ne. 0) then
                Write(USER_NHSTRY,*) ’ Error storing retentate concentration’
                 ifail=1
       endif


      RETURN
      END
```

➤ Save the file in your work directory using the *Save As* option from the File menu.

➤ Exit the development environment.

Export the Variable Configuration File

You must export the definition file from the User Model Configuration Editor to enable reference to the configured variables within the external User2 routine if you have not already done so. Please see the *Creating and Configuring Variables* section of Chapter 3 for a description on how to save and compile the configured variable definition file.

Compile User2 Routine

Now that the User2 routine is complete, it must be compiled using Digital's Visual Fortran Version 5.0C or higher before it can be linked with your simulation. You must activate the compiler from the Simulation Engine Window.

➤ Activate the Simulation Engine Window from the Windows Start menu by selecting AspenTech from Programs, then Aspen Plus 10.1-0, then Aspen Plus Simulation Engine.

➤ Move to your work directory.

➤ Compile the Fortran file using the Aspen Plus aspcomp procedure. Type aspcomp followed by the Fortran file root name. Both the User2 routine and configured variable definition file must be compiled.

Running the Fortran Model

You are now ready to apply the User2 routine. The only remaining specification is to verify the subroutine name associated with the model and to remove the reference to the Excel spreadsheet. To do this:

➤ Open the HollowFiber block and declare the Fortran subroutine name on the Subroutines form. Remove the Excel Workbook name.

➤ Run the simulation.

➤ Observe that only the last two real parameters are displayed in the block results.

❖ ❖ ❖ ❖

5 Creating GUI Forms for Aspen Plus Proprietary Models

Creating graphical forms using Microsoft Visual Basic is the last level of customization for supporting proprietary or third party models in Aspen Plus. Configured variables that were defined for these models can be linked into these forms. Forms can be created with a free form layout to suit your modeling needs.

The ability to create forms combined with the templating capabilities that were outlined in Chapter 3 is a very powerful customization capability. It is now possible to create forms for an existing plant unit that hides design parameters for users interested in performing rating only calculations for example. These forms are easily inserted into Aspen Plus by registering the OCX file created by Visual Basic using the Aspen Plus *ApwnSetup* utility. Visual Basic template files are provided with Aspen Plus to reduce the amount of time required to develop custom forms. Fields commonly used by the Aspen Plus GUI can be applied to a new form by dragging it from a Controls Toolbox.

In this session, you will learn how to create an Input and Result form for the Membrane Model that you created in Chapter 4. You will create fields for Channel Hydraulic Diameter and Channel Length on the Input form. Two result fields will be created for displaying the concentration of protein in the permeate and retentate streams.

This session assumes you have a working knowledge of Microsoft Visual Basic and have access to version 5.

Allow 40 minutes for this session.

Defining the Visual Basic Environment

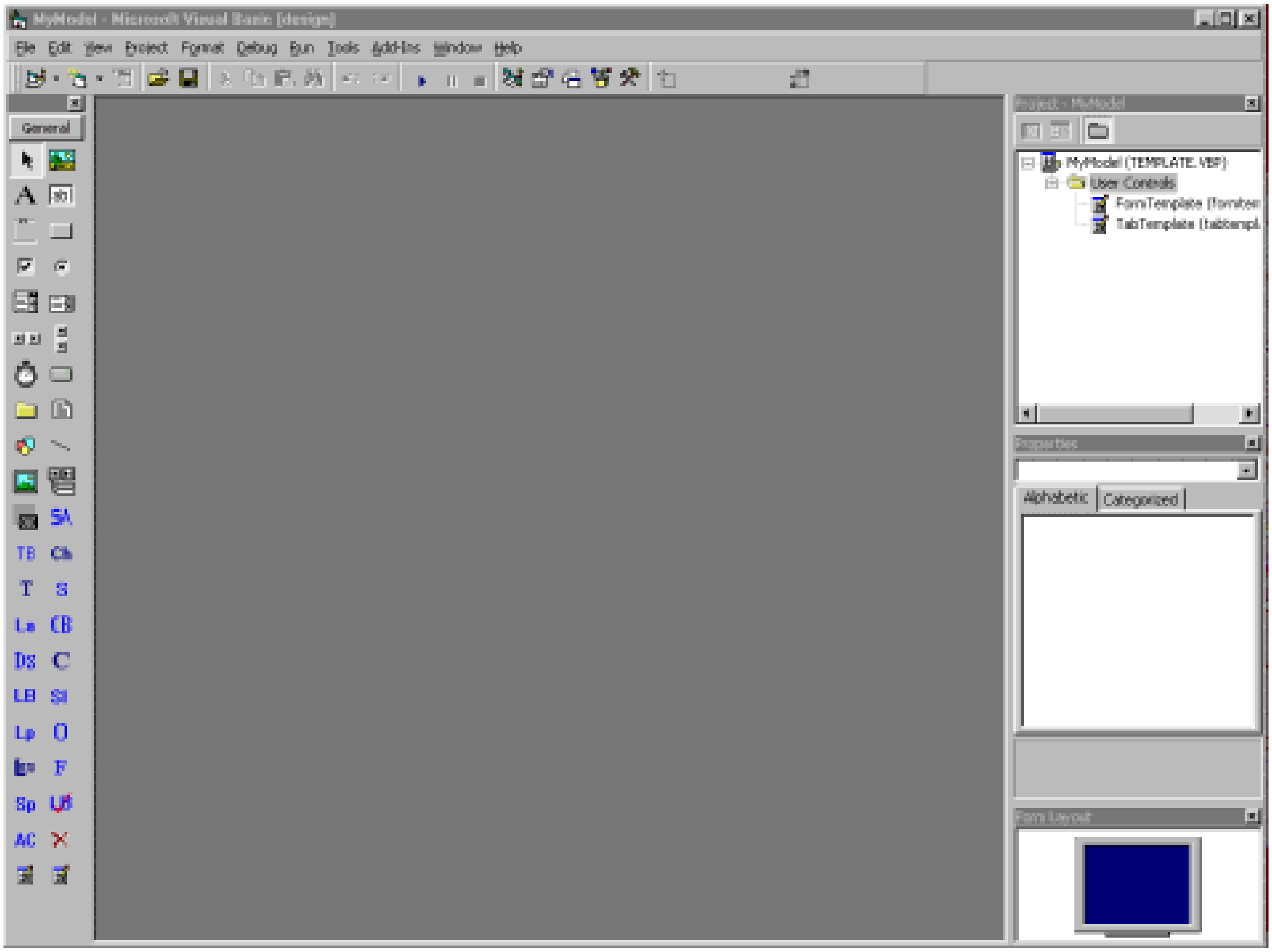
Aspen Plus provides a Visual Basic project and two Visual Basic control files to define the necessary development environment for creating Aspen Plus GUI forms. Every GUI form contains a Tab Control to select subforms. In this example, an Input and Result subform will be created. Each subform can be accessed by clicking the associated tab.

To set up the Visual Basic environment:

- ➤ Create a new folder that you will use as a Visual Basic Work directory.
- ➤ Copy the Visual Basic Project template file, template.vbp, from the GUI xeq subfolder to your new work directory.
- ➤ Open the Visual Basic project by double clicking template.vbp.
- ➤ From the Project menu, select Add User Control and import the **formtemplate** and **tabtemplate** controls located in the GUI xeq subfolder. You must select the Existing tab first to locate these files via a File Open dialog box.

The controls used to create common Aspen Plus fields should now be visible. You can display a short descriptor for each Toolbox control by placing the mouse pointer over the control in the Toolbox. All controls prefixed with *MM* are Aspen Plus controls that are delivered with Version 10.1. There should be 19 of these controls located from the middle to the end of the Toolbox as shown in Figure 5.1. The tab and form control will be visible in the pane located in the upper right hand corner.

Figure 5.1 Visual Basic Aspen Plus Forms Environment



Creating Forms for Membrane Model

In this section, you will create an Input and Result form each containing two variables along with a Parent tab that you will use to access the forms within the GUI. To do this:

➤ Select the tab template control and rename it to Mem_HFiberTab. Save this form as Mem_HFiberTab.ctl in your work directory.

➤ Select the form template control and rename it to Mem_HFiberI that will later become our input form. Save this form as Mem_HFiberI.ctl in your work directory.

- ➤ Rename the project to HollowF. You will store a copy of the HollowFiber model in Aspen Plus as HollowF later in this chapter for testing the new forms. We recommend using the same name for the Visual Basic Project and Aspen Plus model.

- ➤ Similarly, import another form control by adding another instance of the **form template** control. Rename this control to Mem_HFiberR that will later become the Results form. Save this control as Mem_HFiberR.ctl in your work directory.

Note The form template used for Mem_HFiberI must be renamed and saved as Mem_HFiberI.ctl before a new instance of this control can be imported.

Building the Tab Control

You use the Tab Control to select both the Input and Results subform within the Aspen Plus GUI. Set the captions for the tab strip and link this strip to the Input and Output forms defined for the HollowF model:

- ➤ Drag the MMTabStrip control from the Control Toolbox and place it on the tab form.

- ➤ Change the caption for the first tab to **Input** by selecting the tab strip, then click with the right mouse button and select the Properties option. Enter **HollowF.Mem_HFiberI** into the Form Name field to link the Input tab to the Input form and the Aspen Plus model.

- ➤ Advance the tab from 0 to 1 displayed in the Current Tab field.

- ➤ Change the caption for the Results tab to **Results** and specify **HollowF.Mem_HfiberR** for the FormName.

- ➤ Advance the Current Tab to 2 and select the Remove Tab button to delete. Only two tabs are needed, 0 and 1, for input and results.

Building the Input Control

You will map the Channel Hydraulic Diameter and the Channel Length variables created by the Aspen Plus configuration editor to the Input Control. To do this:

- ➤ Drag the Frame control onto the Input Control and size it by stretching it with the mouse.
- ➤ Change the caption to **Membrane Parameters**.
- ➤ Drag the MMLBUText control and size it to fit within the frame. This control provides a label, data field, basis for converting between input types such as mole or mass, and units.
- ➤ Select the new control, then click the right mouse button for properties.
- ➤ On the MMLBTUText tab, deselect the Basis and Units field types.
- ➤ Set the width for the Label to 2350. Set the Text Left parameter to 2400.
- ➤ On the MMLBTUTextLabel tab, enter Channel Hydraulic Diameter into the Label Caption field.
- ➤ On the Variable tab, select the Add button to create a variable having a **Node** datatype. In the first Path column, enter main, Input, User Table followed by X. A space must be entered between the words User and Table.
- ➤ Similarly, create a field for the Channel Length that will be assigned to the Aspen Plus main, Input, User Table, L. data path.

Tip The data path can be determined in Aspen Plus by selecting the Variable Explorer option from the Tools menu and navigating through the folders. The path down to and including the block of interest will be assigned to the variable main.

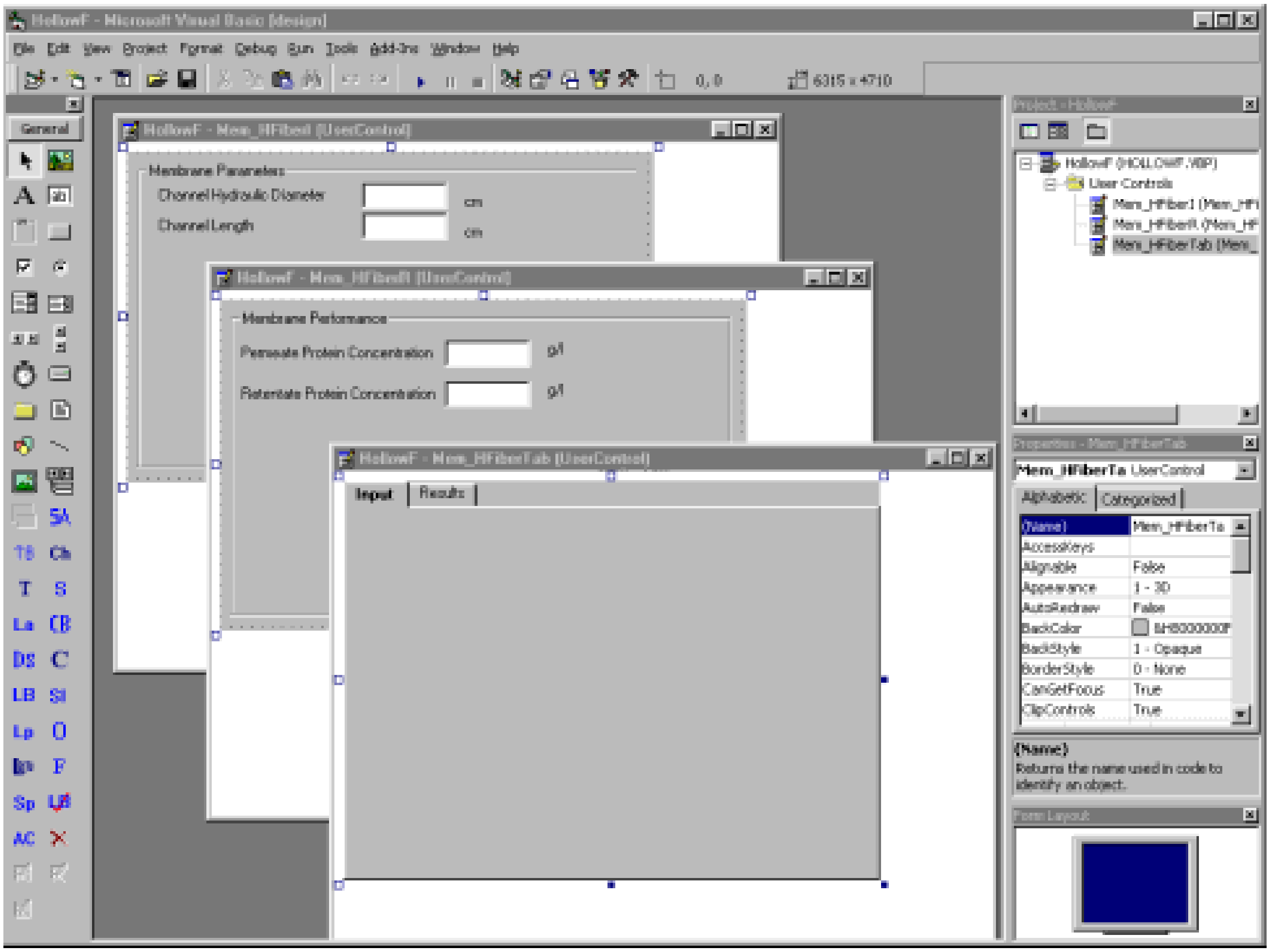
- ➤ Drag out the **A** label control and specify **cm** for the caption.

Building the Result Control

The protein concentration for both the permeate and retentate streams can be mapped to the Result Control using the same procedure as outlined for the Input Control. Map the variables to main, Output, User Tree, CP_PROTEIN for the protein permeate concentration, and main, Output, User Tree, CR_PROTEIN for the protein retentate concentration. Add the concentration units using the **A** label control.

The finished forms should appear as shown in Figure 5.2.

Figure 5.2 Completed Membrane Model GUI Forms


HollowF - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Tools Add-Ins Window Help
0, 0
6315 x 4710
General
HollowF - Mem_HFiberI (UserControl)
HollowF - Mem_HFiberR (UserControl)
HollowF - Mem_HFiberTab (UserControl)
Input
Results
Project - HollowF
HollowF (HOLLOWF.VBP)
User Controls
Mem_HFiberI (Mem_HFi
Mem_HFiberR (Mem_HF
Mem_HFiberTab (Mem_
Properties - Mem_HFiberTab
Alphabetic
Categorized
Form Layout

Membrane Parameters

Channel Hydraulic Diameter □ cm

Channel Length □ cm

Membrane Performance

Permeate Protein Concentration □ g/l

Retentate Protein Concentration □ g/l

Mem_HFiberTa UserControl

(Name)	Mem_HFiberTa
AccessKeys	
Alignable	False
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F
BackStyle	1 - Opaque
BorderStyle	0 - None
CanGetFocus	True
ClipControls	True

(Name)
Returns the name used in code to identify an object.

Integrating Forms with Aspen Plus

The Input and Output forms can be easily integrated with the Aspen Plus GUI by registering an OCX file with the operating system. To integrate these forms:

- ➤ Save the Input, Output and Tab control individually by selecting the save option via a right mouse click from the project window. Save the Visual Basic project by selecting the Save option from the File menu.

- ➤ Select *Make ocx…*from the File menu. Save the OCX file in your work directory as HollowF.ocx.

Note The Aspen Plus model name should be used for both the ocx and ocr root names.

- ➤ Create an OCR file using a text editor. Template.ocr, shown in Figure 5.3, can be copied into your work directory from the GUI xeq subfolder to speed the specification process.

Figure 5.3 OCR Template File

```
Blocks\MYMODEL, MyModel.MyModel_Input, Inputs, 0, 1
Blocks\MYMODEL, MyModel.MyModel_Input, Inputs, Default, 1
Blocks\MYMODEL, MyModel.MyModel_Result, Results, 41, 1, ResultsOnly
Blocks\MYMODEL, MyModel.MyModel_Result, Results, ResDefault, 1
```

The name following “Blocks\” is the name of the Aspen Plus model that will own the forms. The second parameter declares the name of the ocx file that the ocr file describes followed by the name of the tab form separated by a period. The third parameter, Inputs or Results, will be the name of the tab form displayed in the data browser underneath the block folder. The fourth parameter is the Data Browser index of the form. Inputs will be listed in the Data Browser before Results.

The first occurrence of the tab is used to register the tab control. The second line defines the Tab that will be used by default. Both an input and result default are supported. The default is automatically displayed first when the Next key is pressed or displayed after a simulation block is double clicked.

The contents of the completed OCR file for this example are shown in Figure 5.4. Only one tab will be registered that will contain two subforms, one for input and one for results.

Figure 5.4 OCR file for HollowF Model

```
Blocks\HOLLOWF, HollowF.Mem_HFiberTab, HFib, 0, 1
Blocks\HOLLOWF, HollowF.Mem_HFiberTab, HFib, Default, 1
```

➤ Copy the OCR and OCX files to the GUI Forms subfolder.

➤ Invoke the ApwnSetup procedure in the xeq subfolder of the GUI installation folder.

➤ On the first ApwnSetup form, define the path for the GUI xeq and Forms subdirectories then select the Next button. The xeq and Forms directories are installed directly below the GUI directory by default.

➤ Press the Deselect button to deselect all controls, select only the HollowF control, then choose the **Update Data Browser** option instead of the Rebuild Data Browser option. Select Next.

➤ Press the Deselect button to deselect all plot wizard files then select Finish.

Creating HollowF Model

Open the simulation that was developed in Chapter 4. A HollowF model will be created from the HollowFiber unit operation block to test the new forms. Maintaining production and development versions is recommended when not all input variables are mapped into the interface.

Usually, these models are stored in two separate libraries where only the production library is given to the user. In this example, both models will be stored in the same library.

➤ Open the simulation developed in Chapter 4.

➤ Open the MemModels library for write by selecting MemModels from the Library menu then Edit. Select MemModels from the Library menu again, then select Write.

➤ Open the membrane block in the simulation and delete the product flash specifications. Since this form is not available to our new HollowF model, these specifications must be removed before creating a template from this block for the HollowF model. The HollowF model will have an Incomplete Input status if this specification is not deleted before creating the model.

Note To complete an energy balance around the block, a flash monitor call must be added to either the Usrxls Excel interface file or the User2 model. Since the purpose of this example is to only illustrate how to integrate custom forms within Aspen Plus, we will not add the flash call to this example. The model will run without performing an energy balance. See *Chapter 2 of the User Models Manual* for details on calling our flash monitor.

➤ Select the hollow fiber block in the simulation and select the *Add to Model Library* option from the right mouse pop-up menu.

➤ Copy over the current icon only and name the model HollowF.

➤ Save the changes by selecting MemModels from the Library menu then select Save.

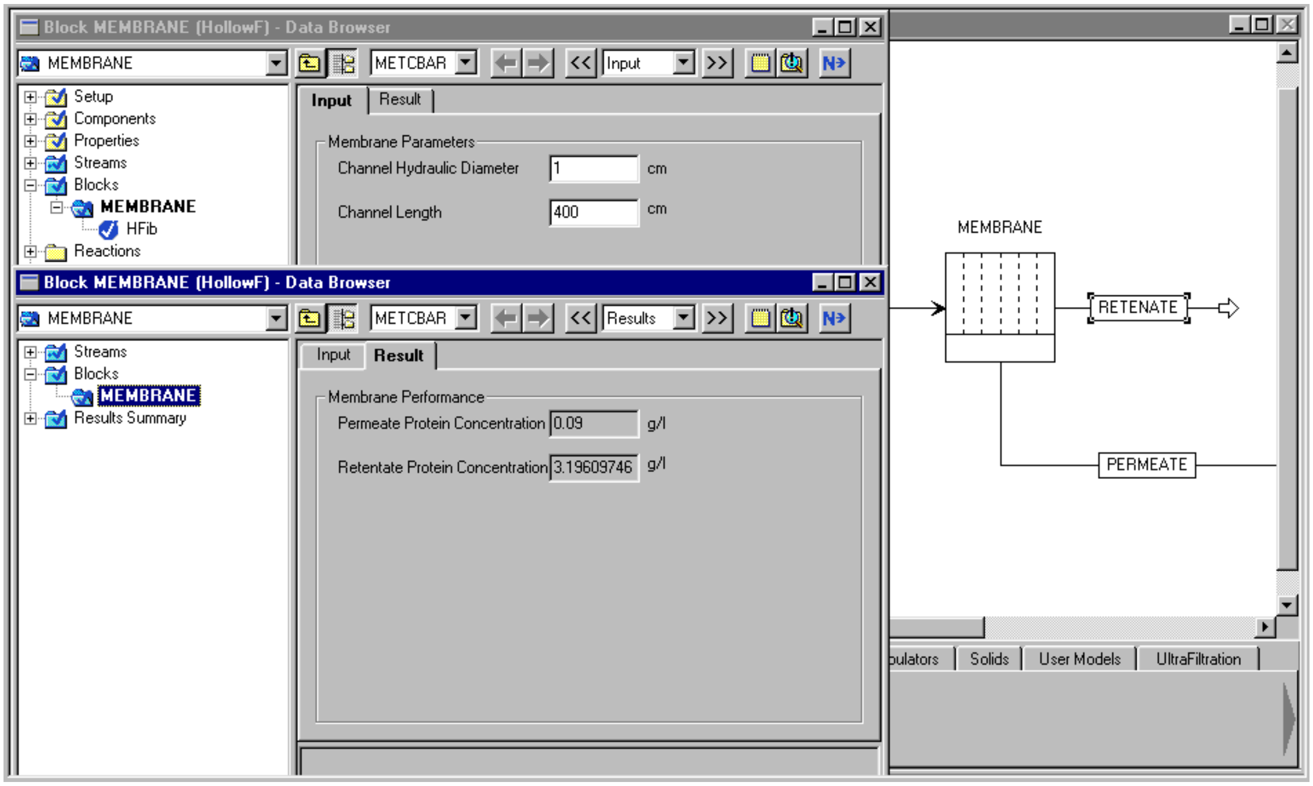
➤ Exit the library editor.

➤ Drag out the HollowF model, and select Input. The newly created forms will now load.

➤ The HollowFiber unit operation block can be replaced with HollowF to test the forms.

The application should appear as shown in Figure 5.5. Notice the *HFib* Data Browser folder that was defined in the OCR file and the input and result variables accessed via their respective tabs.

Figure 5.5 Membrane Forms Integrated with GUI



Membrane Parameters

Channel Hydraulic Diameter	1	cm
Channel Length	400	cm

Membrane Performance

Permeate Protein Concentration	0.09	g/l
Retentate Protein Concentration	3.19609746	g/l

Exiting Aspen Plus

To exit from Aspen Plus:

➤ From the File menu, select Exit.

➤ When the dialog box appears asking if you want to save the run, select No to exit without saving the run;
– or –
select Yes if you want to save the run, and enter a Run ID when prompted.

❖ ❖ ❖ ❖

A Connecting to the Aspen Plus Simulation Engine

If either of the following conditions exist, you will be prompted to specify the host computer for the Aspen Plus simulation engine after you start the Aspen Plus User Interface:

- The simulation engine is not installed on your PC.
- The simulation engine is installed on your PC, but the Activator security device is not connected to your PC.

In these cases, the Connect to Engine dialog box appears.

➤ Click the Server Type list box and select the type of host computer for the simulation engine.

If you choose Local PC as the server for the simulation engine, you do not need to enter any more information into the dialog box.

➤ Click OK to continue.

If you choose UNIX host, OpenVMS host, or Windows NT server as the server for the simulation engine:

➤ Enter the node name of the computer on which the Aspen Plus simulation engine will execute.

➤ Enter your User Name, Password, and Working Directory, and click OK.

When the network connection is established, a message box appears saying Connection Established.

If the Connection Established message does not appear, see your Aspen Plus system administrator for more information on network protocols and host computers for the Aspen Plus simulation engine.

❖ ❖ ❖ ❖