

Contents

Part 1 Getting Started

- Chapter 1** Introducing *Limn*
- Chapter 2** Installing and starting *Limn*
- Chapter 3** Drawing a *Limn* flowsheet
- Chapter 4** A *Limn* workbook
- Chapter 5** Solving the flowsheet

Part 2 Examples

- Chapter 1** Introduction to Examples
- Chapter 2** DEWATER.XLS - Simple flowsheet balance
- Chapter 3** 2CYCS.XLS - Data block formatting
- Chapter 4** CRUSHING.XLS - Size distributions
- Chapter 5** FLOTATN.XLS - Assays
- Chapter 6** EFFCURVE.XLS - Modelling options

Part 3 *Limn* Wizards

- Chapter 1** Introduction
- Chapter 2** The *Limn* DataWizard

Part 4 *Limn* Reference

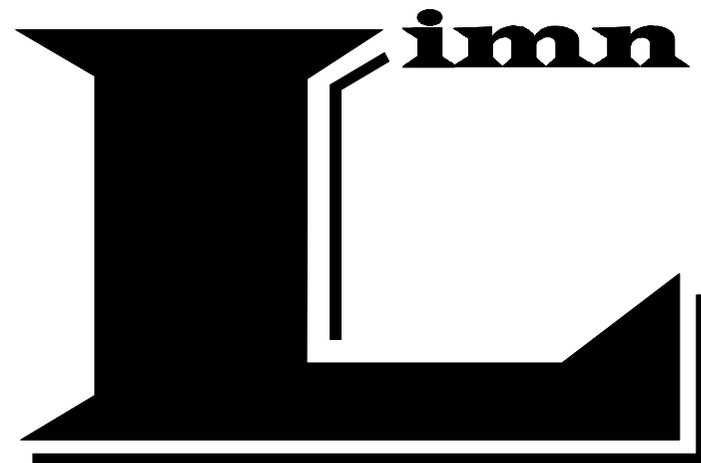
Replaced by On-line help

Part 5 *Limn* Technical Reference

- Chapter 1** Creating User-defined icons
- Chapter 2** LIMN.INI - Initialisation file
- Chapter 3** Excel Names used in *Limn*

Part 1

Getting Started



“The Flowsheet Processor”

Release 2.10 - September 2002

Copyright David Wiseman Pty Ltd 1994 - 2002

Part 1 Getting Started

Chapter 1 Introducing *Limn*

*What is **Limn*** 4

Conventions used in this manual 6

Chapter 2 Installing and starting *Limn*

System requirements 8

*Installing **Limn*** 8

*Loading the **Limn** Excel Add-Ins* 10

*Starting **Limn*** 12

Chapter 3 Drawing a *Limn* flowsheet

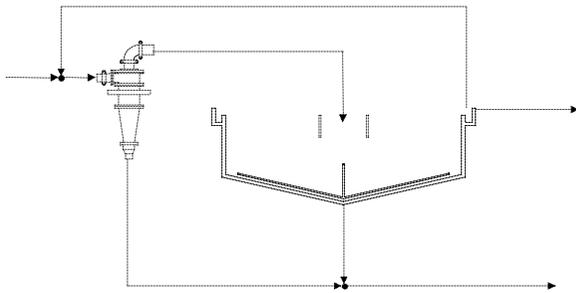
*Drawing a **Limn** Flowsheet* 14

Chapter 4 A *Limn* workbook

*A **Limn** workbook* 30

Chapter 5 Solving the flowsheet

Solving the flowsheet 48



What is **Limn**

Limn¹ is a “Flowsheet Processor”, designed to bring the same ease of use to flowsheet drawing and solution that word processors bring to the preparation of text based documents.

Process oriented engineers now frequently use spreadsheets to perform the data presentation and calculation tasks associated with their jobs. **Limn** functions as an add-in module to the Microsoft[®] Windows[™] Excel spreadsheet, to enhance the value of spreadsheets in this role by providing a comprehensive set of flowsheet specific drawing tools, within the familiar spreadsheet environment. Once drawn, the flowsheet may be coupled with **Limn's** general purpose flowsheet solution engine. This approach removes many of the difficulties associated with preparation of flowsheet balances using spreadsheets.

By building on an existing and widely used tool such as Excel, **Limn** is able to make full use of the features of the Windows[™] operating environment. This includes the ability to incorporate **Limn** flowsheets within the documents of other applications such as word processors. **Limn** follows the Windows[™] user interface standards, allowing users to apply existing skills to this package.

Because **Limn** is a general purpose and “open” package (for example, all data kept within standard Excel sheets), it can also function as a toolkit from which more specific packages may be built. In addition to the spreadsheet modelling capability, facilities are provided to incorporate user models, which may be written in any language capable of producing Windows[™] Dynamic Link Libraries.

¹**limn** (lim), v.t. **1.** to represent in drawing or painting. **2.** to illuminate.



Tip: Watch this space!

Shortcuts, hints, and tips will appear in boxes such as this on the left page.

Conventions used in this manual

Layout

This user manual is arranged with the bulk of the text on the right hand page of each pair of open pages. The left hand page is used for supplementary information (often additional hints and tips), as well as providing space for user notes and annotation.

Text formatting

Special fonts and typefaces are used in the text to indicate the following:

Bold type	Words or characters that you type. Also used to highlight Windows™ menu items that you must access.
<i>Italic type</i>	Place-holders for items you must supply (for example, directory or file names)
Capital letters	Keys or combinations of keys (for example, CTRL+1, or ENTER, or ESC)

Mouse buttons

For ease of reading, the text always refers to a mouse as the default pointing device, with the left button defined as the primary button. If you are using another form of pointing device (for example, a track ball), or have changed the default mouse buttons, any references to the mouse left button should be read as a reference to the device primary button as defined in the Windows™ Control Panel.

Important information

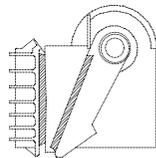
Is highlighted by an exclamation mark in the margin.





Tip: *Limn* is part of Excel!!

Remember that ***Limn*** is meant to enhance the capabilities of Excel, not to re-invent the wheel. If you cannot find a required function in the ***Limn***.Draw module, it is likely that it already exists as a native Excel function. Simply exit from ***Limn***, and carry out the required action.



System requirements

Software

Limn is designed to operate with Microsoft® Windows™ Excel. The software has been tested using Excel versions 5, 7, 97 and 2000 under Windows™ 3.1, Windows™ for Workgroups 3.11, Windows95™, Windows98™ and WindowsNT™ 4.0

Verify that Microsoft® Windows™ and Excel are installed and operating correctly before attempting to install **Limn**.

Hardware

Any system that is capable of running Excel in an adequate manner will be suitable for **Limn**.

As a guide, a Pentium 100 is probably the minimum comfortable configuration for Excel versions 5 and 7, with a Pentium 200 the minimum comfortable configuration for later versions of Excel.

Limn requires approximately 6MBytes of hard disk space.

Installing **Limn**

Limn is distributed on CDROM with updates on 1.44MByte disks.

Place the CDROM into the CDROM drive. If Autoplay is enabled, setup will start automatically. Otherwise, run program **SETUP.EXE** from the CDROM.

Where update disks are provided, load **Limn** from the CDROM first, then open **SETUP.HTM** on the update disk, and follow the instructions shown there.



Tip: Excel Add-ins - Load as required.

It is possible to use the Excel Add-In Manager (menu Tools, option Add-Ins...) to load the **Limn** and **Limn DataWizard** Add-Ins. [For details, refer to Excel on-line Help, Search for 'Add-ins Command (Tools menu)'].

Add-Ins loaded in this way are automatically loaded each time Excel is run. Add-Ins accessed from a toolbar button are only loaded when required.

Choosing the toolbar method of loading the **Limn** Add-Ins as detailed opposite will result in faster loading of Excel when it is required for normal spreadsheet use.

The Auto-Configure option sets up Excel for this toolbar load method.

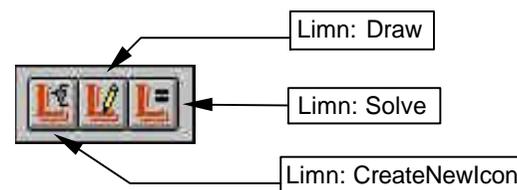
Loading the *Limn* Excel Add-Ins

If Auto-Configure was chosen during setup, the *Limn* and *Limn DataWizard* toolbars should be present and ready for use.

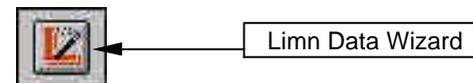
If you did not select Auto-Configure on the setup dialog, you will need to manually configure the Excel toolbars to allow access to *Limn*. To do this:

1. Run the *Limn* setup routine in the normal manner.
2. Run Excel.
3. Using the Open command on the File menu, find and open the files LIMN.XLA and LIMNWIZD.XLA.
4. Close Excel.

There should now be a new *Limn* toolbar with three buttons present in the Excel toolbar area.



A new *Limn DataWizard* toolbar should also be present.



It is usually convenient to place the two toolbars next to one another:





Tip: VBA access to *Limn:Solve*

It is possible to call the ***Limn:Solve*** module from the VBA macro language within Excel. This can be useful for optimisation studies, or automated examination of a range of scenarios. For more information on extending ***Limn***, see Part 4 of this manual - "The ***Limn*** Technical Reference".

Starting *Limn*

The *Limn* toolbar and the *Limn DataWizard* toolbar the four major components of the *Limn* package:

Limn: Draw

Limn: Draw provides the enhanced tools for drawing detailed process flowsheets. To start the *Limn: Draw* module, click the mouse left button on the *Limn: Draw* toolbar button. For more information on using the *Limn: Draw* module, see Chapter 3 of Part 1 of this manual.

Limn: Solve

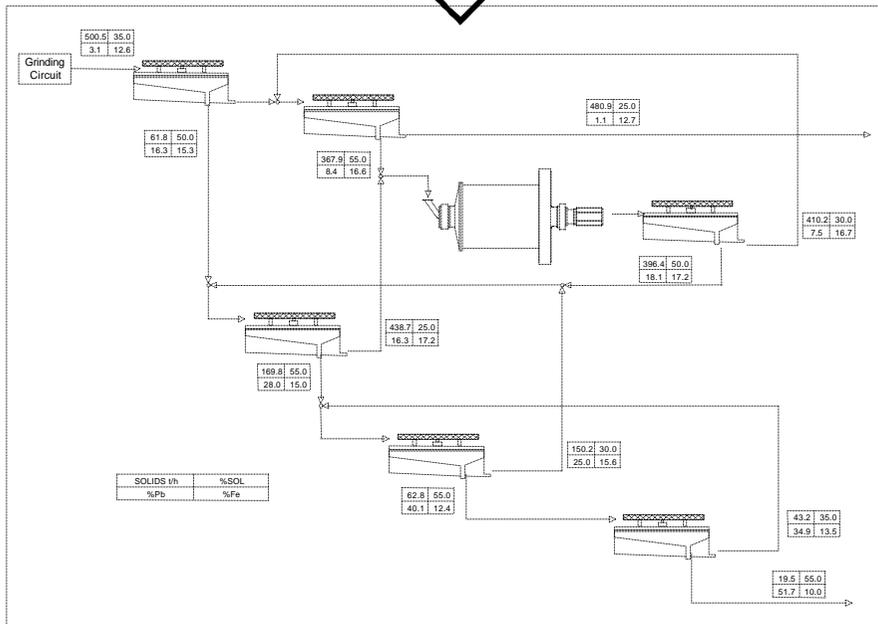
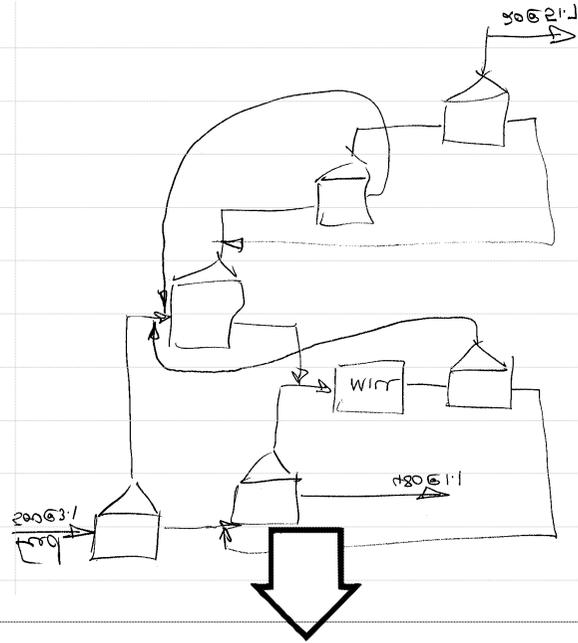
Once the flowsheet is defined using *Limn: Draw*, and data and process models specified in Excel, *Limn: Solve* may be used to calculate the flowsheet balance or solution. The *Limn: Solve* module is started by clicking the *Limn: Solve* toolbar. For more information on using the *Limn: Solve* module, see Chapter 5 of Part 1 of this manual.

Limn: CreateNewIcon

Limn: CreateNewIcon is used to incorporate user site specific artwork into the *Limn* package as process icons. The module is accessed through the *Limn: CreateNewIcon* toolbar button. For more information on using the *Limn: CreateNewIcon* module, see Part 4 of this manual - “*Limn* Technical Reference”.

Limn Data Wizard

The *Limn DataWizard* automates several of the data entry steps associated with setting up a flowsheet solution in *Limn*. The ‘wizard’ (series of step by step dialogs) is accessed through the button on the *Limn DataWizard* toolbar. More information may be found in Part 3 of this manual.



Drawing a *Limn* Flowsheet

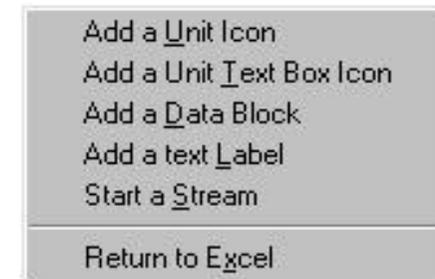
The following describes the steps involved in drawing a simple water recovery circuit containing two process units and connecting streams. To draw more complicated circuits requires only a repetition of the principles outlined. To begin:

1. Install Excel and configure the toolbars for *Limn*. (See previous section.)
2. Start Excel.
3. Open a new worksheet.
4. Maximise the Excel window.
5. Click the "*Limn: Draw*" icon in the middle of the *Limn* toolbar.
6. The *Limn* drawing mode screen should now be present. (For a detailed description of the *Limn* menu and toolbar see the *Limn* on-line Help).

Adding a unit icon.

Move the mouse cursor to the centre of the screen.

Double click the mouse left button.



Select **Add a Unit Icon** from the displayed pop-up menu.

 **Tip: Alternative access to icon select dialog**

The **Add Icon** toolbar button  can be used to access the “Select a Unit Icon” dialog. Menu **Add**, option **Unit Icon**, or the short-cut key combination CTRL+U will also access this dialog.

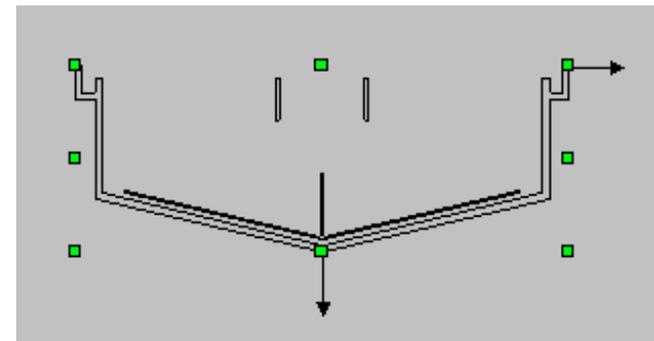
From the unit icon selection dialog which pops up, navigate to the \LIMN\ICONS directory containing the icon files installed during *Limn* setup.



Browse the list of icons to examine the available icon drawings.

Select a suitable thickener icon name. (double click left button on icon description line, select the OK button, or position cursor on description line and press ENTER)

A picture of a thickener and its product streams will appear on the screen.





Tip: Text Box Unit Icons

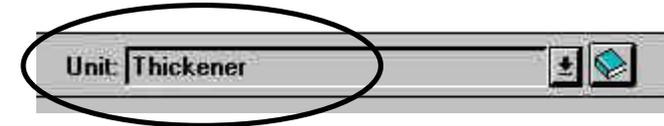
Limn treats Text Box Unit Icons in the same manner as Picture Unit Icons (i.e. they represent process units in the flowsheet sense). Text Box Unit Icons are thus useful for drawing block diagrams, or for representing an object for which no picture icon is available.



Tip: Sizing drawing objects

While dragging a corner “selection tag”, hold the Shift key down to ensure that the original aspect ratio of a drawing object is maintained during resizing.

Position the mouse cursor over the picture (note the cursor shape changes) and click the mouse left button. Note that “selection tags” appear about the unit icon picture, and that the name selection box on the right hand end of the toolbar is filled with a default name (Thickener).



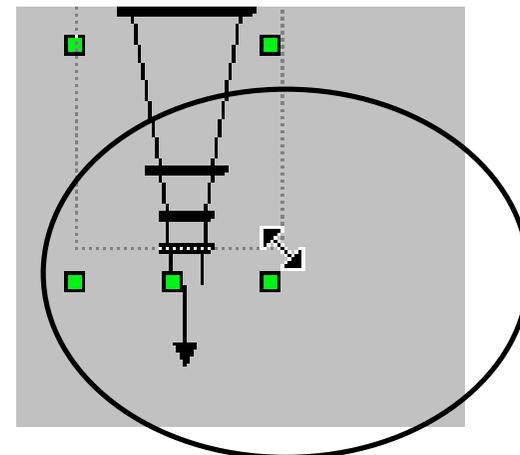
Naming Units

Move the mouse cursor to the name selection box and click the left button to select the entire temporary name. Type in a name for the icon (e.g. Water recovery thickener) and press ENTER.



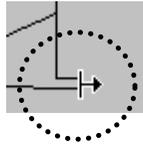
Adding a second Unit Icon

Repeat the process outlined above, selecting a hydrocyclone unit icon, and placing it to the left of the thickener. Resize the hydrocyclone icon to an appropriate scale by clicking on the unit “selection tags” and dragging the corner of the unit to the desired location.

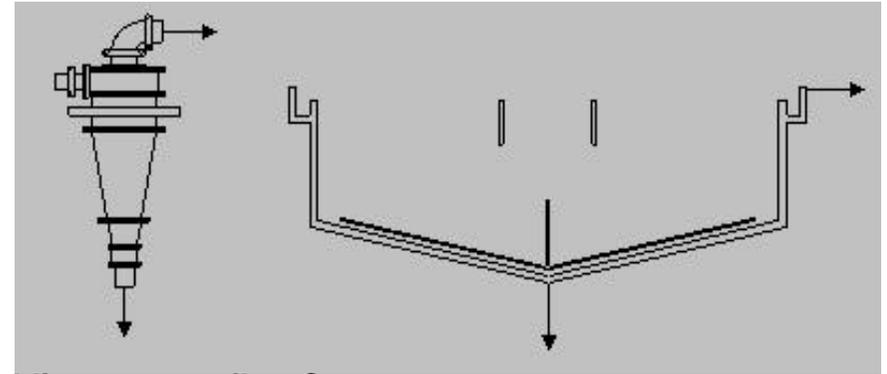


 **Tip: Drawing new streams**

If new streams are required (additional to those associated with units), they may be started by simply dragging with the mouse left button down, on any free area of the screen. New streams attached to a unit can be started by clicking the mouse left button when the cursor is over the unit boundary rectangle. i.e. when the cursor looks like:

 **Tip: Drawing object selection**

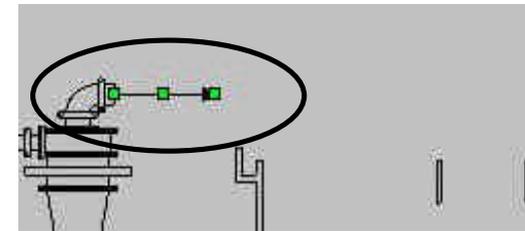
Drawing objects (unit icons, streams etc) may also be selected from a list accessible via the **Edit** menu, **Select Limn Objects** option, or by shortcut key combination CTRL+O.



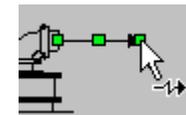
Adding process flow Streams

To draw a connecting stream or flow line from the hydrocyclone overflow into the thickener feed well:

1. Select the stream coming from the hydrocyclone overflow.



2. Position the mouse cursor over the “selection tag” on the end of the selected stream flow line.



Note the change to the cursor when it is positioned correctly.



Part 1 Getting Started

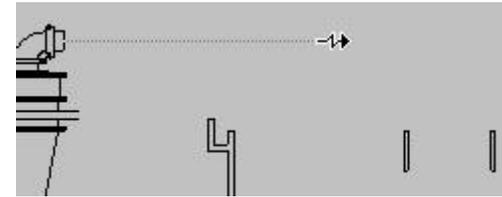


Tip: Stream naming

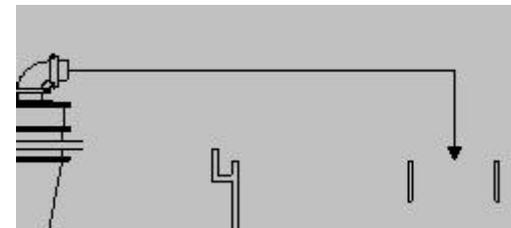
Stream flow lines may be given meaningful names in the same manner as unit icons.

(Using the toolbar name selection box)

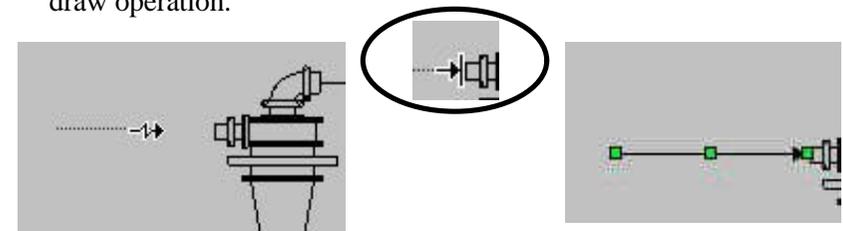
3. Drag the stream end to extend it toward the thickener.



4. When satisfied with the line segment length, click the mouse left button to allow a change of direction. Note that the line segment changes from dotted to solid. Drag the cursor down to extend the flow line downwards. Double click the left button to terminate the stream line.



5. The other stream flow lines may be extended in a similar manner.
6. To create a Feed Stream for the cyclone, click and drag the mouse left button in the required stream start location, and extend the stream toward the cyclone until it just touches. (Note the cursor appearance on contact.) Double click the mouse left button to complete the stream draw operation.



 **Tip: Cursor shapes**

The following special cursor shapes are used in the **Limn: Draw** module as the cursor moves about the screen:

	'Normal' (default) cursor.
	Click to move object under cursor.
	Click to start a stream in direction shown.
	Click to extend stream forward in direction shown.
	Click to extend stream backward in direction shown.
	Click to move a vertical stream segment.
	Click to move a horizontal stream segment.
	Click to size object diagonally in direction shown.
	Click to size object in direction shown.
	Left button to zoom in, right to zoom out. (CTRL+SHIFT to access)

 **Tip: Moving streams/stream segments**

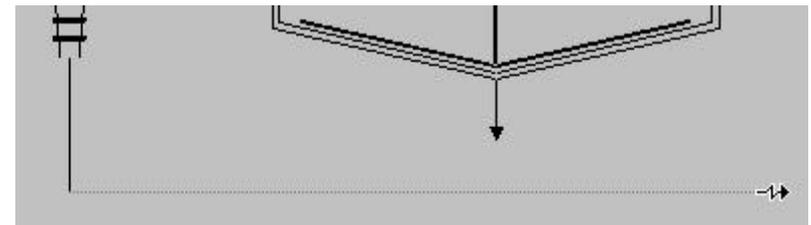
Segments of stream flow lines may be moved perpendicular to their drawn direction simply by dragging. To move the entire stream, hold the SHIFT key while dragging.

In this circuit, the high density streams from the thickener and hydrocyclone join to form a single high density product stream. To do this:

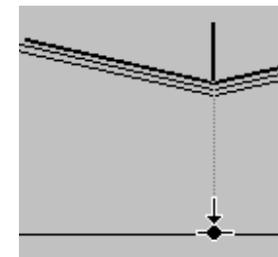
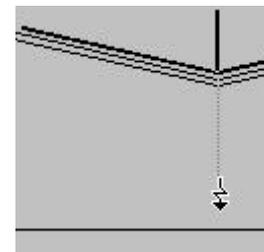
1. Select the stream flow line at the hydrocyclone underflow.



2. Extend the hydrocyclone underflow stream down and to the right, past the thickener.



3. Double click to complete this stream. Select the stream flow line at the thickener discharge. Extend it down toward the hydrocyclone underflow,



Note the change in cursor shape to indicate that a joining node will be inserted if the stream flow line is terminated at this point.

4.

 **Tip: More cursor shapes**

The following special cursor shapes are used in the **Limn** Draw module during move, draw or shape actions:



Move selected object in all directions.



Move a copy of the selected object.



Move a vertical stream segment.



Move a horizontal stream segment.



Extend stream forward in direction shown.



Extend stream backward in direction shown.



Size object diagonally in direction shown.



Size object in direction shown.



Size object while maintaining aspect ratio.

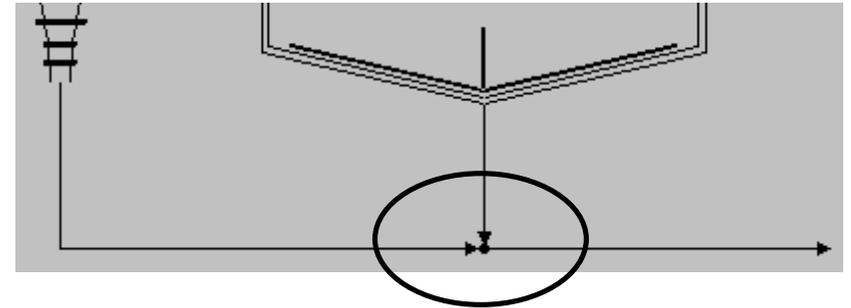


Terminate stream line at unit edge.



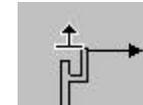
Terminate a stream line at a new joining node.

Double click to complete the stream. Note that *Limn* inserts a joining node and terminates the lines entering the node with arrowheads. The stream coming from the node is given a name based on the stream into which the node was inserted.

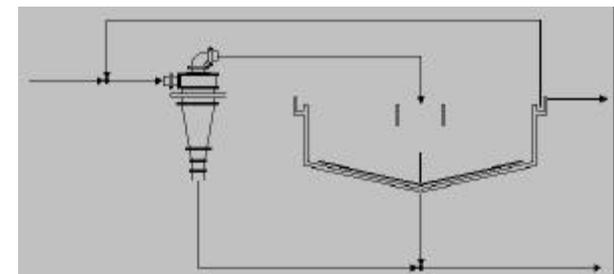


To illustrate the effect of recycle streams on the flowsheet solution, this example includes a recycle of water from the thickener overflow to the cyclone feed. This requires addition of a new product stream from the thickener. To achieve this:

1. Position the cursor on the boundary of the thickener near the overflow lip. When on the boundary, the cursor will change shape to:



2. Click the mouse left button to begin drawing the stream flow line. Extend the line up, to the left, and then down to join with the cyclone feed (at an automatically inserted joining node), clicking with the mouse left button whenever a change in line heading is needed.



**General drawing tips**

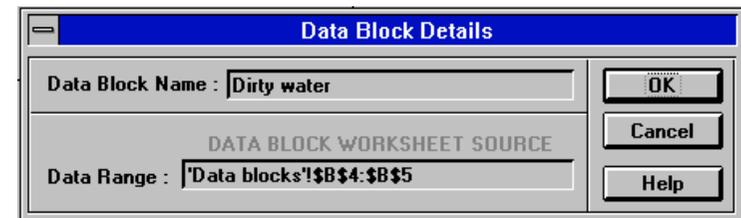
- To delete a unit icon, stream, text box icon, or data block, select it and press the Delete key.
- Once streams are connected (i.e. terminated near) to a unit, moving the unit will move the connected streams. Where streams join two units, the stream segments closest to the moved unit are adjusted to accommodate the movement. To move a unit without moving the connected streams, hold the Shift key down while performing the move.
- Stream joining nodes can be moved in the same manner as a unit, and will always remain connected to their streams.
- To copy a unit icon, text box icon, or data block, hold the Ctrl key down while moving the object. As the copy is given an incremental name (suffix _2 etc.), it can save time to name the first object with a generic name (e.g. Cleaner).
- To maintain the aspect ratio of a unit icon, text box icon, or data block during sizing, hold the Shift key down while dragging the object selection tag.

Adding a Data Block

Data Blocks are pictures of spreadsheet cells from elsewhere in the Workbook, which may be displayed on the process flowsheet.

To add a data block:

1. Move the mouse cursor to a clear area of the screen.
2. Double click the mouse left button. Select “Add a new Data Block” from the pop-up menu. A small blank rectangle will appear at the cursor position.
3. To associate the data block with spreadsheet cells, move the cursor over the block and double click with the mouse left button. This opens a data entry dialog which allows a spreadsheet range to be specified.



4. Move the cursor to the range entry field and select the current contents (if any). If a range is selected, the range location will be displayed. Select the sheet on which the data for display is present. Select the data range (click and drag) and click the OK button on the dialog.
5. The data block is resized and will display the contents of the selected cells.

A data block is a linked picture of the cells which are specified as its range. Any formatting applied to those cells (column width, row height, colour, font, borders etc.) will be displayed on the flowsheet. Best results will be gained if a special worksheet is set aside for summary data to be used in data blocks. Any required formatting and arrangement of data (e.g., 2x2, 2x1 cells etc.) can then be performed by copying data to this sheet, without affecting normal data relationships.

Since it is easier to set up data blocks once the stream data has been specified, we will return to them in chapter 4.



Tip: The *Limn* Flowsheet

Limn can handle multiple flowsheets within the one Excel Workbook, but will only solve one sheet at a time. The sheet solved is either the currently selected sheet (if it is a flowsheet), or the first flowsheet encountered in the sheet tab order.

A *Limn* workbook

Introduction

There are many different ways to set up worksheets for problem solving using Excel. The addition of *Limn* to Excel does not change this. As you gain experience in using *Limn*, you will undoubtedly find your own preferred methods of arranging the problem to suit your requirements.

The following is a suggested arrangement of worksheets within a workbook which can be used as the basis of your own workbooks, and which should make the data requirements for flowsheet solution with *Limn* a little clearer.

The Flowsheet

As a minimum, *Limn* requires one worksheet on which to draw and display the process flowsheet. Because of the way in which *Limn* adjusts column heights and row widths during drawing, you will generally not find it convenient to use this sheet to perform calculations, or to store data.

The default worksheet on which we have drawn our example process flowsheet during Chapter 3 is the *Limn* flowsheet. It would be useful to rename the sheet at this point to represent that fact. (In Excel, click the mouse right button over the sheet tab, or double click the mouse left button over the sheet tab to access the Excel rename sheet dialog.)

Process Stream data

It is convenient to store all process stream data on a single worksheet. To set up the process stream data sheet required to continue the exercise begun in Chapter 3:

1. Select (or add) an unused sheet.
2. Rename the sheet to “Stream Data”.

The form consists of a grid of horizontal lines for data entry. A vertical line is positioned on the left side, creating a narrow column. The top and bottom edges of the grid feature small tick marks, suggesting a scale or measurement system. The grid is currently empty, providing space for recording data.

- 3 Across the top of the sheet, in individual columns starting from the second column, enter the names of the streams on the flowsheet.
- 4 In the first column, enter the names of the components of each stream. For this exercise, they are tonnes per hour of solids and tonnes per hour of water.
- 5 Enter the formulae for any derived values (for example %solids and volume flowrate) and set up any other fields that are required for calculation of derived values (such as solids specific gravity)
- 6 Highlighting the cells which require user input is a useful practice. For *Limn* flowsheet solution, only the feed stream data are required to be entered.
- 7 Another useful addition to the stream data is a column which sums the circuit products, allowing checks against the summed circuit feed data. The use of such self auditing data summaries can reduce flowsheet solution debugging time considerably.

The completed sheet could look something like:

	A	B	C	D	E	F	G	H	I	J	K
1											
2		Fresh	Cyclone	Cyclone	Cyclone	Slimes	Slimes	Thickener	Solids		All
3		feed	feed	overflow	underflow	recycle	product	solids	product		products
4	Solids t/h	100.00									0.00
5	Water t/h	500.00									0.00
6	%Solids	16.67	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!		#DIV/0!
7	Volume m3/h	531.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00
8											
9											
10	Solids SG	3.20									
11											

Estimates of the mass flowrates for each of the various streams could be entered at this stage if desired. These will be the start points for the flowsheet solution.

Note that the data from which *Limn* calculates the flowsheet solution must be in terms of mass, volume, or equivalent properties. i.e. the components must be able to simply be summed at any joining node (or when being combined for presentation to the process unit model).

Assays and other fractional or percentage measures are not suitable, and must first be converted to a property which can simply be summed (for instance, mass). This is usually done by setting up an extra worksheet to hold the data in 'user space' (e.g. assays) and entering Excel formulae to convert to and from mass related data. (For an example and more details, see Part 2 of this manual.)

Model data

It is advisable (at least when starting out) to set up each individual unit process model on a separate worksheet. Each model should be a stand-alone block of calculations with no links to any other worksheet. (The ***Limn:Solve*** engine uses the process connection matrix derived from the flowsheet to link these stand-alone models together.)

In the example under consideration, there are two spreadsheet models required, one for the hydrocyclone, and one for the water recovery thickener. These will require two sheets. Set them up in the following manner:

1. Select (or add) two unused worksheets, and name them "Cyclone Model" and "Thickener Model" respectively.
2. Access the "Cyclone Model" sheet. We will use an elementary mass split model to describe the hydrocyclone, simply specifying the fraction of each feed component which reports to the cyclone underflow.
3. Set up columns for the feed to the unit model, for the split fraction to be used, and for the two products from the model.
4. Set up rows for each component in the feed (in this case, just Solids and Water).
5. Highlight the data to be modified by the user (the split factors).



Tip: Example Workbooks

Completed Excel Workbooks for the examples used in this manual may be found in the "EXAMPLES" subdirectory created when **Limn** is loaded. The workbook for this "Getting Started" section is named "DEWATER.XLS".

6. Specify the formulae which results in the feed data being allocated to each of the products according to the split factor. For the underflow, this is the feed component mass times the split fraction for that component. For the overflow, this is the feed component mass minus the mass reporting to the underflow.
7. Test the unit model by entering a set of dummy feed data and ensuring that the correct data appears in the product data fields. The result should appear something like:

	A	B	C	D	E	F
1						
2		Cyclone Model				
3			t/h	Split to	t/h	t/h
4			Feed	Underflow	Underflow	Overflow
5		Solids	100.00	0.70	70.00	30.00
6		Water	100.00	0.05	5.00	95.00
7						
8						

8. Access the “Thickener Model” sheet. The model used for the thickener will be similar in concept to that for the hydrocyclone, but complicated slightly by the need to consider the amount of slimes product recycled to the circuit feed.
9. Repeat the procedure already performed for the hydrocyclone, setting up rows for each component (Solids and Water) and columns for each stream. For clarity, we will consider the thickener and subsequent slimes recycle split as two processes, and set them up on the sheet accordingly. Thus, the first stage of the model is identical to that used for the hydrocyclone, with the user specifying the fraction of each feed component reporting to the thickener solids product. The second stage of the model is set up to allow a further split factor to be specified; the total split between the recycle slimes stream and the circuit product slimes stream.

10. On completion of formulae specification, test this new model with a set of dummy feed data.

11. Results should be something like:

	A	B	C	D	E	F
1						
2				Thickener Model		
3						
4				Solids/Slimes split		
5			t/h	Split to	t/h	t/h
6			Feed	Underflow	Underflow	Overflow
7		Solids	100.00	0.95	95.00	5.00
8		Water	100.00	0.01	1.00	99.00
9						
10				Recycle Slimes split		
11			Slimes split to recycle	0.20		
12			t/h	Split to	t/h	t/h
13			Slimes	Recycle	Slimes recycle	Slimes product
14		Solids	5.00	0.20	1.00	4.00
15		Water	99.00	0.20	19.80	79.20
16						
17						



It is important to note that, so far, all setting up for the unit models has been done solely in Excel, and that the models are **NOT** linked back to the stream data sheet. They are truly stand-alone models.

Data Block links to spreadsheet data

Data blocks are used as a way of displaying data from Excel fields at any location on the **Limn** flowsheet. The Excel 'Picture' object, which forms the basis of the **Limn** data block, provides a 'picture' of a cell range which can then be treated as any other drawing object. Because the data block is a 'picture' of a group of cells, the formatting of the 'pictured' cells is present in the picture object.

Thus, to obtain appropriately formatted data blocks, the cells on which they are based must be formatted in the required manner. There is often a conflict between the formatting requirements of stream data and that required for data blocks. For this reason, a separate sheet is used to hold a copy of the stream data in a form suitable for display as data blocks on the flowsheet.

To set this sheet up:

1. Rename an unused (or new) worksheet as “Datablock Data”.
2. Access the sheet and set up Excel formula to copy data from the “Stream Data” sheet to this new sheet. We will use a 2x2 data block for the example, with Solids tph and Water tph on the top line, and % Solids and volume flowrate on the bottom line.
3. A useful way to set up the data blocks is to copy the stream names (appropriately separated) into column 1, and then to set up the copied data next to them, with appropriate formatting applied. A section of the new sheet might look like:

	A	B	C	D	E	F
1						
2		Legend			Solids t/h	Water t/h
3					%Solids	Volume m3/h
4						
5		Fresh	100.0	500.0		
6		feed	16.7	531.3		
7						
8		Cyclone	0.0	0.0		
9		feed	#DIV/0!	0.0		
10						
11		Cyclone	0.0	0.0		
12		overflow	#DIV/0!	0.0		
13						

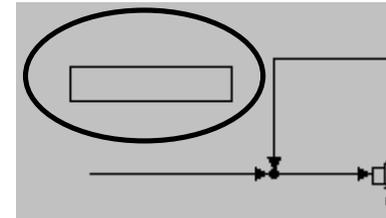
= 'Stream Data'!\$C\$4	= 'Stream Data'!\$C\$5
= 'Stream Data'!\$C\$6	= 'Stream Data'!\$C\$7

The image shows a blank grid for a flowsheet processor. It features a vertical margin line on the left side, creating a narrow column. The rest of the page is filled with horizontal lines, providing a structured area for writing or drawing. The grid is bounded by a dotted line at the top and a solid line at the bottom.

Creating the Data Blocks

In Chapter 3, we deferred creating the data blocks until we had set up a sheet to contain the source data. Having completed that step, we can now create the data blocks and place them on the *Limn* flowsheet. To achieve this:

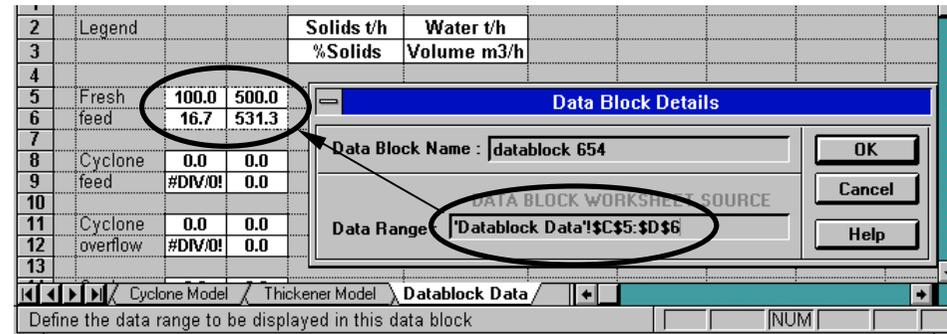
1. Switch to the Excel worksheet labelled “Flowsheet”.
2. Enter the *Limn*.Draw mode (click on the *Limn*.Draw button on the *Limn* toolbar).
3. Move to an area adjacent to the circuit feed stream flow line.
4. Double Click the mouse left button and select option **Add a new Data Block**.
5. An outline box will appear at the cursor position:



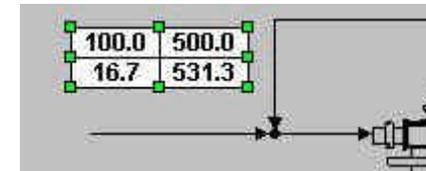
6. Position the mouse cursor over the box and double click the left button.
7. A dialog box will appear, allowing a change to the data block name, and entry of a group of cells associated with the block.

Data Block Details	
Data Block Name : <input type="text" value="Feed data"/>	<input type="button" value="OK"/>
DATA BLOCK WORKSHEET SOURCE	
Data Range : <input type="text"/>	<input type="button" value="Cancel"/>
	<input type="button" value="Help"/>

8. Change the name to something more meaningful, and click with mouse left button in the Data Range field.
9. The sheet tabs will appear at the bottom of the worksheet, allowing access to the “Datablock Data” sheet. Select that sheet, and select the range appropriate to the data block (in this case, the “Fresh Feed” data, or cell range ‘Datablock Data’!\$C\$5:\$D\$6).



10. Click the OK button (or press ENTER). The outline box will be replaced with a picture of the cells from the “Datablock Data” worksheet:

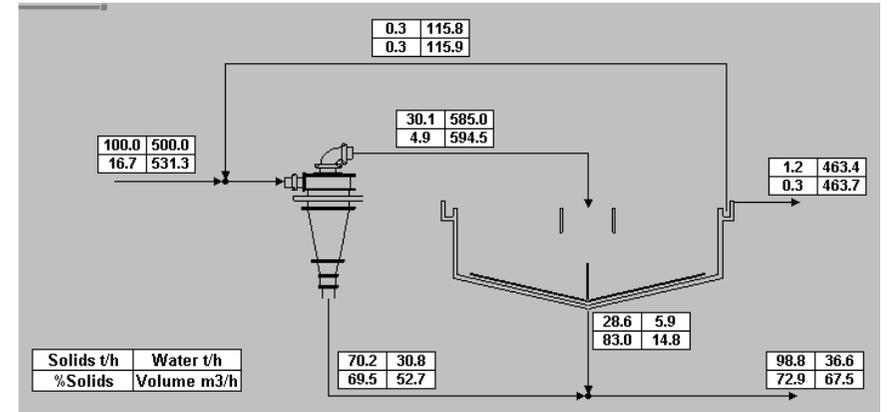


11. The data block may be moved about the screen in the normal manner to position it in the desired location.

12. Although the data block has ‘selection tags’ which function as sizing handles in the same manner as those on unit icons and text box unit icons, it is recommended that a data block is not resized on the flowsheet unless absolutely necessary. Resizing should be done by changing the cell sizing of the linked cells of which the data block is a ‘picture’. It is a feature of Excel that Picture objects, once sized manually, will not resize automatically.



13. Other data blocks may be added to the flowsheet as required to show the flowrate and other information for each stream. A data block legend may be included if desired. The completed flowsheet looks like:



The form consists of a grid of horizontal lines for data entry. A vertical line is positioned on the left side, creating a narrow column. The top and bottom edges of the grid feature small tick marks, suggesting a scale or measurement system. The grid is currently empty.

Solving the flowsheet

Linking Worksheet Data to Flowsheet Objects

Data block links to spreadsheet data

We saw in the previous chapter how the detail data entry dialogs for data blocks are used to link the formatted cells on the worksheet to the picture elements displayed on the flowsheet.

Stream flow lines and process unit icons are linked to their underlying data in a similar manner.

Stream data

For flowsheet solution, a stream is represented as a vector or matrix of data consisting of the masses (or mass related properties) of each component of the stream which is of interest.

In the simple dewatering circuit example, the stream data is the tonnes per hour of solids, and the tonnes per hour of water. We have already specified fields for these data on the “Stream Data” worksheet.

For the solution engine, only those streams which input fresh material to the circuit are absolutely required. In most cases however, the purpose of performing the flowsheet solution is to examine the estimated data in other streams in the flowsheet. **Limn** maintains internal tables of stream data during the solution; we now need to link those tables with the data fields we have set up on the “Stream Data” worksheet. To do this:

1. Select the “Flowsheet” worksheet and enter **Limn:Draw** mode.
2. Position the mouse cursor over the required stream and select it by double clicking the left button.
3. A stream detail data entry dialog is displayed (similar to that used for data block data entry), in which can be entered the spreadsheet cell range corresponding to the data for this stream.

The form consists of a grid of horizontal lines for data entry. A vertical line is positioned on the left side, creating a narrow column. The top and bottom edges of the grid feature small tick marks, suggesting a scale or measurement system. The grid is currently empty.

4. Use the sheet tabs to move to the “Stream Data” worksheet and select the cell range for the stream.

	A	B	C	D	E	F	G	H	
1									
2		Fresh	Cyclone	Cyclone	Cyclone	Slimes	Slimes	Thickener	Sc
3		feed	feed	overflow	underflow	recycle	product	solids	pro
4	Solids t/h	100.00	100.30	30.09	70.21	0.30	1.20	28.59	96
5	Water t/h	500.00	675.84	585.05	30.79	115.84	463.36	5.85	36
6	%Solids	16.67	14.01	4.89	69.51	0.26	0.26	63.01	72
7	Volume m3/h	531.25	647.18	594.45	52.73	115.93	463.73	14.78	67
8									
9									
10	Solids SG								
11									
12									
13									
14									
15									
16									
17									
18									

Stream Details	
Stream Name :	Fresh Feed
STREAM DATA WORKSHEET REFERENCE	
Data Range :	[DEWATER.XLS]Stream Data'!\$B\$4:\$B\$5
From:	nil
To:	Cyclone feed joining node
	OK
	Cancel
	Help

If the stream range represents an input stream, the data in this range will be used by the **Limn:Solve** engine as inputs to the solution.

If the stream range represents an output or intermediate stream, the **Limn:Solve** engine will overwrite the data in the selected range with the results of the flowsheet solution.

Unit Data

The purpose of the unit detail data is to tell the **Limn:Solve** engine where on the worksheet to place the combined feed data for the unit, and where to find the unit products once the worksheet has been calculated.

The fields with this data have already been specified on the individual unit model worksheets.

To specify the unit detail data:

1. Select the “Flowsheet” and enter **Limn:Draw**.
2. Select the first unit (e.g. the hydrocyclone) and double click the left mouse button to open the dialog.

The form consists of a grid of horizontal lines for data entry. A vertical line is positioned on the left side, creating a narrow column. The top and bottom edges of the grid have small tick marks. The grid is currently empty.

3. Select the Feed Range field, use the sheet tabs to access the “Cyclone Model” sheet and select the two cells which represent feed to the model.
4. Select the field for Product 1 Range. Note that the right hand block of the dialog has a list of the unit product streams in order as drawn on the flowsheet. Thus, Product 1 is Cyclone Overflow.
5. Tab to the “Cyclone Model” sheet and select the cells which represent the Cyclone Overflow stream.
6. Repeat for the second product (Cyclone Underflow)

 **Tip: Product Stream Order**
 Unit product streams are ranked in the order in which they would be encountered in moving clockwise about the perimeter of the (unrotated) unit icon, starting at the top left corner.

	A	B	C	D	E	F	G	H	I
1									
2			Cyclone Model						
3			t/h	Split to	t/h	t/h			
4			Feed	Underflow	Underflow	Overflow			
5		Solids	100.30	0.70	70.21	30.09			
6		Water	615.84	0.05	30.79	585.05			
7									

Unit Details

Unit Name : <input style="width: 90%;" type="text" value="Cyclone"/>	MODEL TYPE <input checked="" type="radio"/> Spreadsheet <input type="radio"/> User (DLL)
SPREADSHEET MODEL - DATA REFERENCES	
Feed : <input style="width: 95%;" type="text" value="'[DEWATER.XLS]Cyclone Model!\$C\$5:\$C\$6"/>	Unit Products 1 Cyclone Overflow 2 Cyclone Underflow 3 nil 4 nil 5 nil 6 nil
Product 1: <input style="width: 95%;" type="text" value="'[DEWATER.XLS]Cyclone Model!\$F\$5:\$F\$6"/>	<input type="button" value="OK"/>
Product 2: <input style="width: 95%;" type="text" value="'[DEWATER.XLS]Cyclone Model!\$E\$5:\$E\$6"/>	<input type="button" value="Cancel"/>
Product 3: <input style="width: 95%;" type="text"/>	<input type="button" value="Help"/>
Product 4: <input style="width: 95%;" type="text"/>	
Product 5: <input style="width: 95%;" type="text"/>	
Product 6: <input style="width: 95%;" type="text"/>	

The data for the “Thickener Model” should be linked to the flowsheet in the same manner. Note that there are three product streams to specify.

The form consists of a grid of horizontal lines for data entry. A vertical line is positioned on the left side, creating a narrow column. The top and bottom edges of the grid have small tick marks. The grid is currently empty.

Solution Options

The ***Limn:Draw*** menu has a **Solve** sub-menu, one item of which allows setting of solution options. Using this dialog allows the user to obtain diagnostic flowsheet information, to control the display of error information, and to control screen updating.

Solving the Flowsheet

Flowsheet solution may be started by accessing the **Solve** option on the ***Limn:Draw Solve*** menu, or by clicking on the ***Limn:Solve*** button on the ***Limn*** toolbar.

If the ***Limn:Solve*** engine detects problems with the entered data, or the flowsheet network, a list of errors is displayed (this display can be controlled using the Solution Options dialog mentioned above).

If no errors are detected, the engine will attempt to solve the flowsheet using the logic detailed below.

On completion, data blocks are updated to reflect the balanced data now resident in the worksheet data fields.

The Solution logic

A spreadsheet model consists of a spreadsheet range representing the feed components to a process unit, and up to six spreadsheet ranges representing the products from that unit. Components are user defined, with the restriction that they must be additive, i.e. where streams join, a simple sum of the component value will describe the value of the component in the combined stream. Thus, flowrates expressed in mass or volume are suitable, percentage assays are not. (These latter can be handled by converting to mass or similar in the spreadsheet.) The calculations which connect the feed and the products must be written in the spreadsheet using normal Excel functions.

The model feed and product ranges are entered into the respective fields on the unit icon detailed information dialog.

The flowsheet solution engine for spreadsheet models works in the following manner:

- For each unit in turn:
 - From the flowsheet drawing get a list of all streams feeding the unit.
 - For each feed stream in turn:
 - Get the feed stream spreadsheet data range. This range is user specified in the stream details dialog.
 - Get the component data from this range.
 - Add the component data to the combined feed data.
 - Store the summed feed components in the "Model Feed Range" spreadsheet range. This range is user specified on the unit details dialog.
 - Do the next feed stream.
 - From the flowsheet drawing get a list of all streams exiting the unit (product streams).
 - For each product stream in turn:
 - Access the data in the "Model Product Range" spreadsheet range. This range is user specified on the unit details dialog, and will contain the results of the spreadsheet calculations.
 - Get the product stream spreadsheet data range. This range is user specified in the stream details dialog.

- Copy the model product data to the stream data range
- Do the next product stream.
- Do the next unit.
- When all units have been done, recalculate the spreadsheet. (This allows the user written spreadsheet models to operate on the newly summed "Model Feed" data, to produce the "Model Product" data for use in the next iteration)
- Solution is complete if there has been no significant change in the sum of all stream components since last cycle. Otherwise, repeat the cycle for all units.

Examples provided

Several examples have been provided to highlight various aspects of the **Limn** package. While Excel workbook layout for data reporting and flowsheet solution is a matter of personal preference, these examples offer suggestions which can form the basis of a personal layout. As experience is gained with **Limn** and Excel, re-examination of the examples could prove useful as a source of ideas for data handling and manipulation.

DEWATER.XLS

This is the example used in the tutorial. (Part 1 of this manual). It demonstrates how to set up a simple flowsheet, flowsheet solution models and data display.

2CYCS.XLS

Although this example uses data derived from a simple solids and water balance, it demonstrates a method of data block data layout and formatting which would be useful for reporting plant survey work.

CRUSHING.XLS

The **Limn** engine is designed to handle properties which are additive at a joining node, such as mass, or volume. To enable properties such as a size distribution to be handled, the size fraction data must be converted to mass. This workbook demonstrates how that may be done. It also introduces two more simple flowsheet balance models, and shows how to use VBA code to calculate a %passing size.

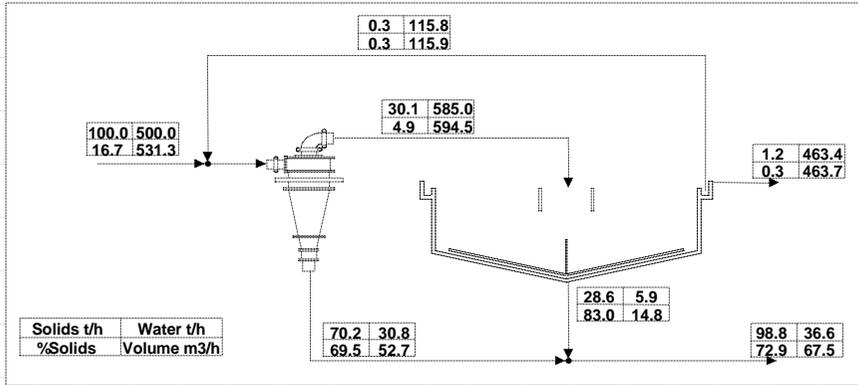
FLOTATN.XLS

For the same reason that size data must be converted to mass, assay data must also be converted. This workbook shows a method of handling assay data for flowsheet solution using **Limn**, and demonstrates how to implement flowsheet solving models based on the two-product formula.

EFFCURVE.XLS

The simplest method of flowsheet solution modelling using ***Limn*** is to write the models using normal spreadsheet functions. While this can be a surprisingly powerful technique, it has some limitations, for example, where internal iteration is required in a model. If the cell formula approach should prove inadequate, there are several features of Excel/Windows™ which allow complex process models to be included in ***Limn*** solutions. These alternatives are demonstrated in this example.

This example also demonstrates how ‘heritage code’ written in a (possibly outdated) high level language, can be incorporated into the ***Limn*** structure.



DEWATER.XLS

This example is a completed workbook based on the tutorial described in Part 1 of this manual.

Although a thickener would usually only have two product streams, it has been implemented in this example with three products (the third being a recycle to the head of the circuit) to demonstrate:

- that recycle streams can easily be handled.
- that extra products may be added to any unit - the user is in control, since they are writing the flowsheet solution model.

Workbook layout

The example illustrates one possible layout of a workbook for flowsheet solution work using *Limn*. This layout uses separate worksheets for:

- the flowsheet
- stream data
- each process unit model
- data block formatting.

Separating the components of the flowsheet solution in this manner makes it easier to see that the process unit models are individual ‘sub-routines’ of the overall solution.

Note that the process unit model sheets have no spreadsheet cell function links with any other worksheet in the workbook. They are truly ‘stand-alone’ models. Likewise, the stream data summary sheet has no spreadsheet cell function links to the models. (The *Limn* engine handles the process of connecting the units and streams, according to the connection logic defined by the user drawn flowsheet.)

Keeping the data block formatting separate from the stream data allows the stream data to be arranged in a manner suitable for data input, while the data formatting arranges it in a manner suited to display on the flowsheet.



Tip: Stand-alone models

To check that the models are indeed completely 'stand alone', use the menu **Tools, Auditing, Trace Precedents** while selecting any of the feed components. There should be no connections.

Manual entry of data in the feed component cells should result in the model calculating new products for that set of feed conditions. Individual models should be debugged in this way before the **Limn:Solve** engine is run.

The models

There are only two models used in this flowsheet solution, and while they are trivial, they are typical of a class of models which may be useful for simple solids and water balances. Both models are based on the concept of a split factor, which determines the fraction of a particular component (in this case, solids, or water) reporting to a particular product stream.

Cyclone model

This model simply multiplies the split factor by the feed for each component (solids and water) to define the flowrate of product to the underflow stream.

The overflow stream is calculated by difference.

Thickener model

Although similar to the cyclone model, the thickener model has the added complication of an extra stream (the recycle stream). To facilitate this, the split is handled in two stages.

Stage 1 is exactly analogous to the cyclone model, with the split factor for each component (solids and water) multiplied by the feed flowrate to define the flowrate of product to the underflow stream. An intermediate stream to represent the total thickener overflow product is calculated by difference.

This stream is then split using the same split factor for both solids and water (as if the split were being done via a diverter valve), to produce the recycle stream, and the final thickener low density product stream.



Tip: *AUDIT CHECKS*

Note the column of data at '[DEWATER.XLS]Stream Data!'K2:K7 which sums all products resulting from the flowsheet solution. If the solution is correct, this should equal the sum of all feed (input) streams. Such 'audit checks' are a very useful tool for determining if the solution is functioning correctly.



Tip: *Datablock sizing*

Do not adjust the size of data blocks by using the sizing handles in ***Limn:Draw***. Rather, change the column width and row heights on the sheet to which the data blocks are linked. Since the data blocks are an Excel 'linked picture' of the cells, the size of the data block automatically changes to reflect the cell height and width.

Stream Data

The 'Stream Data' worksheet is formatted for ease of data entry. For this simple flowsheet solution, the only data required from the user are the flowrates of water and solids in the feed to the circuit. Derived values are calculated from the mass flow data (e.g. %solids from tph Solids and tph Water - Worksheet '[DEWATER.XLS]Stream Data!'B6:I6). A user input solids SG is required to allow these calculations.

All of the data input fields are highlighted to prompt the user (white text on blue background).

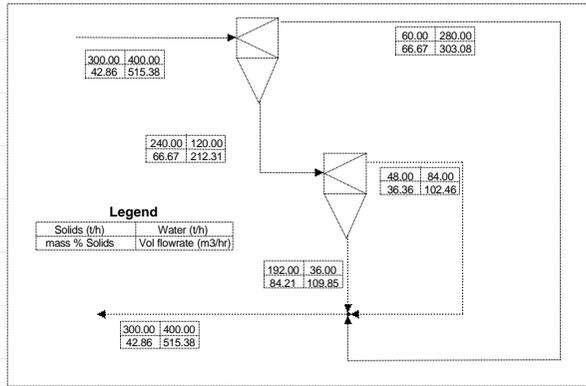
The mass flow data for solids and water for all streams other than process input streams are calculated and placed in the cell locations associated with each stream's mass flow data by the ***Limn:Solve*** engine. Note that there are no cell equations in these cells (Worksheet '[DEWATER.XLS]Stream Data!'C4:I5)

Datablock Data

This sheet shows one method of accessing the stream data and formatting it suitably for display on the flowsheet. See Example 2 (following) for a more flexible method.

Each cell in the required 2x2 block is linked with a cell formula to the corresponding data on the 'Stream Data' sheet. Suitable formatting (borders, cell colours, alignment, number of decimal places etc.) is then applied.

The data blocks on the flowsheet are linked (Excel 'linked picture' objects) to the 2x2 cell blocks on the 'Datablock Data' sheet and any change in data or formatting on this sheet is immediately echoed on the flowsheet.



2CYCS.XLS

This example is included to show a more flexible method of associating formatted data blocks with the stream data (which is usually formatted for ease of data entry or for examination in tabular form).

Workbook layout

The workbook is arranged in much the same manner as the previous example - individual sheets for flowsheet, stream data and data block formatting. In this example, however, both cyclone models are present on one 'Cyclone Models' unit data sheet. Note, however, that they are both still stand-alone models, with no cell formula relationships between them.

The flowsheet is simply two cyclones in series, with all product streams re-combined; hardly a likely real-life situation. On the 'Stream Data' sheet, an audit check is done by subtracting the combined product stream from the feed stream. The result should be zero for each component.

Data block formatting

Sheet 'Datablock Data' demonstrates the improved links between the stream data in its native form, and in its formatted form.

Rather than simply copying the stream data to the required formatting cell by a direct link (cell formula "='Stream Data!'G7" or similar), the OFFSET formula is used allow the link to each formatting cell to be indexed by the value in another cell.

The OFFSET formula row index is set to reference a cell which contains the component number required in that data block location. Similarly, the OFFSET formula column index is set to reference a cell which contains the stream number on the 'Stream Data' sheet.

Once the formula is set up and working in one group of 2 x 2 cells in the data block, it can be copied into each of the other data block locations. (Care will be required with absolute and relative cell addressing. See location 'DataBlock Data'!B8 for an example)

 **Tip: 'Forms' elements on worksheets.**
Excel allows many of the elements from the **Forms** toolbar (such as drop down lists) to be used on the worksheet. Several of these are useful for enhancing the user interface on worksheets which will be used by others. To find out more about the **Forms** toolbar, search the Excel on-line help with key word *FORMS*.

	H	I	J	K
3	Define data for data blocks			
4	Solids (t/h) ↓		Water (t/h) ↓	
5	mass % Solids ↓		Vol flowrate (m3/hr) ↓	
6				

By using this method, simply changing the component index number (e.g. any cell in the range 'DataBlock Data'!B2:C3), will result in the component displayed at that location in the 2 x 2 data block grid also being changed for all stream data blocks.

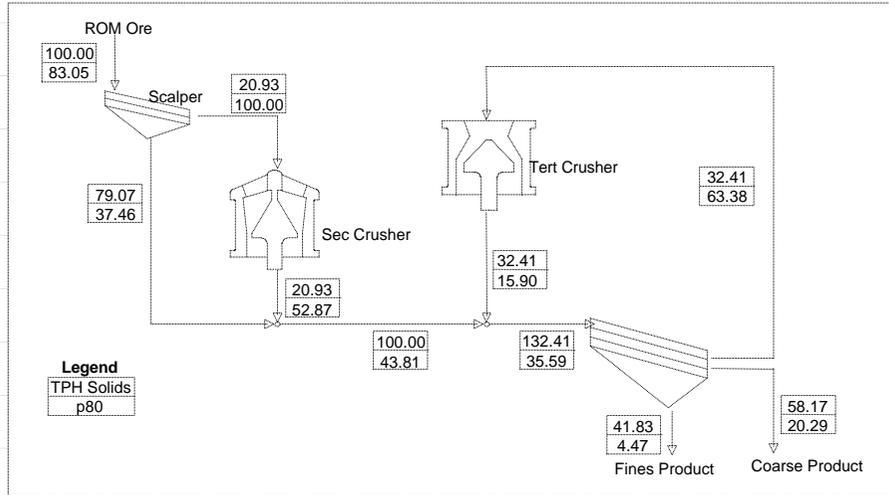
2CYCS_2.XLS

Example 2CYCS_2.XLS demonstrates a further enhancement of this approach.

At location 'Flow sheet'!G3:K6 in workbook 2CYCS_2.XLS, 4 drop down lists have been placed on the flowsheet. Each drop down list has the list of components ('Stream Data'!\$B\$5:\$B\$10) as its source 'Input range'. The 'Cell link' for each drop down is the index location on the data block formatting sheet (e.g. 'DataBlock Data'!\$B\$2).

By clicking the mouse on any of these drop down lists, a selection may be made from the components. The drop down list automatically places the index of the chosen component in the correct location on the data block formatting sheet. This causes a change in the OFFSET reference in each of the 2 x 2 data block grid of cells, which displays the new component.

Since the data blocks on the flowsheet are linked pictures of the 2 x 2 grid of cells on the data block formatting sheet, these data blocks are updated with the correct stream data also.



Tip: Flowsheet - Data connections

To find out which worksheet data cells are connected to a particular flowsheet stream or unit, click on the stream or unit when viewing the flowsheet in Excel. This activates a macro which will cause Excel to jump to and select the data associated with each stream or unit. (For a unit, the selected block is the combined unit feed data).

CRUSHING.XLS

This example demonstrates the following:

- conversion of size distribution data to mass flow data to allow solution using the **Limn:Solve** engine.
- crusher modelling using an approach similar to the manufacturer's 'grading curve' approach
- screen modelling using the split factor approach
- the use of a local function macro in VBA code to calculate a % passing size in each stream.

Worksheet 'StreamDataSheet'

There are two blocks of data on this worksheet. The upper block ('StreamDataSheet'!\$A\$2:\$L\$20) is the stream data in a format suitable for data entry and analysis (in this case, cumulative % passing). The lower block ('StreamDataSheet'!\$A\$22:\$L\$39) is the same data, in mass flowrate form suitable for flowsheet solution using **Limn:Solve**. It is this lower block which is connected to each stream on the flowsheet using the 'Stream Detail Data' entry dialog in **Limn:Draw**.

Note that the data entry area is highlighted (white on blue).

Conversion of size distribution data to mass flowrate

The raw data for the feed, which is entered by the user, must be converted to mass flowrate data. This is achieved in cells 'StreamDataSheet'!\$C\$26:\$C\$38, where cell formula convert the percentage of the feed in each size fraction to a mass, using the cumulative amounts either side of the size fraction, and the total flowrate.

Mass flowrate data for each size fraction in the feed is all the stream data that **Limn:Solve** requires. During flowsheet solution, **Limn** writes the calculated mass data into the cells 'StreamDataSheet'!\$D\$26:\$L\$38. Upon completion, this should represent a set of balanced flowsheet mass flowrate data.

Conversion of mass flowrate data back to a size distribution

To convert the mass flowrate data in range 'StreamDataSheet'!\$D\$26:\$L\$38 to cumulative % passing data for the user, more cell calculations must be carried out. Thus, cells 'StreamDataSheet'!\$D\$19:\$L\$19 sum the fractional mass flowrates to calculate the total flowrate for the stream, an cells 'StreamDataSheet'!\$D\$6:\$L\$18 contain formulae to take the fractional mass flows, and convert them to cumulative % passing.

Use of local function Macro

Cells 'StreamDataSheet'!\$C\$20:\$L\$20 contain formulae of the form:

=PassingSize(80,Sizes,C6:C17)

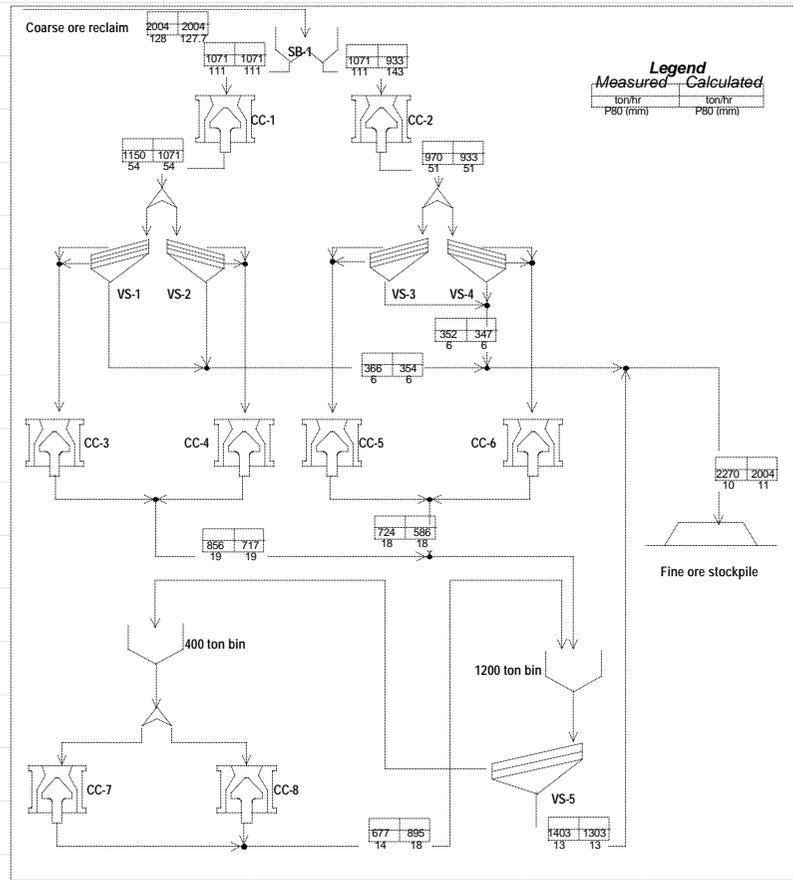
This formula is a local function, written in Visual Basic for Applications (VBA), the macro language of Excel 5. It calculates the passing size using linear interpolation in Rosin-Rammler space, given a particular percent value, the range of particle size data, and the range of cumulative size distribution data. It is being used in this instance to calculate the 80% passing size for each stream size distribution.

The VBA code for this function is present in sheet 'VBA_Calculations'.

Unit Models

Crushers

The simple models of crushers in worksheets 'Unit_SecCrusher' and 'Unit_TertCrusher' use a calculation which give results similar to that obtained from using the manufacturer's 'grading curve' approach. The required product size distribution is entered (cells \$D\$6:\$D\$18 on both worksheets). These data could be taken direct from 'grading curves'. The model then calculates the product mass flowrate in each size fraction by applying the size distribution percentage to the total feed tonnes.



Screens

The screen models used in this workbook are extensions of the simple split fraction models used in the previous examples. A split fraction to coarse product is entered for each size, the feed mass flow rate is allocated to the coarse stream according to this split, while the screen undersize is calculated by difference.

For the double deck screen, a second set of split fractions is provided to specify the split of the top deck fines product as it passes over the lower deck. Note that audit sums of the total flowrate of each deck product are calculated as a check.

Module sheet 'VBA_Calculations'

This module contains the code for the local function 'PassingSize', used to calculate the 80% passing size on the Stream Data sheet. It performs a linear interpolation in Rosin-Rammler space:

$$(\log_{10}(-\log_{10}(\text{cumulative \% passing})) \text{ versus } \log(\text{size})),$$

between the two bounding size fractions, to calculate a size value. Interpolation beyond the upper and lower size measurements is not performed.

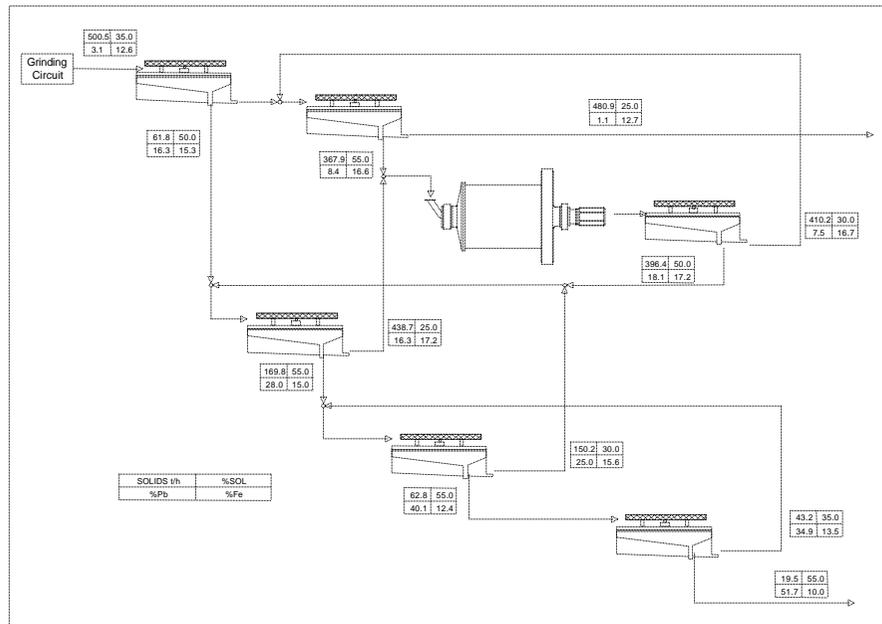
CRUSH_2.XLS

This workbook is an example of a more complicated crushing circuit. It represents three stages of crushing, with multiple recycle.

The workbook layout is much the same as for CRUSHING.XLS, but the unit models are arranged several to each page.

The parameters for the unit models (split factors, product size distributions) were manually calculated from the data presented in worksheet 'Size Distbns. (Meas)'. It should be noted that this measured data has some severe mass balance inconsistencies.

The flowsheet and measured data were part of the comminution simulation test set prepared for the International Comminution Research Association workshop, ICRA 95, by Dr R.P. King of the Comminution Center, University of Utah.



FLOTATN.XLS

The flowsheet in this workbook depicts a typical flotation circuit, consisting of a rougher and scavenger banks, a regrind mill and retreat bank, and three stages of cleaning.

This example demonstrates the following:

- conversion of assay data to mass flow data to allow solution using the **Limn:Solve** engine.
- The use of the two product formula as the basis of a simple flowsheet model of a flotation bank.

Worksheet 'Stream Data'

The stream data are arranged with the stream names across the top, and the component names to the left of a block of mass data (for use by **Limn:Solve**), and a block of data derived from the mass data (for ease of assimilation by the user).

The required user entered data are highlighted (white on blue).

Conversion of assay data to mass flowrate

The user entered raw data for the feed ('Stream Data'!\$D\$10:\$D\$13) must be converted to mass flowrate form. This is readily calculated from the entered total flowrate of solids and individual metal assays. Water in the feed is calculated from the entered feed %solids data.

Mass flowrate data for each component in the feed is all the stream data that **Limn:Solve** requires. During flowsheet solution, **Limn** writes the calculated mass data into the cells 'Stream Data'!\$E\$5:\$V\$8. Upon completion, this should represent a set of balanced flowsheet mass flowrate data.

Conversion of mass flowrate data back to assays

To convert the mass flowrate data in range 'Stream Data'!\$E\$5:\$V\$8 to assays, cell equations in the range 'Stream Data'!\$E\$5:\$V\$8 calculate percentage values.

Model for Unit: Rougher			
	Feed	Tail	Conc.
tph Pb	15.50	5.25	10.25
tph Fe	63.00	53.37	9.63
tph Gangue	421.50	378.47	43.03
tph Water	928.57	1311.26	94.37
tph Solids	500.00	437.09	62.91
% Solids	35.00	25.00	40.00
% Pb	3.10	1.20	16.30
% Fe	12.60	12.21	15.30
Model Parameters			
	<i>in Tail</i>	<i>in Conc.</i>	
<i>Required Pb Assay</i>	1.2	16.3	
<i>Required Fe Assay</i>		15.3	
<i>Required % Solids</i>	25.0	40.0	
<i>Required Water Addition (tph)</i>	477.06		

Unit Models

Flotation Banks

The 'flowsheet solution' models of flotation banks in worksheets 'Unit_Rougher', 'Unit_Scavenger', 'Unit_Retreat', and the three 'Unit_Cleaner_#X' worksheets, use the 2 product formula based on required Pb assays in Concentrate and Tailings to calculate an overall solids split.

From this split, the flowrates of solids to concentrate and tail are calculated.

The required concentrate assay for Fe, and the concentrate %Solids then define the other split factors.

The user entered tailings %Solids is used to determine the required feed water addition.

The required user entered data for each unit (e.g. 'Unit_Rougher!\$F\$17:\$G\$19') is highlighted (white on blue).

Regrind Mill

The regrind mill can have no effect on the components presented to it (mass flowrates of components), so the model merely copies the feed data to the product stream.

EFFCURVE.XLS

Modelling options in Limn

There are three main options for modelling in *Limn*, which may be used in combination to provide many more possibilities.

The workbook EFFCURVE.XLS gives an example of the use of each of these three options. There are four unit icons on the flowsheet, linked to four unit models, one written in the spreadsheet, one written using VBA, the macro language of Excel 5, and two written using higher level languages (one in Fortran, and one in C++).

Spreadsheet modelling

This is the most commonly used method of modelling for *Limn*, and the method which has been used in all of the previous examples in this Part of the manual.

In spreadsheet modelling, the mathematical description of the unit is expressed in cell formulae on a worksheet (or across multiple worksheets if necessary).

A group of cells is defined to hold the data which represents the components in the feed to the unit. (e.g. worksheet '[EFFCURVE.XLS]Spreadsheet Model'!\$C\$4:\$C\$12, where nine cells are allocated to hold the mass flowrate of material in each size fraction of interest in the feed).

If required, a group of cells is allocated to hold user entered model parameters (e.g. 'Spreadsheet Model'!\$C\$15:\$C\$17, where the cyclone efficiency curve parameters are entered).

By a combination of cell equations, the vector of feed data is transformed according to the properties of the model, producing one or more vectors of data which represent the mass flowrate of material in each size fraction of interest in each product.

In this example, the model is the Lynch/Rao hydrocyclone model, and the data is taken from Lynch '*Mineral Crushing and Grinding Circuits*', Elsevier, 1977, Example 6-4, pages 127-130.

The data in columns D through J represent the various steps necessary to calculate the component mass flows in the hydrocyclone underflow and overflow streams, given the entered values of α , d_{50c} , and C .

Ranges Spreadsheet Model'!\$K\$4:\$K\$12 and \$L\$4:\$L\$12 represent the underflow and overflow streams respectively.

The model defined in this way is a stand-alone model, not unlike a subroutine written in a conventional computer language. There are no cell equations relating the feed stream vector on this sheet to any other worksheets (**Limn:Solve** uses the flowsheet to provide this information at solution time.)

Why Wizards?

With the increasing complexity of modern software, many vendors have realised that there is a need for on-line assistance to provide user access to the features of a package, without requiring the user to become totally familiar with the details. Agents, Assistants, Tutors, are all names for software 'helpers' which guide a user through a complex operation. With Excel 5, Microsoft® introduced the term Wizard to describe its version of these 'helpers', and the technology (and name) is now used in many of the Microsoft® Office™ component programs.

Limn Wizards

There are several situations which arise when using *Limn*, where automation would provide significant benefits. The most obvious of these is in the definition of stream component data, and the linking of spreadsheet data ranges to the process unit icons and stream flow lines which make up process flowsheet. The *Limn* DataWizard which is part of the version 1.00 release of *Limn* provides that automation.

Also planned for release in early updates to version 1.00 of *Limn* are a DatablockWizard to automate the set-up of the process flowsheet information blocks, and a PrintWizard to assist in printing of flowsheets.

Where *Limn* is to be used in a specific modelling/flowsheet solution regime, potential exists for more complete automation. The ***CoalWizard Add-On for Limn*** (available separately) is one such package which handles many of the data set-up requirements for spreadsheet based flowsheet solution work in coal processing plants.

Introduction

After drawing a flowsheet using *Limn:Draw*, and before proceeding to *Limn:Solve*, a user must:

1. Define the component data required for the flowsheet solution.
2. Link the flowsheet unit and stream representations to their associated data elsewhere within the spreadsheet.
3. Build the flowsheet solution models required for each of the process units.

The *Limn* DataWizard assists in the first task, performs the second and provides a basis for the third task, removing much of the need for tedious and potentially error prone manual configuration.

Loading the DataWizard

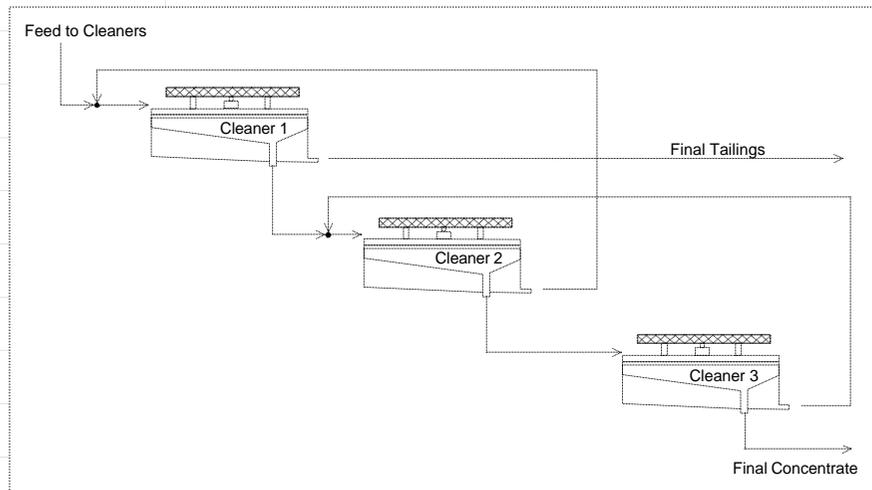
Load procedure

The *Limn* DataWizard should be loaded automatically during the normal *Limn* setup procedure. If the 'Configure Excel for *Limn* (set up toolbars)' option was checked on the main setup dialog will have automatically incorporated the *Limn* DataWizard toolbar and button into Excel.

Limn DataWizard Components

The *Limn* DataWizard consists of the file, LIMNWIZD.XLA, which should be present in the *Limn* installation (default *Limn*) subdirectory.

The DataWizard has its own Excel toolbar and button which install themselves the first time LIMNWIZD.XLA is loaded into Excel (See Part 1 of this manual for more information about loading Excel AddIns.)



Limn Data Wizard - Sheet 1 of 4

Select the required layout for component data:

Vertical Horizontal 2-Dimensions

In *Vertical* layout, stream names are displayed across the top of the page, with component names down the left side. *Horizontal* layout has stream names down the left side, with component names across the top. *2-Dimensions* results in separate component data blocks for each stream, repeating down the page.

In each layout, streams are all placed on a single page, while units are displayed individually on separate pages.

Help **Cancel** **< Back** **Next >** **Finish**

Using the DataWizard

1. Draw Flowsheet

Before the DataWizard can be used, **Limn:Draw** must be run to create a flowsheet. For this example, a multi stage flotation cleaning circuit has been chosen (see flowsheet opposite).

2. Start DataWizard

Having drawn a flowsheet, exit **Limn:Draw** and return to Excel. With the required flowsheet as the active worksheet, press the DataWizard button on the **Limn** DataWizard toolbar.



3. DataWizard Sheet 1 - layout for component data

The first dialog sheet in the DataWizard allows selection of the required layout for component data. Options are 'Vertical', 'Horizontal' and '2-Dimensions'. Options are chosen by clicking on the button showing the preferred layout. For the example, we have chosen option 1, 'Vertical' component layout. (See opposite)

4. DataWizard Sheet 2 - component name entry

Vertical & Horizontal name entry

Pressing the **Next** button on DataWizard Sheet 1 progresses the Wizard to Sheet 2. This sheet allows entry of the component or species name for the flowsheet solution. There are two types of components, those that play a direct part in the **Limn:Solve** solution, and which must be able to be summed at a joining node by simple addition (e.g. solids flowrate in t/h, liquid volume flowrate in m³/h), and those which are calculated from the above components using spreadsheet formula (e.g. % solids, assays etc.)



Limn Data Wizard - Sheet 2 of 4

Enter the **VERTICAL** component names:

Mass related components

Remember, Limn can only solve for properties which can be combined by simple addition.

Derived components

The derived components are calculated from the above mass type components.

Buttons: Insert ..., Edit ..., Remove, Move, Help, Cancel, < Back, Next >, Finish

Because we selected the 'Vertical' option on DataWizard Sheet 1, Sheet 2 refers to VERTICAL component names. (See opposite)

To enter mass related components, click the mouse in the top list box, then press the **Insert** button. A sub-dialog appears, asking for name entry.

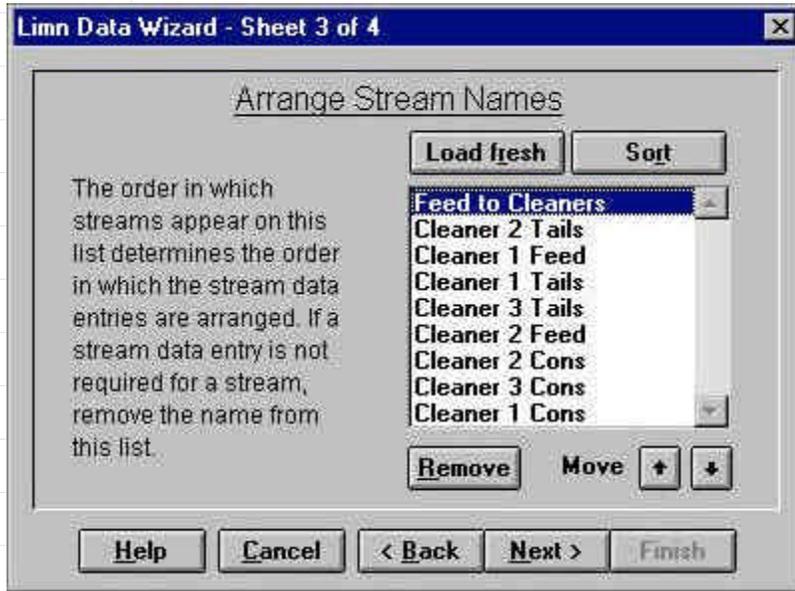


The **Edit** button may be used to modify existing names (select the name in the required name list and press the **Edit** button, then edit the old name in the name entry sub-dialog), and the **Remove** button used to delete them. Normal Windows™ editing keys work during entry of names, allowing copying from, or pasting of names into the name entry field on the name entry sub-dialog.

In the example, we wish to track the flowrates of solids and water, and the percentages of two metals in the ore minerals. To allow the **Limn:Solve** engine to solve for these, we must enter mass flowrates of each of the metals required, the mass flowrate of the remainder of solids in the stream, and the mass flowrate of water. These form the 'Mass related components for the solution. The derived components are thus the total solids flowrate (sum of all of the solids sub-species), assays of each metal (calculated from flows) and %Solids (calculated from total solids and water flowrates).

On completion of the entry of component names for the example, the component name list will contain:





2-Dimensional data

When the 2-Dimensions option is selected on Sheet 1, the number of sheets in the DataWizard is increased to 5, and two component name entry sheets are displayed in sequence. They allow entry of names for components to be displayed vertically (e.g. size distribution intervals) and those to be displayed horizontally (e.g. mineral species) respectively.

5. DataWizard Sheet 3 - Arrange Stream Names

This sheet allows the user to define the presentation order of process streams in the spreadsheet range allocated to stream data. (See opposite.) The **Load Fresh** button is used to copy all stream names from the flowsheet. This function is automatically carried out when Sheet 3 is accessed for the first time. The order of streams in the list after initial entry, or after **Load Fresh**, is the order in which they were placed on the flowsheet. Since this may not be the most useful order for data entry, the user has the option of re-arranging the stream names. This may be done by selecting the name in the list, and moving it up or down, using the **Move** arrow buttons. Alternatively, an alphabetic sort order may be imposed by pressing the **Sort** button.

The Remove button may be used to eliminate any stream from the list if it is not required. (The **Limn:Solve** engine maintains an internal copy of all stream data during solution. Removing a stream from this list stops the **Limn:Solve** engine writing the data back to the sheet after solution.)

For the example, a reasonable arrangement of stream names might be:



6. DataWizard Sheet 4 - Arrange Unit Names

The final sheet (see opposite) in the *Limn* DataWizard allows rearrangement of the process unit names in much the same manner as the previous sheet.

7. The DataWizard in action

After the component layout and component names have been specified, and the arrangement of stream and unit names is correct, the user may press the **Finish** button to have the *Limn* DataWizard carry out the required modifications to the Excel workbook.

When activated, the DataWizard creates a new sheet which contains an array of stream data, and a new sheet for each process unit requiring a model.

Stream Data Sheet

The stream data sheet created by the *Limn* DataWizard consists of an array of cells (Vertical and Horizontal component layout - multiple arrays in 2-Dimensional layout) which represents the process stream data for each entered component name, with space for user entered cell equations to represent derived properties. The sheet generated in the example is shown opposite. Note that since 'Vertical Component Layout' was selected on the DataWizard Sheet 1, that the stream names are arranged horizontally across the sheet, with the component names down the left hand side. There are two distinct blocks of data, those components used by the engine, and those components which are derived from them.

The *Limn* DataWizard links the flowsheet process stream flow-lines to these stream data ranges. If **Limn:Draw** is entered after the DataWizard has been run, examination of the stream detail data for streams will show that the correct StreamData sheet range has been allocated to each stream.. In Excel, 'clicking' the mouse on a process stream flow-line on the flowsheet, will cause a jump to the associated data range.

Unit Data Sheets

Limn DataWizard creates a separate unit data sheet for each process unit icon on the flowsheet. Each sheet is named 'Unit_' followed by the user specified name for that unit. Thus, in the example, three sheets would be created, 'Unit_Cleaner 1', 'Unit_Cleaner 2', and 'Unit_Cleaner 3'. Each sheet is a template of a user defined spreadsheet model, with the data ranges required to link that model to the process unit representation on the flowsheet.

The freshly created template for 'Unit_Cleaner 1' is shown opposite. Note that the derived data are also present on this sheet. This is not strictly necessary, since modelling must take place using the 'mass related' component data. In many cases, however, the derived data provides useful information which can assist to validate or debug the model. If the derived data are not required, the cell ranges may be deleted.

After the DataWizard

Calculation

The **Limn** DataWizard provides template sheets for the stream data, and for each of the unit models. It is then up to the user to specify the particular cell equations required to calculate the 'derived components', and to specify the spreadsheet models for the process units.

In the example, the derived components calculations would involve calculating the total 'tph Solids' from the individual component mass flowrates, and then calculating the solids and assay percentages from the individual flowrates and the calculated total solids flow.

For the process models, the user must specify in cell equations describing the manner in which the feed components are converted, split, transferred, or otherwise transformed into the components in each product stream. On the template worksheet, a 'User Modelling Area' has been reserved for entry of these equations, partition numbers, model parameters, or whatever the user decides is necessary.

Formatting

Formatting of the template sheets created by the *Limn* DataWizard is entirely at the discretion of the user. The only critical aspects of the sheet are the cell ranges representing the ‘mass related components’ on the StreamData sheet, and the corresponding Feed and Product stream ‘mass related components’ on the process unit model sheets. Because these are linked to the flowsheet representations of streams and units, care must be taken to ensure that any formatting (particularly cell insertion or deletion) does not destroy that linkage. (The linkage mechanism is the Excel ‘Named Range’. Each block of ‘mass related component’ data is described by a hidden range name derived from the flowsheet object name - for more information, see Part 4 of this manual, *Limn* Technical Reference, Chapter 3.)

Sources of Icons

The Excel drawing toolbar

Excel itself can be used to draw rudimentary process unit icons. An example later in this chapter shows how this may be done.

Icons drawn in this manner will generally be of lower quality than those drawn using a specialist drawing package.

Drawing with MS Word for Windows™

The drawing package within Word for Windows™ has several features which are lacking in the Excel equivalent. These include a drawing grid, and drawing object rotation. Drawings created in Word may be copied, and pasted into Excel for conversion to a process unit icon.

Other drawing packages

Any drawing package which is capable of copying a picture to the clipboard in Windows™ Metafile Format, or which can save to a Windows™ Metafile (typically with a WMF extension) can be used to create **Limn** process unit icons. There are many low priced examples of suitable packages. AutoSketch® release 2 for Windows™ from Autodesk is one which works well. The second example later in this chapter explains how to use Windows™ Metafile Format data from the clipboard to create an icon.

CAD packages

Most CAD packages which are intended for use in the Windows™ environment have the ability to copy data to the clipboard in Windows™ Metafile Format.



Tips: Drawing icons using Excel

1. It is easier to connect and align drawing components if a large scale is used (up to 400% zoom may be manually entered, even though the **Zoom Control** menu stops at 200%).
2. The grid-lines on a worksheet make useful guides for drawing alignment.
3. Double clicking a **Drawing** toolbar button 'latches' the button and allows that drawing object type (line, square etc.) to be repeated.

Windows Metafiles

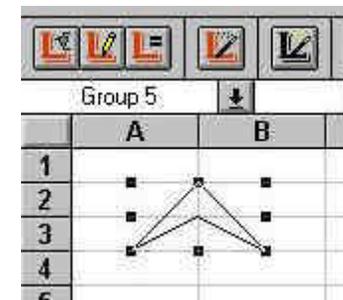
Existing Windows™ Metafiles may be used as a source of icons in two ways. They may be inserted into an Excel worksheet, using the menu **Insert Picture** command (see example 3 below). Alternatively, **Limn** will recognise files with a .WMF extent as Windows™ Metafiles during the **Limn** ‘Select a Unit Icon’ dialog, and will include them in the list of available icons as “_file: filename.wmf”. The disadvantage of this approach is that there are no default process unit names, or default stream connections associated with the icon.

Examples

1 - Using Excel drawing tools

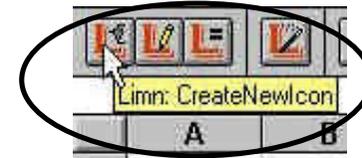
The following steps demonstrate how to add a process unit icon (a simple splitter icon) to the **Limn** package, when the icon is drawn using the Excel drawing tools.

1. Activate the Excel **Drawing** toolbar (Menu option **View, Toolbars**, check **Drawing**).
2. Draw the representation of the process unit (the icon) using a combination of the drawing tools.
3. Select all components of the new picture and **GROUP THEM TOGETHER** using the drawing toolbar **Group** button.



! The **Limn: CreateNewIcon** utility will only recognise an Excel **Group, Picture** or **Drawing Object** for conversion to an icon.

4. **SELECT** the new unit icon group, and click the **Limn: CreateNewIcon** button on the **Limn** toolbar.



5. A dialog box will appear, allowing selection of a directory and file location for the new icon Metafile, and allowing entry of a description (as displayed on the **Limn** Add Icon dialog) and a default name (the **Limn**/Excel object name). For the moment, ignore the fields at the bottom of the dialog referring to the default feed and product streams.



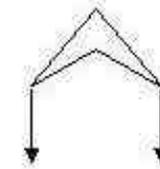
6. Select the directory and filename in which you wish to store the new icon.
7. Enter a description and default name for the Icon.
8. Press OK to close the 'New Icon and Icon File Details' dialog.
9. Open a new Excel workbook.



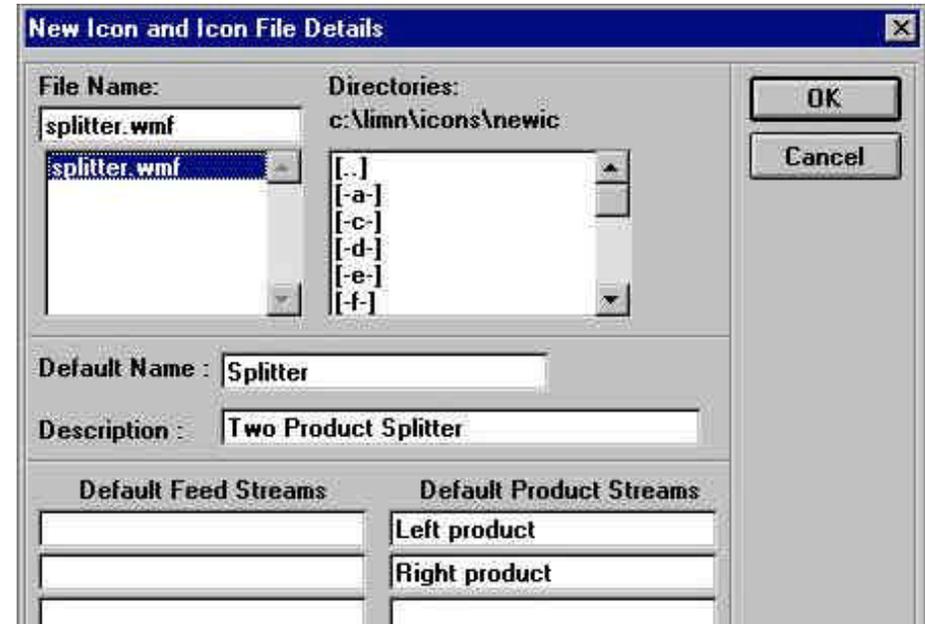
Tip: Aligning streams to unit icons

Zooming to 400% within ***Limn: Draw*** makes it easier to precisely align stream tags with the required point on the new unit icon picture.

10. Start **Limn: Draw** by selecting the appropriate toolbar button.
11. Within **Limn: Draw**, select the Add a New Unit Icon option, and navigate to the subdirectory in which the new icon was placed. The icon description entered in step 7 above should now be included in the list of available icons. Clicking on that description should show the new icon picture in the preview box.
12. Click OK to place the new icon on the flowsheet. Note that at this stage it has no streams attached.
13. Using the normal **Limn** stream draw methods, draw product stream tags from appropriate points on the new unit icon. These tags should be short single segment streams. Draw feed streams at this point too, if desired.



14. Exit **Limn: Draw** and return to Excel.
15. Select the new icon picture. (It will be necessary to use the mouse RIGHT button, or to hold the control key down while selecting the picture, or to use the Drawing Selection button on the Drawing toolbar. As it does for all unit icons and streams, **Limn** has assigned a data access macro to the unit icon picture on this sheet. A simple click will activate this macro, rather than selecting the drawing object.)
16. Click the **Limn: CreateNewIcon** button on the **Limn** toolbar.
17. The 'New Icon and Icon File Details' dialog will appear. Note that there are now names for each of the stream tags which were drawn using **Limn: Draw**.



18. Select the original file to save the new icon picture into (now with stream tag details included). Stream names and unit icon default name or description may also be updated at this time.

The addition of a new icon to **Limn** is now complete.

! The icon file (*filename.WMF*) may be moved to a suitable directory using normal windows file handling tools. If this is done, be sure to delete the file **ICONWMFS.LMN** if one exists in the directory into which the new icon file is moved.

The file **ICONWMFS.LMN** is an index file, used to speed up the display of icon descriptions in the 'Select a Unit Icon' dialog. Deleting the index file ensures that a new index, including the new unit icon, will be created next time the 'Select a Unit Icon' dialog is accessed.

2 - Using other drawing packages

The process for using other drawing packages to create new unit icon pictures for **Limn** is similar to that for using the Excel drawing tools.

1. Draw the unit icon picture in the preferred drawing package.
2. Copy the picture to the clipboard. (If necessary, specify that the clipboard format is to be Windows™ Metafile Format.)
3. Open a new Excel workbook.
4. Select Excel menu **Edit**, item **Paste Special**, and select the **Paste** option as **Picture**.
5. Select Excel menu **Format**, item **Object**. On the **Patterns** page of the Format Object dialog, select option None for both Line and Fill.
6. With the new unit icon picture still selected, proceed as per steps 4 and following in Example 1, above.

3 - Using Windows™ Metafiles

If a suitable picture already exists in a Windows™ Metafile file, it can be used as follows:

1. Open a new Excel workbook.
2. Select Excel menu Insert, item Picture.
3. Select the required Metafile file from the Picture dialog. (Preview may be used to check the contents of the file.)
4. Proceed as per step 5 and following in Example 2 above.



Please Note

The information in this section is provided in the interests of completeness and the "open" nature of the *Limn* package.

There is no need to manually edit the *Limn* "INI" file under normal operation.

Advanced users may wish to modify entries in this file to customise the default operation of *Limn*, or to access extended functionality.



Tip: "INI" file editing

"INI" files may be edited with any text editor which can save the file as simple text. Windows™ Notepad works well.

!!!WARNINGS!!!

The format of an "INI" file may be important.

Changes to the file may prevent the parent application from functioning correctly (or at all).

Make sure that a backup copy is made before attempting to edit any "INI" file.

Spaces in parameter strings may be critical. If manually modifying "INI" files, ensure that the current formatting is maintained.

Introduction

“INI” files

Microsoft® Windows™ programs use a special file to store user preference information (default settings, preferred screen layout etc.). Typically, these file use the name of the application, followed by the extension “.INI” (for “INItialisation file”), and are stored in the \WINDOWS subdirectory.

“INI” files are text files, consisting of a series of *SECTIONS*, each of which may have several *ENTRIES*. A section is defined by a heading which is simply some text enclosed in square brackets. Entries follow the section heading, and are of the form “*entry_text=data_text*”, where *data_text* contains the parameters associated with the entry.

Thus, the generic form for an “INI” file is:

```
[SECTION_1]
entry_1=data_1
entry_2=data_2
[SECTION_2]
entry_1=data_1
entry_2=data_2
```

“LIMN.INI”

The Initialisation file for **Limn: “The Flowsheet Processor”** is a file named LIMN.INI, present in the **Limn** subdirectory (default \LIMN). It contains various sections whose entries determine the default user interface options for the **Limn** package.



Parameter definitions for *Entry* "Patterns=" in *Sections*:

[Handles]
[UnitFormat]
[TextboxFormat]
[NodeFormat]
[LabelFormat]

All parameters refer to equivalent options on the various tabbed pages of the Excel **Format Object** dialog box

Bauto is a number from 0 to 2 specifying border settings.

If **bauto** is Border settings are
0 *Set by the user (custom)*
1 *Automatic (set by Microsoft Excel)*
2 None

Bcolor is a number from 1 to 56 corresponding to the 56 border colors in the Border tab.

Bstyle is a number from 1 to 8 corresponding to the eight border styles in the Border tab.

Bwt is a number from 1 to 4 corresponding to the four border weights in the Border tab.

If **bwt** is Border is
1 *Hairline*
2 *Thin*
3 *Medium*
4 *Thick*

Shadow is a logical value corresponding to the Shadow check box. If shadow is **1**, Microsoft Excel adds a shadow; if **0**, Microsoft Excel removes the shadow.

Aauto is a number from 0 to 2 specifying area settings (that is, the object's "surface area").

If **aauto** is Area settings are
0 *Set by the user (custom)*
1 *Automatic (set by Microsoft Excel)*
2 *None*

Aback is a number from 1 to 56 corresponding to the 56 area background colors in the Patterns tab.

Afore is a number from 1 to 56 corresponding to the 56 area foreground colors in the Patterns tab.

Apattern is a number corresponding to the area patterns in the Patterns tab. **Apattern** can be from 1 to 18.

Rounded is a logical value corresponding to the Round Corners check box. If **rounded** is **1**, the corners are rounded; if **0**, the corners are square.

Newui is a logical value that specifies whether to use the new user interface features introduced in Microsoft Excel 5.0. It should always be **1**.

INI file sections

The following pages describe the various sections which may be present in the LIMN.INI initialisation file. There are three general categories of section in the file:

1. Manual entry required
2. Automatically updated
3. Updated by user action in a dialog

Manual entry required

The INI file sections in this group contain entries which define default aspects of the **Limn** user interface and of **Limn** extended functionality which should rarely require modification. For this reason, no access to them is available from dialogs within **Limn**. Advanced users may wish to manually modify entries in these sections to change the user interface defaults, or to access the extended functionality.

Section [HitTest]

There is only one entry in this section.

Entry: "BaseSize="

This entry defines the 'sensitive area' around the mouse cursor which is used when determining which object is 'under' the cursor. It corresponds approximately to a number of points from the cursor 'hot spot'. The size of the HitTest spot also determines the size of the 'object selection', or 'sizing' handles on each Excel/**Limn** object.

The default entry is: BaseSize=2

Section [MACROS]

The entries in this section define two Excel Macro names which are used by **Limn** to control aspects of Excel operation.



***Tip: Manual changes to the Assigned
Macro name***

Although the Assigned Macro name for an Excel/*Limn* object may be manually altered (Menu **Tools**, option **Assign Macro**), so that a new macro is called on object 'selection', this is not particularly useful. If ***Limn:Draw*** is entered subsequent to the manual alteration, the altered macro name will be overwritten with the LIMN.INI default macro name during normal ***Limn*** exit processing.

Entry: "ObjectSelect="

This entry is used by **Limn** as the macro name assigned to any Excel/**Limn** objects on exit from a **Limn:Draw** session. A macro with this name is called whenever a user 'selects' the object in Excel.

The default entry is: OnLimnObjectSelect

OnLimnObjectSelect is a macro defined on a hidden VBA sheet created whenever **Limn** is run for the first time in a workbook. (See '*Hidden Sheets*' later in Part 5 of this manual - this Part).

The default action of this macro is to jump to the data associated with the Excel/**Limn** object, or to warn if no such data exists. There are two ways to specify alternative behaviour:

1. A different macro name may be specified in the LIMN.INI file. On subsequent **Limn:Draw** sessions, this macro will become the assigned macro for all Excel/**Limn** objects on the worksheet.
2. The hidden macro OnLimnObjectSelect may be modified to produce the desired behaviour. **Limn** checks for the presence of this macro, and will not overwrite it if it already exists within a workbook.

Entry: "SolveCallback="

If a macro name is specified using this entry, it will be called once during each iteration of the **Limn:Solve** engine. This facility is most often used to apply constraints to the solution.

The default entry specifies no call-back routine: SolveCallback=

If a call-back routine is specified e.g.:

[MACROS]

SolveCallback=MyCallbackRoutine

the macro routine (MyCallbackRoutine) must exist in the current workbook, or an Excel error will occur and a warning message will be displayed on each iteration of the **Limn:Solve** engine. A dummy routine consisting simply of:

```
Sub MyCallbackRoutine()  
End Sub
```

may be used if the functionality of the call-back routine is not initially required.

Automatically Updated

This group is used to store internal data from one **Limn** session to the next. There is only one INI file section in this category.

Section [IconPath]

Entry: "MRU="

This section and entry stores the full file system path to the location of the **M**ost **R**ecently **U**sed process unit icon. This allows the icon selection dialog to be opened at the location from which the most recent selection was made.

There is no default.

Updated by user action in a dialog

This group of INI file sections act as storage locations for user interface and object formatting preferences.

Section [DatablockFormat]

This section determines the default appearance of the ‘datablocks’ on a **Limn** flowsheet. There may be up to two entries.

Entry: “Pattern=“

This entry has 11 required parameters. The parameters are integer values, and must be in the following order: (see earlier table opposite *Section [Handles]* for meanings of each parameter)

bauto, bstyle, bcolor, bwt, shadow, aauto, apattern, afore, aback, rounded, newui

The default entry is: Pattern=1 1 1 1 0 1 1 1 1 0 1

Entry: “ExtentOption=“

This entry has one parameter which records the latest user selection on the Format Object dialog. It may have either of two values. A value of 1 means to apply formatting changes to all like objects on the **Limn** flowsheet. A value of 0 means to record the formatting changes for use as future default patterns.

The default entry is: ExtentOption=1

Section [NodeFormat]

This section determines the default appearance of the stream connecting ‘nodes’ on a **Limn** flowsheet. The entries are identical to those for Section [DatablockFormat] (above).

The default entries are: Pattern=0 1 1 1 0 0 1 1 1 0 1

ExtentOption=1



Parameter definitions for *Entry* "Font=" in *Sections*:

[TextboxFormat]

[LabelFormat]

All parameters refer to equivalent options on the various tabbed pages of the Excel **Format Font** dialog box

- Size_num** is the font size, in points.
- Bold** corresponds to the Bold item in the Font Style list box.
- Italic** corresponds to the Italic item in the Font Style list box.
- Underline** corresponds to the Underline check box.
- Strike** corresponds to the Strikethrough check box.
- Color** is a number from 0 to 56 corresponding to the colors in the Font tab; 0 corresponds to automatic color.



Parameter definitions for *Entry* "Align=" in *Sections*:

[TextboxFormat]

[LabelFormat]

All parameters refer to equivalent options on the Excel
Format Object - Alignment dialog.

X_align is a number from 1 to 4 specifying the horizontal alignment of the text.

X_align	Horizontal alignment
1	<i>Left</i>
2	<i>Centre</i>
3	<i>Right</i>
4	<i>Justify</i>

Y_align is a number from 1 to 4 specifying the vertical alignment of the text.

Y_align	Vertical alignment
1	<i>Top</i>
2	<i>Centre</i>
3	<i>Bottom</i>
4	<i>Justify</i>

Orient_num is a number from 0 to 3 specifying the orientation of the text.

Orient_num	Text orientation
0	<i>Horizontal</i>
1	<i>Vertical</i>
2	<i>Upward</i>
3	<i>Downward</i>

Auto_size corresponds to the Automatic Size check box. Automatic size makes the border fit exactly around the text no matter how you change the text.

Entry: "Align="

This entry records the user preferred text alignment for 'labels' on the **Limn** flowsheet.

This entry has 4 required parameters which are all integer values. The parameters - in order - are: (see table opposite for the meanings of each parameter).

x_align, y_align, orient_num, auto_size

The default entry is: Align=2 2 0 0

Section [TextboxFormat]

This section determines the default appearance of the 'textbox' unit icons on a **Limn** flowsheet.

Entry: "Pattern="**Entry: "ExtentOption="****Entry: "FontName="****Entry: "Font="****Entry: "Align="**

These entries are identical to those for Section [LabelFormat] (above).

The default entries are:

Pattern=2 1 1 1 0 2 1 1 1 0 1

ExtentOption=1

FontName=Arial

Font=10.00 0 0 0 0 1

Align=2 2 0 0

Entry: "DefaultText="

This entry defines the default text displayed in the text box unit icon.

The default entry is: DefaultText=Your Text Here

Section [Grid]

This section stores the user preferences for the drawing grid. The grid show and snap states may be independently manipulated.

Entry: "ShowGrid="

This entry requires one integer parameter. **1** means show the grid, **0** means hide it.

The default entry is: ShowGrid=0

Entry: "SnapToGrid="

This entry requires one integer parameter. **1** means force objects to align (Snap) to the grid, **0** means allow free placement.

The default entry is: SnapToGrid=0

Entry: "SizingIndex="

This entry requires one integer parameter. The number in the range 1 to 5 refers to the required grid interval, i.e.:

Sizing Index	Grid Interval
1	<i>Very Coarse</i>
2	<i>Coarse</i>
3	<i>Medium</i>
4	<i>Fine</i>
5	<i>Very Fine</i>

The default entry is: SizingIndex=4

Section [Solution]

This section stores the user preferences for the **Limn:Solve** engine, as entered from the **Solution Options** dialog accessed via the **Limn:Draw** menu **Solve**.

Entry: "UpdateInterval="

This entry requires one integer parameter which represents the number of solution iterations between successive screen updates.

The default entry is: UpdateInterval=1

Entry: "UpdateOption="

This entry requires one integer parameter which represents the **Screen Update** option button selection on the **Solution Options** dialog, where:

Screen Update Option	Means
1	<i>Do not display</i>
2	<i>Display after first iteration</i>
3	<i>Display after solution convergence</i>

The default entry is: UpdateOption=3

Entry: "LimitIndex="

This entry requires one integer parameter which represents the index into the convergence limit menu. The convergence limit decreases by a factor of 10 for each unit index increase, based on a start value of 10^{e+4} , i.e. an index of 1 represents a convergence limit of 10^{e+4} , an index of 2 represents a convergence limit of 10^{e+3} , an index of 3 represents a convergence limit of 10^{e+2} , and so on. An index of 10 represents a convergence limit of 10^{e-5} .

The default entry is: LimitIndex=10

Entry: "DiagnosticOption="

This entry requires seven integer parameters. The first parameter represents the **Diagnostic / Error Display options** option button selection on the **Solution Options** dialog, where:

Diagnostic / Error Display option	Means
1	<i>Do not display</i>
2	<i>Display after first iteration</i>
3	<i>Display after solution convergence</i>

Parameters 2 to 4 represent the state of the check boxes in the **Warning options** sections of the **Solution Options** dialog, where:

Parameter	If set to 1 means
2	<i>Warn if units have no feed streams</i>
3	<i>Warn if streams come from no unit</i>
4	<i>Warn if streams have no data ranges</i>

Parameters 5 to 7 represent the state of the check boxes in the **Diagnostic display options** sections of the **Solution Options** dialog, where:

Parameter	If set to 1 means
5	<i>Display unit/stream connections</i>
6	<i>Display unit data ranges</i>
7	<i>Display stream data ranges</i>

Setting the parameters to zero disables the corresponding warning or diagnostic display option.

The default entry is: DiagnosticOption=3 1 1 1 0 0 0

Introduction

Limn functions by using the component objects of Excel, and has no native data or objects of its' own. It is thus necessary for *Limn* to impose a set of naming conventions over certain Excel resources in order to determine which aspects of an Excel workbook 'belong' to *Limn*.

The two distinct areas where *Limn* imposes its own naming conventions are on graphical objects used to make up the *Limn* flowsheet, and spreadsheet data ranges used to hold the process data relating to those objects.

In order to distinguish *Limn* specific names from other names in a workbook, they are prefixed by a four character code. The first character is a character such an underscore or back slash, which is used to divide the names into classes, and which is chosen to minimise the likelihood of name clashes with normal Excel user defined names. The second character is an alpha object type identifier, and the third and fourth characters are numeric indices, specific to the object type.

Graphical object names

All Excel drawing objects have an associated name. Unless changed by the user, this name will consist of an identifier (e.g. 'Line', 'Rectangle', 'Oval', 'Picture' etc.) followed by an index (a number which increments by one for each graphic object on the worksheet). Users may change the name of any graphics object, either from Macro code, or by overwriting the default name in the 'Name Box' (The Box on the left side of the formula bar - for more information, search Excel On-Line Help, topic 'Name Box')

Excel drawing object names are limited to 32 characters in length, and must consist only of the characters a to z, A to Z, 0 to 9, space and underscore.



***Limn* naming convention for Flowsheet Graphical Objects**

Flowsheet object	Prefix	Valid codes (<i>nn</i>)	Meaning
'Picture' unit icon	<i>_Unn</i>	00	Unit is unrotated
		01	Unit is flipped top to bottom
		02	Unit is flipped side to side
		03	Unit is flipped top to bottom and side to side
'Textbox' unit icon	<i>_T00</i>	00	none
'Datablock'	<i>_D00</i>	00	none
'Label'	<i>_L00</i>	00	none
'Stream'	<i>_Snn</i>	00-99	00 = segment 1 01 = segment 2 02 = segment 3 etc.
'Node' joiner	<i>_N00</i>	00	none

***Limn* naming convention for Flowsheet Graphical Objects**

To enable *Limn* to use the various Excel drawing objects as components of a process flowsheet, a specific naming convention is employed.

Limn drawing object names are always prefixed by a four letter identifier of the form: ***_ann***, where ***a*** is an alpha character identifying the type of object, and ***nn*** are numeric codes, whose meaning depends on the alpha identifier.

The table opposite provides details.

The remainder of a *Limn* drawing object name consists of the user name given to the object. Thus, a unit ‘picture’ type icon which has not been rotated, and which has been given the name ‘Flotation Bank’ by the user, will be assigned an Excel name of ‘***_U00Flotation Bank***’. If the unit had been flipped side to side, the name would be ‘***_U02Flotation Bank***’. Similarly, the first segment in a stream named ‘Concentrate’ would be given an Excel name of ‘***_S00Concentrate***’, while the third segment of such a stream would be named ‘***_S02Concentrate***’.

Since the *Limn* prefix is 4 characters long, and the maximum length of Excel drawing object names is 32 characters, the length of user defined names for *Limn* flowsheet graphics objects is limited to a maximum of 28 characters. The character set for user defined names in *Limn* is also limited to the Excel drawing object name set detailed above.

Data range names

When the *Limn:Solve* engine attempts to solve a flowsheet, it requires access to the spreadsheet data ranges containing the component data for each stream as well as the location of the data which represents the feed and products of any spreadsheet unit models. These data are linked to their graphical representation on the flowsheet by the application of Excel ‘Range Names’ to the spreadsheet data ranges (for more information on range names, search Excel On-Line Help, topic ‘ranges names’). The name used to define the *Limn* data range for an object incorporates the user defined name for that particular object.



Tip: Range Names

Excel range names cannot include a space character. For this reason, range names used to define ranges for *Limn* data have any spaces in the user defined name replaced with the underscore character.

Limn naming convention for Spreadsheet Data Ranges

Object	Prefix	Valid codes (<i>nn</i>)	Meaning
'Stream'	\S <i>nn</i>	01	01 = stream data
Unit (Spreadsheet model)	\U <i>nn</i>	01	Unit Feed data
		02	Unit Product 1 data
		03	Unit Product 2 data
		04	Unit Product 3 data
		05	Unit Product 4 data
		06	Unit Product 5 data
		07	Unit Product 6 data
Unit (User DLL model)	\U <i>nn</i>	21	Model parameter data
		22	DLL library name
		23	Model name in DLL
		24	always = 'UserDLL'
		25	Model return data

Limn naming convention for Spreadsheet Data Ranges

Limn spreadsheet data range names are always prefixed by a four letter identifier of the form: ***lann***, where ***a*** is an alpha character identifying the type of object with which the range is associated, and ***nn*** are numeric codes, whose meaning depends on the alpha identifier. The table opposite provides details.

The remainder of a ***Limn*** spreadsheet data range name consists of the user name given to the object, *with spaces replaced with underscores*. Thus, the data range representing the feed to a spreadsheet model of a unit named 'Scalping Screen' by the user, will be assigned an Excel Range Name of '***U01Scalping_Screen***'. The data calculated by the spreadsheet model for the first product from such a unit would be found in the range referred to by the name '***U02Scalping_Screen***'.

If a unit named 'Dewatering Cyclone' had been assigned a User model, resident in a Dynamic Link Library (DLL), the parameters for the model would be found in the range referred to by the Range Name '***_U021Dewatering_Cyclone***'.

The component data Range Name for a stream named 'Circuit Feed' would be '***_S01Circuit_Feed***'.

Hidden names

The Spreadsheet Data Range names referred to in this section have been implemented in ***Limn*** as sheet specific hidden names. This means that they are not accessible from the Excel menu **Insert Name Define** dialog. They are, however, accessible from VBA macro code. The following subroutine will list all of the names in the active workbook, and list the ranges or data they refer to:

```
Sub ListNames()
    Dim aName As Object
    Dim i As Integer
    i = 1
    For Each aName In ActiveWorkbook.Names
        ActiveCell.Offset(i, 0) = aName.Name
        ActiveCell.Offset(i, 1) = " " & aName.RefersTo
        i = i + 1
    Next aName
End Sub
```


'Temporary' names

Limn:Solve

Two names are created by the ***Limn:Solve*** engine for use by user written VBA Macros during the flowsheet solution. They are only valid for the life of the ***Limn:Solve*** engine.

\IterationCount contains the number of times the engine has calculated all units and all streams, i.e. the cycle, or iteration number. After the engine completes, this name contains 0.

\Convergence contains the current convergence factor, i.e., the sum of the absolute differences between all components in all streams on successive cycles or iterations of the ***Limn:Solve*** engine. After the engine completes, this name contains the largest possible positive double precision number.