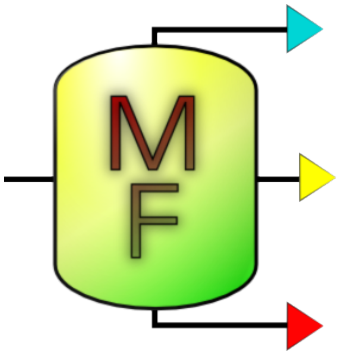
User Guide for

Multiflash Excel Interface

Infochem/KBC Advanced Technologies ltd

Version 6.1

Jul16 2015



Infochem/KBC Advanced Technologies ltd
42-50 Hersham Road
Walton on Thames
Surrey
KT12 1RZ
UK
Tel: +44 (0) 1932 242 424
e-mail: software@kbcat.com

This User Guide and the information contained within is the copyright of KBC Advanced Technologies ltd © 2015.

Disclaimer

While every effort has been made to ensure that the information contained in this document is correct and that the software and data to which it relates are free from errors, no guarantee is given or implied as to their correctness or accuracy. Neither KBC Advanced Technologies ltd nor any of its employees, contractors or agents shall be liable for direct, indirect or consequential losses, damages, costs, expenses, claims or fee of any kind resulting from any deficiency, defect or error in this document, the software or the data.

Contents

Contents iii
Installation 1
Introduction 1
Hardware requirements 1
Installing the Multiflash program 1
Installing KBC SLM security system 1
Installing the RLM software security system 2
USB dongle security device 2
Environment variables 2
Databank path 2
Installing the Multiflash Excel interface 3
Removing previous versions of the Multiflash add-in 3
64 bit versions of Excel 4
Testing the Multiflash Excel Interface 4
Removing Multiflash from your computer 5
Help 5
New features and changes in Version 6.1 7
Introduction 7
Calculations 7
General information 9
The Multiflash Excel interfaces 9
Different editions of Excel 9
Different versions of Excel 9
Using the Multiflash Excel interface 11
Getting started 11
Initialization scripts 15
Creating an initialization script directly in Excel 16
Creating an initialization script using the Multiflash GUI 17
Updating worksheets 17
Entering Multiflash functions 18
Cell references 20
Relative cell references 20
Absolute cell references 20
Mixed cell references 20
Setting up flash functions as array functions 20
Defining start values for array flash functions 21
Defining a flash function as an array function in Excel 22
Setting up the phase envelope functions 22
Multiflash Excel functions 23
General functions 23
Multiflash version number (MF_VERS) 23
BIP parameter (MF_BDAT) 23
Sum of squares of N cells (MF_SUMSQN) 24
Bubble and dew point functions 24
Bubble point at fixed pressure (MF_PBUBF) 24
Bubble point at fixed temperature (MF_TBUBF) 24
Dew point at fixed pressure (MF_PDEWF) 24
Dew point at fixed temperature (MF_TDEWF) 25
Component functions 25
Component name (MF_COMP) 25
Component feed composition (MF_FEED) 25
Component properties (MF_PDAT) 25
Temperature-dependent component properties (MF_PDAT_T) 31

Fixed phase fraction functions 32
Fixed phase fraction flash at fixed pressure (MF_PFRACF) 32
Fixed phase fraction flash at fixed temperature (MF_TFRACF) 32
Fixed wax phase fraction flash at fixed pressure (MF_PWAT) 33
Tolerance calculation for fixed phase fraction (MF_FRACT) 33
Flash functions 33
Flash at fixed pressure and temperature (MF_PTF) 33
Flash at fixed pressure and enthalpy (MF_PHF) 33
Flash at fixed temperature and enthalpy (MF_THF) 33
Flash at fixed pressure and entropy (MF_PSF) 33
Flash at fixed temperature and entropy (MF_TSF) 34
Flash at fixed pressure and volume or density (MF_PVF) 34
Flash at fixed temperature and volume or density (MF_TVF) 34
Flash at fixed pressure and internal energy (MF_PUF) 34
Flash at fixed temperature and internal energy (MF_TUF) 34
Flash at fixed internal energy and volume or density (MF_UVF) 34
Flash at fixed enthalpy and volume or density (MF_HVF) 34
Flash at fixed enthalpy and entropy (MF_HSF) 34
Flash at fixed entropy and volume or density (MF_SVF) 35
Flash function arguments 35
Phase envelope functions 40
Phase envelope (MF_PHENV) 40
The column headings for phase envelope (MF_PHENV_COLS) 41
Fixed enthalpy line (MF_HPHENV) 41
Fixed entropy line (MF_SPHENV) 42
Fixed internal energy line (MF_UPHENV) 43
Fixed volume line (MF_VPHENV) 44
Salt calculator functions 44
Multiflash ion names (MF_ION_INAM) 45
Ion ratio calculator for defined salinity – ion concentrations as input (MF_ION_IC) 45
Ion ratio calculator for defined salinity - salt mass fractions as input (MF_ION_MASS_FR) 46
Ion ratio calculator for defined salinity - salt molalities as input (MF_ION_MOLAL) 47
Ion ratio calculator for defined salinity - salt mole fractions as input (MF_ION_MOLE_FR) 48
Ion ratio calculator for defined salinity - total dissolved salt as input (MF_ION_TDS) 49
Salinity calculator – ion concentrations as input (MF_SAL_IC) 50
Salinity calculator – salt mass fractions as input (MF_SAL_MASS_FR) 51
Salinity calculator – salt molalities as input (MF_SAL_MOLAL) 51
Salinity calculator – salt mole fractions as input (MF_SAL_MOLE_FR) 52
Salinity calculator – total dissolved salt as input (MF_SAL_TDS) 53
Units and conversion functions 54
Input units (MF_IU) 54
Output units (MF_OU) 54
Unit conversion for density (MF_UND) 55
Unit conversion for enthalpy (MF_UNH) 56
Unit conversion for entropy (MF_UNS) 57
Unit conversion for pressure (MF_UNP) 58
Unit conversion for surface tension (MF_UNST) 59
Unit conversion for temperature (MF_UNT) 59
Unit conversion for thermal conductivity (MF_UNTC) 60
Unit conversion for viscosity (MF_UNVIS) 61
Troubleshooting 63
#N/A 63
#NULL! 63
#NUM! 63
Single occurrences of #NUM! 63

Multiple occurrences of #NUM! 64
#VALUE! 64
Troubleshooting other problems 65
Calculation of physical properties 67
Calculating pure component properties 67
Constant physical properties and temperature-dependent correlation coefficients 67
Temperature-dependent properties of a pure component or petroleum fraction 68
Example Excel spreadsheets 69
Example Excel spreadsheets 69
Pure component properties 69
Stored properties of a pure component 70
Phase envelope 70
Linked flashes 71
Stream types 72
UNIFACFIT.xls 73
Using the spreadsheet 73
VLEFIT.xls 74
Using the spreadsheet 75
Interfacial tension 76
PVT analysis 76
Index 79

Continue
to the next
section

Installation

Introduction

This chapter contains information that will help you install and run Multiflash with Microsoft® Excel® on a Windows® system. The 32-bit version of the Multiflash Excel interface works with Microsoft Excel 97 and later versions. The 64-bit version of the Multiflash Excel interface works with Microsoft Excel 2010 and later versions.

To run Multiflash on a Windows system, you must install a Multiflash license server with a valid Multiflash license or a Sentinel Software Protection driver and server for the security device. For further details, see the *Multiflash Installation Guide*.

Hardware requirements

The minimum requirement to run Multiflash is a computer capable of running Windows XP (or more recent versions). A USB port is required if the Multiflash license is controlled by a security key.

The program itself and associated files require about 200 Mb of disk storage.

Installing the Multiflash program

To install Multiflash on your computer, run the Setup program as described in the *Multiflash Installation Guide*. Setup installs the following components:

- Multiflash
- The Help system
- Infochem databanks
- The standard model configuration files
- Sample mixture files
- The Excel interface and associated files.

Installing KBC SLM security system

Since Multiflash 4.4 the KBC SLM security system is supported to control licensing. As of Multiflash 6.0, this is the default security system. For single user licenses, you need only copy the license file to the location specified in the Installation Manual.

For network license users, the Sentinel license server must be installed in the license server machine. In the users' machines the KBC License tester can be used to set license server.

For more details, refer to the *Multiflash Installation Guide* and to the KBC License Tester Manual.

Installing the RLM software security system

Multiflash 6.0 still supports the RLM security system for existing users of that system.

The Multiflash license manager and RLM license serverare now installed using a separate installer.

More information on how to run and use the RLM Multiflash license manager program is described in the section *RLM license system installation and management* in the *Multiflash Installation Guide*.

USB dongle security device

Since Multiflash 6.0, the legacy SafeNet dongles issued by Infochem are no longer supported. Dongles that are supported by the KBC SLM system can now be used. More information can be found in the *Multiflash Installation Guide*.

Environment variables

The variable Infochem_License is used to control which licensing system should be used. If it empty or non-existent, the KBC SLM is used, otherwise the RLM system is used.

The variables LSHOST and LSFORCEHOST can be used to set the KBC SLM license server. For more information refer to the KBC License Tester manual.

Databank path

Multiflash uses the data files shown in the table.

File	Description
Infodata.bin	Databank for fluid components
Infodata.idx	Infodata databank index
Mflash.msg	Displays for Multiflash GUI
Mflash.mix	Mflash message index
Mflash.err	Error message file
Mflash.eix	Error message index
Dippr.bin	DIPPR databank
Dippr.idx	DIPPR databank index
Infobips.bin	BIP databank (VLE)
infobips.idx	BIP databank index (VLE)
Infollbips.bin	BIP databank (LLE)
infollbips.idx	BIP databank index (LLE)

These files are normally installed in the same folder as the Multiflash program.

Installing the Multiflash Excel interface

The Excel interface files are copied to the Multiflash installation folder by the Multiflash installation program.

If you have a previous version of the Multiflash add-in already installed, you must first remove it from Excel, see *Removing previous versions of the Multiflash add-in* on page 3.

To install the Multiflash add-in for Excel versions prior to 2007:

1. Start Excel.
2. Choose **Tools > Add-Ins**.
3. Click **Browse**.
4. Navigate to the folder in which you installed Multiflash (by default C:\Program Files\KBC\MF61\ or C:\Program Files(x86)\KBC\MF61\ for 64-bit Windows versions).
5. Do one of the following:
   - If you have a 32-bit version of Excel, select MFXLPVT32.xll, and then click **OK**.
   - If you have a 64-bit version of Excel, select MFXLPVT64.xll, and then click **OK**.

     Tip: To find out which version of Excel is running, see *64 bit versions of Excel* on page 4.

   Result: The Multiflash-Excel XLL is added to the list of Add-ins available and has a check mark next to it.
6. Click **OK**.

   Result: The "Multiflash Add-in" menu item is added to the Excel menu bar.

To install the Multiflash add-in for Excel versions 2007 and later:

1. Start Excel.
2. Choose **File > Options**.
3. On the Excel Options dialog box, click **Add-ins**.
4. In the **Manage** list, select **Excel Add-ins**, and then click **Go**.
5. Click **Browse**.
6. Navigate to the folder in which you installed Multiflash (by default C:\Program Files\KBC\MF61\ or C:\Program Files(x86)\KBC\MF61\ for 64-bit Windows versions).
7. Do one of the following:
   - If you have a 32-bit version of Excel, select MFXLPVT32.xll, and then click **OK**.
   - If you have a 64-bit version of Excel, select MFXLPVT64.xll, and then click **OK**.

     Tip: To find out which version of Excel is running, see *64 bit versions of Excel* on page 4.

   Result: The Multiflash-Excel XLL is added to the list of Add-ins available and has a check mark next to it.
8. Click **OK**.

   Result: The **Add-ins** ribbon in Excel now contains the **Multiflash Add-In** menu.

Removing previous versions of the Multiflash add-in

If you have a previous version of the Multiflash add-in already installed, you must remove it from Excel prior to adding the new version.

To remove the Multiflash add-in for Excel versions prior to 2007:

1. Start Excel.
2. Choose **Tools > Add-Ins**.
3. Clear the check box to the left of Multiflash-Excel XLL, and then clock **OK**.
4. Restart Excel.

   Result: The MF menu item is no longer available on the Excel menu bar.

To remove the Multiflash add-in for Excel versions 2007 and later:

1. Start Excel.
2. Choose **File > Options**.
3. On the Excel Options dialog box, click **Add-ins**.
4. In the **Manage** list, select **Excel Add-ins**, and then click **Go**.
5. Clear the check box to the left of Multiflash-Excel XLL.

   Note: There is a version number after XLL.
6. Click **OK**.

   Result: The **Multiflash Add-In** menu disappears from the **Add-ins** ribbon. If there are no other add-ins on the ribbon, the **Add-ins** ribbon disappears.

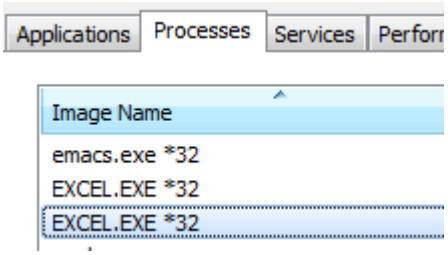
64 bit versions of Excel

Since Version 4.4 of Multiflash, 64bit versions of Excel are supported as well.

To find which version of Excel is running:

1. Start Excel.
2. Run the task manager and click the Processes tab.
3. Search for Excel.exe in the image name.

   If EXCEL.EXE is shown with a "*32" next to it, the 32-bit version is running, otherwise it is the 64-bit version.



Testing the Multiflash Excel Interface

After installing the Excel add-in, we recommended that you test that it is operating correctly.

To test the Excel add-in:

1. Start Excel.
2. Open the sample file test.xls, which is installed in the MFL Files folder of your Multiflash installation.

   This is a simple example of a flash calculation at a specified temperature and pressure.
3. On the Excel Add-Ins ribbon, choose **Multiflash Add-In > Load**.

   Result: A message giving the version of the Multiflash Excel add-in appears.
4. Click **OK**.

   Result: A dialog box appears with the message 'Please enter the range containing the initialisation string'.
5. Drag in the spreadsheet to select the range $A$1:$A$6, or type the range in the dialog box.

   Result: A dialog box appears with the message 'Multiflash initialized successfully'.
6. If the add-in has successfully loaded you should be able to change the input conditions (temperature, pressure, composition) and the outputs will be recalculated. If the results are not recalculated check the calculation tab on the Tools/Options menu. If manual recalculation is selected you need to press F9 to make the spreadsheet recalculate.

If Multiflash is not successfully initialized, a list of error and warning numbers and messages appears. If only numbers are displayed, the error message files MFLASH.err and MFLASH.eix cannot be found. To look up error numbers, on the Excel Add-Ins ribbon, choose **Multiflash Add-In > Help**.

Removing Multiflash from your computer

To uninstall Multiflash:

1. If the Excel add-in is installed, remove it, see *Removing previous versions of the Multiflash add-in* on page 3.
2. Uninstall Multiflash using the Windows Control Panel.

Help

Help is provided in three ways:

- The printed User Guide
- Online Help
- Technical support.

The *Multiflash Excel Interface User Guide* explains how to use Multiflash with Excel in detail and provides several case studies and a simple tutorial.

The Online Help is a standard Windows help system, providing both context-sensitive and procedural help with the Multiflash functions in Excel. If an error code is returned in the Excel interface there is no text associated with it. You can look up the meaning of the error code by selecting the "Multiflash Error Codes" option from the HELP menu.

If you do not find the information that you need in the online help or user guide, you can contact us by email, fax or telephone. The contact details are listed below.

Infochem/KBC Advanced Technologies ltd
42-50 Hersham Road
Walton on Thames
Surrey
KT12 1RZ
UK
Tel: +44 (0) 1932 242 424
e-mail: software@kbcat.com

New features and changes in Version 6.1

Introduction

A number of small improvements have been made to the Excel interface. For the new features of Multiflash 6.1, refer to the Multiflash Online help.

Calculations

It is now possible to do HV flashes by using the MF_HVF function.

Continue
to the next
section

General information

The Multiflash Excel interfaces

The Multiflash Excel interface provides spreadsheet functions that can be used to configure Multiflash and carry out calculations. The 32-bit interface is supported by an Excel add-in called MFXLPVT32.XLL, whereas the 64-bit version of the interface is called MFXLPVT64.XLL. This add-in, in turn, uses the Multiflash DLL (MFPVT32.DLL or MFPVT64.DLL) to perform the physical property calculations.

Different editions of Excel

The Multiflash Excel add-in has been designed as far as possible to work with all language editions of Excel. However, there are minor differences in the way the Multiflash functions are specified.

The function definitions in the manual use the conventions of the English language editions of Excel, where the full stop or period denotes a decimal place and the comma denotes an argument separator when defining functions. In many other language editions of Excel, the comma denotes a decimal point and the semi-colon (;) denotes an argument separator. In these editions, wherever you see a comma in the definition of a Multiflash function in the manual, you must replace it with a semi-colon.

Different versions of Excel

The screen shots used as illustrations in this manual have been taken from different versions of Excel. Different versions of Excel appear slightly different visually; however, the functionality described in the manual should be the same. Also the “MF” menu was replaced with “Multiflash Add-In”.

Continue
to the next
section

Using the Multiflash Excel interface

Getting started

You use Multiflash in Excel by using the following general process:

1. Create an initialization script to specify your fluid and the model that you want to use for your calculations.
2. Create an array (table) of values that specify the properties, conditions and component numbers that you want to consider.
3. Enter one or more Multiflash functions to perform calculations on your fluid.

This introductory example demonstrates using pre-defined files and commands to perform a fixed temperature and pressure flash calculation (PT flash), followed by an isenthalpic (PH flash) calculation, for a ternary mixture of ethane, butane and water using the RKS equation of state.

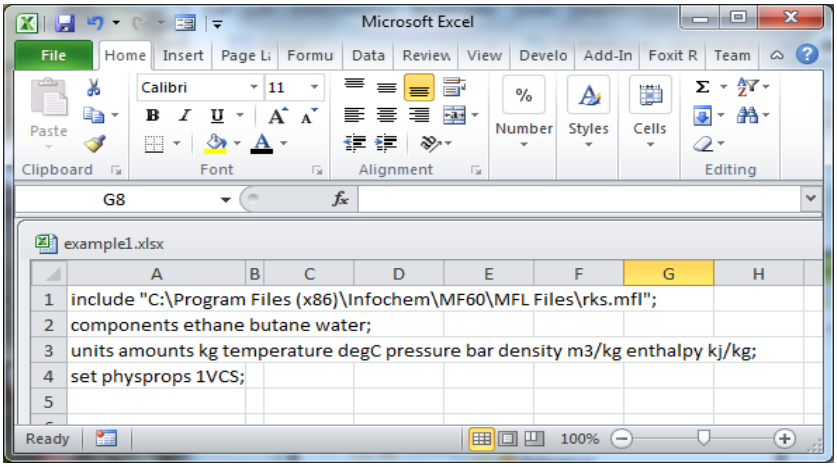
To create a simple Multiflash calculation in Excel:

1. In a new worksheet, enter the commands shown in the first column of the table, into cells A1–A4.

   Requirement: The Multiflash commands must be entered in a contiguous set of cells, which can be in either a column or a row.

Cell	Command	Description
A1	include "c:\Program Files\KBC\MF61\MFL Files\rks.mfl";	The Multiflash configuration file provided with the software for setting Multiflash to use the RKS equation with the Infochem databank. **Note:** The path shown is the default location. If you have installed Multiflash in a different location, enter that full path instead.
A2	components ethane butane water;	The list of components that comprise your fluid. Each component should be referred to by one of its names in the Infochem databank. A complete list of names is given in Multiflash for Windows. **Restriction:** Each Excel cell can accept up to 255 characters. If you require additional space for your components, continue adding component names in the next cell of the initialisation script. You indicate the last component by following its name with a semi-colon, which signifies the end of the components command. Multiflash can accept up to 200 components.

Cell	Command	Description
A3	units amounts kg temperature degC pressure bar density m3/kg enthalpy kj/kg;	The units used for your calculations. The full range of options can be seen by selecting the units command in the Multiflash GUI. If you omit the units command, Multiflash uses its default units for all quantities.
A4	set physprops 1VCS;	The level of physical properties that the Excel interface can calculate. By default, the interface does not calculate enthalpy; this command ensures that enthalpy can be calculated.



2. To specify a table for your flash calculation data, enter the values shown in the table into your worksheet.

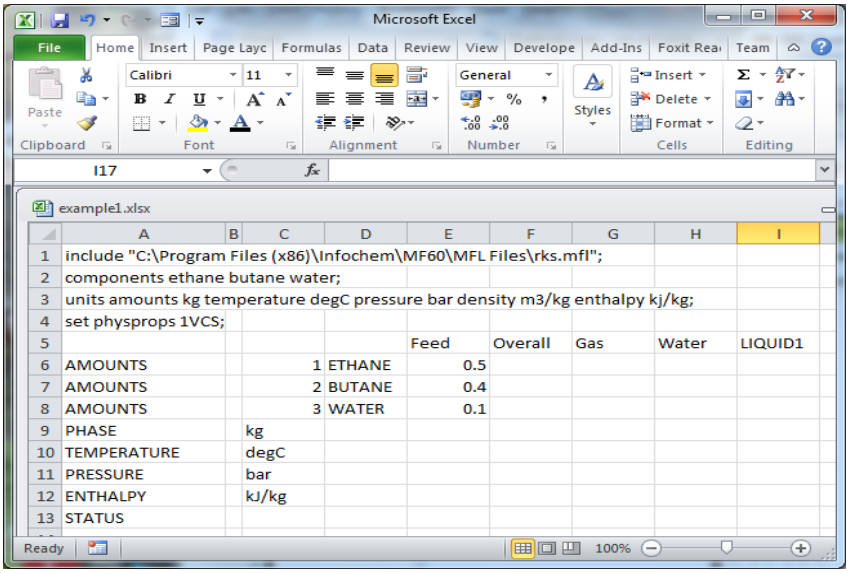
Cell reference	A	C	E	F	G	H	I
5			Feed	Overall	Gas	Water	LIQUID1
6	AMOUNTS	1					
7	AMOUNTS	2					
8	AMOUNTS	3					
9	PHASE						
10	TEMPERATURE						
11	PRESSURE						
12	ENTHALPY						
13	STATUS						

The values in row 5 and column A are used by the Multiflash functions as input arguments to specify the output quantity required, for example, the enthalpy of the gas phase.

3. Enter the following Multiflash functions into your worksheet:

Cell	Function	Description
D6	=MF_COMP ($C6,$A$1:$A$4)	This function returns the name of the components specified in the initialisation script. The first argument, $C6, is the number of the component. The second argument, $A$1:$A$3, is the range of cells containing the initialisation script. **Note:** An optional third argument is used for specifying a stream type. If the name of a stream type is given, the returned components belong to the stream type specified.
C9	=MF_OU ($A9,$A$1:$A$4)	This function returns the output units used for the specified quantity of your calculations. The first argument, $A9, is the quantity whose units you want to know. The second argument, $A$1:$A$3, is the range of cells containing the initialisation script.

4. Copy cell D6 and paste it into cells D7 and D8.
5. Copy cell C9 and paste it into cells C10, C11 and C12.
6. Enter the feed composition in cells E6, E7 and E8 by entering in 0.5, 0.4 and 0.1 respectively.



Result: You now have a column with the names of the three components and a column with the names of the units being used for your output.

7. Enter the temperature and pressure for your calculation by typing 30 and 20 into cells E10 and E11, respectively.
8. Enter the following Multiflash function in cell F6:

   =MF_PTF(F$5,$A6,$C6,$E$11,$E$10,$E$6:$E$8,$A$1:$A$4)

   This is the Multiflash function to perform a flash calculation at a specified temperature and pressure. Some cells are given absolute addresses for row, column, or both, so that the function can be copied to other cells in your table. For this example, the cell ranges specify the following arguments:

   F$5 is the required phase (overall value)
   $A6 is the required property (amount)

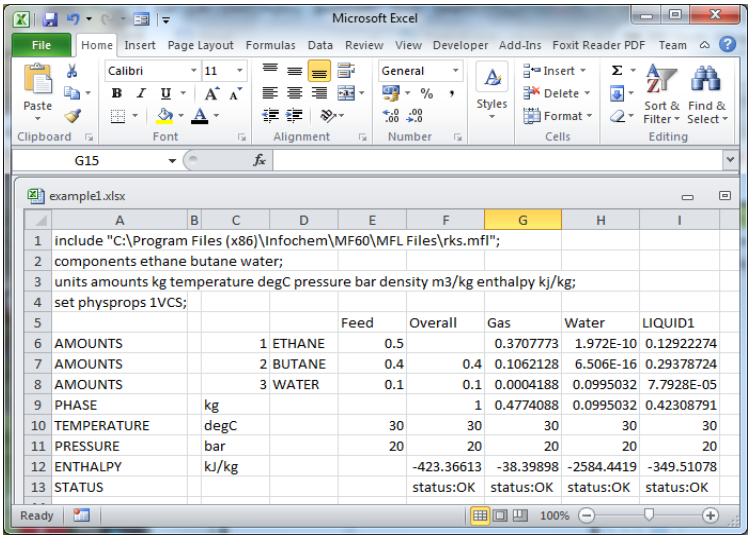
$C6 is the required component number (2)
$E$11 is the pressure at which the flash calculation is performed (20)
$E$10 is the temperature at which the flash calculation is performed (30)
$E$6:$E$8 is the mixture composition
$A$1:$A$4 is the initialisation script

Tip: You can enter the function using the standard Excel function wizard, which enables you to specify each argument in turn. See *Defining Multiflash functions*.

9. Copy cell F6 into the cell range F6:I13.

   Result: The values for all the phases and properties requested appear in the spreadsheet. The cells on the STATUS row display OK, to indicate that the flash function has worked correctly.

   Note: If the status is not OK, refer to *Dealing with Multiflash errors*.



10. To specify a table for your isenthalpic calculation data, enter the values shown in the table into your worksheet.

Cell reference	L	M	N	O
5	Overall	Gas	Water	LIQUID1

11. Enter the value 1.0 into cell K11.

    This is the pressure to use for the isenthalpic calculation, and simulates the effect of reducing the pressure of this fluid using a choke.

12. Enter the following Multiflash function in cell L6:

    =MF_PHF(L$5,$A6,$C6,$K$11,$F$12,$E$6:$E$8,$A$1:$A$4)

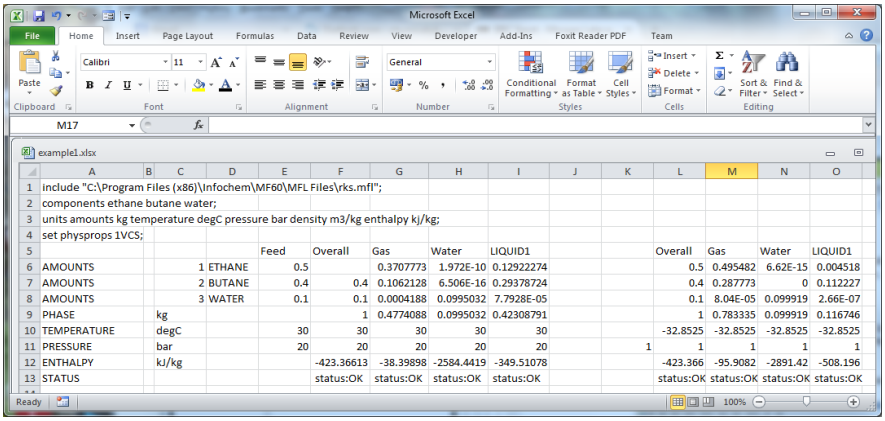
    This is the Multiflash function to perform a flash calculation at a specified temperature and pressure.

    Tip: You can enter the function using the standard Excel function wizard, which enables you to specify each argument in turn. See *Defining Multiflash functions*.

13. Copy cell L6 into the cell range L6:O13.

    Result: The values for all the phases and properties requested appear in the spreadsheet. The cells on the STATUS row display OK, to indicate that the flash function has worked correctly.

    Note: If the status is not OK, refer to *Dealing with Multiflash errors*.



You can vary the conditions by changing the value in any of the cells E6, E7, E8, E10, E11 or K11. Whenever you change these, Excel automatically calls Multiflash to update all the cells, provided automatic calculation is enabled. If automatic calculation is not enabled, you can recalculate the cells by pressing the F9 key.

Tip: You can request other physical properties by putting the appropriate name of the physical property in column 1 of a new row, and then copying the cells F6:O6 to the columns F to O of the new row. You can also specify other flashes by setting up further tables with different flash functions, as you have done for the PH flash.

Related Topics

- » Initialization scripts
- » Defining Multiflash functions
- » Dealing with Multiflash errors

Initialization scripts

An initialization script is a set of instructions comprising a sequence of Multiflash commands that specify all the information required for the successful use of the Multiflash functions within Excel. All Multiflash functions in Excel must reference an initialization script to work properly, because this script defines the problem Multiflash is to solve. For a complete description of all Multiflash commands, see the *Multiflash Command Reference Manual*.

The initialization script can be any of the following:

- A complete set of commands written directly into the worksheet
- One or more .mfl files to define some aspects of the problem, together with other commands written in the worksheet
- A .mfl file that contains a complete definition of the problem.

The way that you create an initialization script depends on your familiarity with Multiflash commands and the nature of the problem. For simple problems, writing the script directly into the worksheet has the following advantages:

- Each section (the model, the components, the units) can be defined in a single cell.
- The script can be easily read and modified.

See *Creating an initialisation script directly in Excel* for further details. For complex problems, using the Multiflash GUI is the easiest way to create an initialization script, see *Creating an initialisation script using the Multiflash GUI*.

There are a few simple rules for writing an initialization script:

- The script must be written in successive cells in a single column.
- Multiflash commands can be written over several consecutive cells, for clarity, or when the command exceeds the limit of 255 characters in one cell.
- Each Multiflash command must end with a semicolon (;).

Related Topics

» Creating an initialization script directly in Excel
» Creating an initialization script using the Multiflash GUI

Creating an initialization script directly in Excel

For simple problems, writing the initialization script directly into the worksheet has the following advantages:

- Each section (the model, the components, the units) can be defined in a single cell.
- The script can be easily read and modified.

To create an initialization script:

1. Do one or more of the following to add lines to your script:
   - Type Multiflash commands directly into the worksheet cells
   - Specify a problem using the Multiflash GUI, display the definition (**Tools > Commands > Current Fluid Definition**, and then copy the instructions to your worksheet.
   - include a .mfl file, using the following command:

     include “c:\MF\example.mfl”;

     Requirement: You must use the full path and file name of the .mfl file.

     Tip: This is the best method to include a set of fixed instructions that you want to use in multiple worksheets. You can include more than one file, for example, a file to specify a model setup, and a file to specify a fluid.
   - Use the Excel **Data > From Text** function to import the contents of a .mfl file, using the delimited option.

     Tip: This method is the easiest way to import problems that you have been working on, and enables you to modify individual commands after they are imported.
2. Save the worksheet.
3. When you reference the script in Multiflash functions, you must specify the cell range as an absolute range, for example, $A$1:$A$5.

Related Topics

» Initialization scripts
» Creating an initialization script using the Multiflash GUI

Creating an initialization script using the Multiflash GUI

For complex problems, using the Multiflash GUI is the easiest way to create an initialization script. You can also use an existing initialization script as a start point for a new problem.

To create an initialization script using the Multiflash GUI:

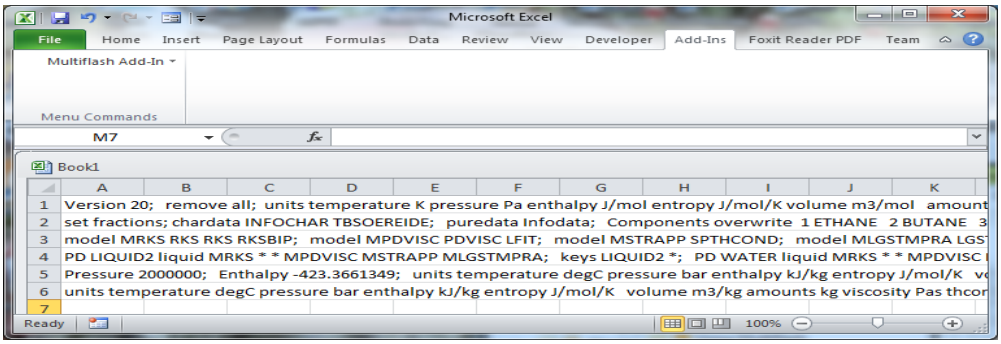
1. In the Excel interface, on the **Add-Ins** ribbon, choose **Multiflash Add-In > Call GUI**, and then click OK on the message.
2. In the Multiflash Add-In dialog box, do one of the following, and then click **OK**:
   - To start a new problem, enter the reference of a blank cell.
   - To use an existing script as a start point, enter the cell range of the script.

   Result: The Multiflash GUI starts. If you specified an existing script, the model, fluid and any other data included in the script are also defined in the Multiflash GUI.
3. In the Multiflash GUI, specify your fluid and model in the usual way, either from scratch or by building on the data input from the initialization script.

   Tip: For new problems, you can load an existing .mfl file to use as a start point.
4. When you are satisfied with your problem setup, choose **File > Quit**.
5. In the Multiflash Add-In dialog box, enter the cell range where you want to write your new initialization script, and then click **OK**.

   Requirement: You must specify a minimum number of rows in your range, as indicated in the message.

   Result: The script is written to the worksheet. The instructions are concatenated to minimise the number of rows used. The script has the same information as if the problem had been saved as a .mfl file using Multiflash interactively.



You can now reference the initialization script in your calculations.

Related Topics

- Initialization scripts
- Creating an initialization script directly in Excel

Updating worksheets

If any file included in the initialization script is altered, you must update the worksheet to ensure that the commands in the modified file are included in your calculations.

To update the worksheet:

1. In the Excel interface, on the **Add-Ins** ribbon, choose **Multiflash Add-In > Load**, and then specify the cell range of the initialization script.

   Result: Multiflash is reinitialized.

2. Click in one of the cells containing the initialization script, place the cursor in the formula bar, and then press enter.

   Result: Every Multiflash function in the worksheet is recalculated, using the new initialization script commands.

Related Topics

» Creating an initialization script directly in Excel
» Creating an initialization script using the Multiflash GUI

Entering Multiflash functions

You can enter Multiflash functions into the spreadsheet cells in one of these ways:

- Type them directly
- Copy them from other cells
- Use the function wizard.

You can type a function directly by referring its description, and assigning the appropriate cell addresses as the input arguments. You can subsequently copy the function to other cells, for example, to calculate the enthalpy of each phase of your fluid.

Note: If you intend to copy any functions to other cells, you must specify the cell addresses appropriately, using absolute values for the rows and columns when required. For example, the initialization script is always in the same cell range, so you should use the cell range representation $A$1:$A$5, so that the function always takes the script from the cell range A1:A5, wherever the function appears in the worksheet.

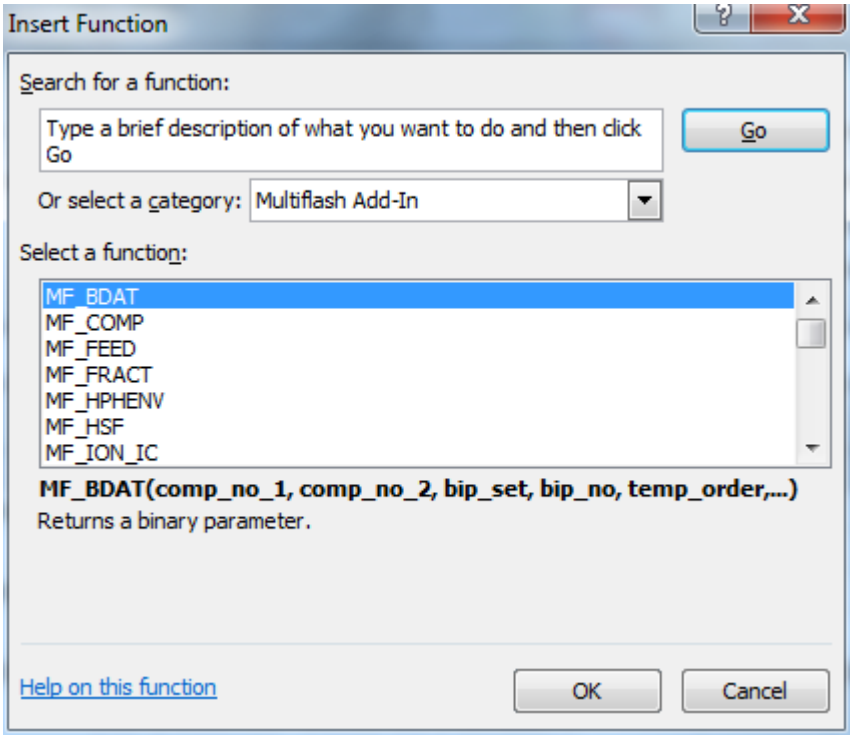
See *Getting started* for an example of specifying absolute cell addresses within a function.

Using the function wizard

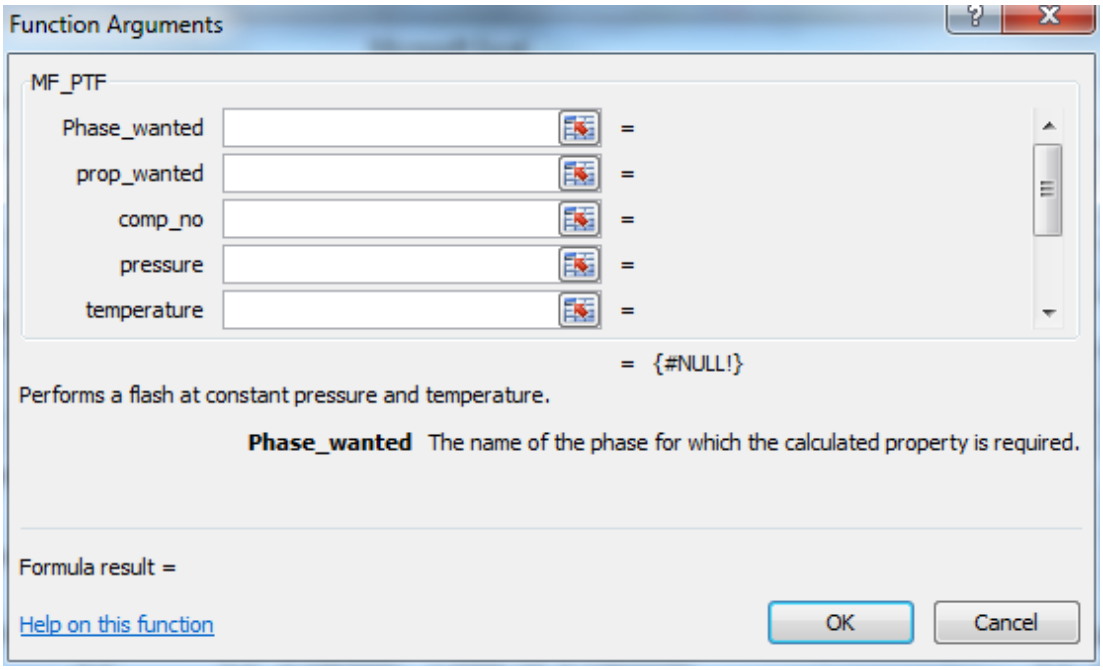
The function wizard prompts you for each argument in turn, so you do not need to remember all the arguments for a particular function.

To use the function wizard:

1. On the Excel interface, click in the cell where you want to add a Multiflash function.
2. On the **Formulas** ribbon, click **Insert Function**.



3. Select the Multiflash Add-In category.
4. From the function list, select the function that you want to add, and then click **OK**.



5. For each argument, enter the relevant cell or range address, using one of these methods:
   - Type the address directly into the box
   - Click in the box for an argument, and then select the cell or range of cells in the work-sheet.

   Tip: Ensure that you provide a cell or range for each argument by scrolling down the argument list.

6. If required, you must add absolute referencing to your addresses.

   You must do this if you intend to copy the function to other cells.

7. Click **OK**.

Related Topics

» Getting started
» Cell references

Cell references

Cell references are used to identify cells in a worksheet. There are three types of reference: relative references; absolute references; and mixed references.

Relative cell references

A reference such as A1 tells Excel how to find another cell, starting from the cell containing the formula. When any formula or function that uses the reference A1 is copied, the relative reference changes by the same number of rows and columns that the cell contents are moved by. For example, if a function in cell D6 is copied to cell F7, a relative reference A1 in cell D6 becomes C2 in cell F7, and the value in cell C2 is used in the function.

Absolute cell references

An absolute reference is designated by adding a dollar sign ($) before the column letter and row number. A reference such as $A$1 tells Excel how to find a cell based on the exact location of that cell in the worksheet. When any formula or function that uses the reference $A$1 is copied, the reference remains unchanged and the value used is still that in cell A1.

Tip: In Multiflash functions, use absolute references for the initialization script, and values that are constant within a calculation, for example, a fixed temperature or pressure.

Mixed cell references

A mixed reference is designated by adding a $ before either the column letter or row number. A reference such as $A1 tells Excel how to find another cell by combining a reference of an exact column or row with a relative row or column. When any formula or function that uses a mixed reference is copied, the absolute reference remains unchanged and the relative reference changes by the same number of rows or columns that the cell contents are moved by. For example, if a function in cell D6 is copied to cell F7, a mixed reference $A1 in cell D6 becomes $A2 in cell F7, and the value in cell A2 is used in the function.

Tip: In Multiflash functions, used mixed references for component names, properties and units.

Setting up flash functions as array functions

All the flash functions can be made to return results as an array of one or two dimensions by performing the following steps:

1. Specify input arguments as arrays in one of these ways:
    - Specify one argument as either a one-dimensional or two-dimensional array. When you do so, the function returns an array of values with the same number of rows and columns as the input array.
    - Specify two arguments for as one-dimensional arrays. When you do so, one must be a horizontal array and the other must be a vertical array. The function returns an array of values with the same number of rows as the vertical input array and the same number of columns as the horizontal input array.
2. Define the function as an array function in Excel.

    Note: If you define a flash function as an array that is too large, the extra cells are filled with an Excel error value; if you define a flash function as an array that is too small, the output is truncated to fill the defined cells only.

Only the following input arguments can be specified as arrays:

- phase_wanted
- prop_wanted and comp_no
- pressure
- temperature
- enthalpy
- entropy
- volume
- internalenergy
- fraction

Note: The argument comp_no does not count as a separate input argument when determining whether the flash function is an array function; comp_no must have the same number of rows and columns as prop_wanted. If comp_no is too large, the excess entries are ignored; if comp_no is too small, the missing entries are taken to be zeros.

Defining start values for array flash functions

Some of the flash functions can optionally be given a starting value for pressure, temperature, or both.

If the flash function in question is defined as an array with one array argument, any starting values are also defined as an array, with the same number of rows and columns as the array argument.

If the flash function in question is defined as an array with two array arguments, any starting values are also defined as an array, with the same number of rows as the vertical input array and the same number of columns as the horizontal input array.

If you define a starting value array that is too large, the extra cells are ignored; if you define a starting value array that is too small, the missing values are set to the last value defined in the row or column in question.

Related Topics

» Defining a flash function as an array function in Excel
» Setting up the phase envelope functions

Defining a flash function as an array function in Excel

You can define a flash function as an array whenever the results from a flash calculation are required in a rectangular block of cells on the spreadsheet.

Restriction: You cannot modify a cell within the array function; you have to delete the whole array and start again.

To specify a flash array function:

1. Enter the required function in a single cell, ensuring that you specify the required array arguments, and array starting values, if appropriate.
2. Click and hold the cell where you have entered the function, then drag the pointer to include the block of cells where you want the results to appear.
3. Click in the formula bar, then press Ctrl-Shift-Enter.

   Result: The results appear in the highlighted block of cells.

Related Topics

» Setting up flash functions as array functions

Setting up the phase envelope functions

The phase envelope functions are a special case of array functions. They always return results as an array of three columns and as many rows as given by the value of the argument 'no_points'.

To specify a phase envelope function calculation:

1. Enter the required function in a single cell.
2. Click and hold the cell where you have entered the function, then drag the pointer to include the block of cells where you want the results to appear.
3. Click in the formula bar, then press Ctrl-Shift-Enter.

   Result: The results appear in the highlighted block of cells.

The results are displayed in three columns, with each row containing a separate point on the phase envelope:

- The first column contains the calculated temperatures in the currently set output units.
- The second column contains the calculated pressures in the currently set output units.
- The third column contains the status code for the calculation in the top row and identifiers for any special points in subsequent rows. The identifier C marks a critical point and identifier D marks a discontinuity, that is, a point where the gradient of the line changes discontinuously.

If you define an array that is too large, the phase envelope will fill the extra cells with Excel error values; if you define an array that is too small, the output is truncated to fill the defined cells only.

Related Topics

» Setting up flash functions as array functions

Multiflash Excel functions

The Multiflash functions are available for use in Excel worksheets. To use any function, you must have a valid initialization script that defines the problem that Multiflash is required to solve.

Optional arguments are shown in italics in the function description. You need not enter a value for them unless they are required for your problem.

Some other arguments can be omitted under certain circumstances, which are described for each function. For example, when carrying out a PH flash, you do not need a reference to the component number if you just want to calculate the temperature. However, you must enter a reference to the component number if the required property directly relates to a component, for example, amounts.

Input and output units for all the properties are set in the Multiflash initialization script.

Related Topics

» Initialization scripts

General functions

Multiflash version number (MF_VERS)

Returns the version number of Multiflash being accessed by Excel.

=MF_VERS(init_script)

Arguments

Argument	Type	Description
init_script	String array (255)	The initialization script that defines the Multiflash problem.

BIP parameter (MF_BDAT)

Returns the binary parameter for a pair of given components.

=MF_BDAT(comp_no_1, comp_no_2, bip_set, bip_no, temp_order, init_script)

Arguments

Argument name	Type (size)	Description
Comp_no_1	Integer	Number of the first component.
Comp_no_2	Integer	Number of the second component.
BIP_set	String	The BIP set name.
Bip_no	Integer	The specific BIP that you want: ■ 1 = symmetric or asymmetric BIP A_{ij} ■ 2 = asymmetric BIP A_{ji} ■ 3 = alpha

Argument name	Type (size)	Description
Temp_order	Integer	The temperature order of the coefficient: ■ 0 = constant ■ 1 = T term ■ 2 = T^2 term
Init_script	String (255)	The initialization script that defines the Multiflash problem.

Sum of squares of N cells (MF_SUMSQN)

Returns the value of the sum of squares for the specified number of cells in a range.

=MF_SUMSQN(Range_To_Sum, Numbers_To_Sum, init_script)

Arguments

Argument	Type	Description
Range_ To_Sum	String (255)	The range of cells used.
Numbers_ To_Sum	Integer	The number of cells within the range used in the summation of squares function. The cells used always start from the upper or leftmost cell of the specified range.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Bubble and dew point functions

Bubble point at fixed pressure (MF_PBUBF)

Performs a bubble point calculation at constant pressure. Optional arguments are shown in italics.

=MF_PBUBF(phase_wanted, prop_wanted, comp_no or phase_wanted, basis, pressure, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Bubble point at fixed temperature (MF_TBUBF)

Performs a bubble point calculation at constant pressure. Optional arguments are shown in italics.

=MF_TBUBF(phase_wanted, prop_wanted, comp_no or phase_wanted, basis, temperature, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Dew point at fixed pressure (MF_PDEWF)

Performs a dew point calculation at constant pressure. Optional arguments are shown in italics.

=MF_PDEWF(phase_wanted, prop_wanted, comp_no or phase_wanted, basis, pressure, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Dew point at fixed temperature (MF_TDEWF)

Performs a dew point calculation at constant temperature. Optional arguments are shown in italics.

=MF_TDEWF(phase_wanted, prop_wanted, comp_no or phase_wanted, basis, temperature, amounts, init_script, *stream_type*, *starting_press*)

For a description of the arguments, see *Flash function arguments* on page 35.

Component functions

Component name (MF_COMP)

Returns the name corresponding to a given component number, with or without a stream type.

=MF_COMP(comp_no, init_script, *stream_type*)

Arguments

Argument name	Type (size)	Description
comp_no	Integer	The number of the component in your fluid.
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_ type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Component feed composition (MF_FEED)

Returns the amount that is defined in the initialisation script for a given component number, with or without a stream type. The result is given in the current input units for amounts.

=MF_FEED(comp_no, init_script, *stream_type*)

Arguments

Argument name	Type (size)	Description
comp_no	Integer	The number of the component in your fluid.
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_ type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Component properties (MF_PDAT)

For temperature-independent physical properties, returns the value of the property.

For temperature-dependent physical properties, returns information about the equation used to calculate the physical property.

=MF_PDAT(comp_no, comp_prop, offset, init_script)

Arguments

Argument name	Type (size)	Description
comp_no	Integer	The number of the component in your fluid.

<table>
<tr><th>Argument name</th><th>Type (size)</th><th>Description</th></tr>
<tr><td>comp_ prop</td><td>String (40)</td><td>The required physical property. The property units are given in parentheses where applicable; units for other properties are set by the initialization script.
Temperature-independent properties:
<table>
<tr><td>Molecularweight / MW</td><td>relative molar mass (molecular weight in g/mol)</td></tr>
<tr><td>Tcrit</td><td>critical temperature</td></tr>
<tr><td>Pcrit</td><td>critical pressure</td></tr>
<tr><td>Vcrit</td><td>critical volume</td></tr>
<tr><td>Acentricfactor</td><td>Pitzer acentric factor defined as</td></tr>
<tr><td>Tboil</td><td>normal boiling point</td></tr>
<tr><td>Hformation</td><td>standard enthalpy change on formation in the perfect gas state at 298.15K</td></tr>
<tr><td>Sformation</td><td>Entropy of formation</td></tr>
<tr><td>Sstandard</td><td>standard entropy in the perfect gas state at 298.15K and 1bar</td></tr>
<tr><td>Tmelt</td><td>normal melting point</td></tr>
<tr><td>Hmelt</td><td>enthalpy change on fusion at the melting point</td></tr>
<tr><td>Smelt</td><td>entropy change on fusion at the melting point</td></tr>
<tr><td>Cpmelt</td><td>solid/liquid difference in Cp at the melting point</td></tr>
<tr><td>Vmelt</td><td>solid/liquid difference in molar volume at the melting point</td></tr>
<tr><td>Runiquac</td><td>UNIQUAC r parameter</td></tr>
<tr><td>Quniquac</td><td>UNIQUAC q parameter</td></tr>
<tr><td>Thlwater</td><td>characteristic temperature for Henry’s Law correlation for component in water</td></tr>
<tr><td>Vhlwater</td><td>characteristic volume for Henry’s Law correlation for component in water</td></tr>
<tr><td>Dipolemoment</td><td>dipole moment (debye)</td></tr>
<tr><td>Parachor</td><td>parachor ($(dyne/cm)^{1/4}$ cm^3/mol)</td></tr>
<tr><td>Radgyr</td><td>radius of gyration (m)</td></tr>
<tr><td>Hocass</td><td>Hayden O’Connell self association parameter</td></tr>
<tr><td>Gformation</td><td>Gibbs energy of formation in the perfect gas state at 298.15K and 1 bar</td></tr>
</table></td></tr>
</table>

Argument name	Type (size)	Description	
		Ttriple	triple point temperature
		Ptriple	triple point pressure
		Hcombustion	enthalpy of combustion
		v25	molar volume at 298.15K for use with solubility parameters
		Solupar	liquid phase solubility parameter
		Solidsolupar	solid solution phase solubility parameter
		Zcrit	critical compressibility factor
		Refractindex	refractive index
		Tflash	flash temperature
		Tauto	autoignition temperature
		Flamlower	lower flammability limit (volume % in air)
		Flamupper	upper flammability limit (volume % in air)
		SG	Specific gravity
		Expansivity	Thermal expansivity of liquid at 1 atm and 60 F(K^{-1})
		OMARKS	Scaling factor to give conventional value of Ω_a specific to RKS model.
		OMBRKS	Scaling factor to give conventional value of Ω_b specific to RKS model.
		OMAPR	Scaling factor to give conventional value of Ω_a specific to PR model.
		OMBPR	Scaling factor to give conventional value of Ω_b specific to PR model.
		CNUMBER	Carbon number for petroleum fractions.
		REFVISCOSITY	Liquid viscosity at reference temperature and 1 atm (Pa s).
		REFVISST	reference viscosity for liquid at the boiling point for SUPERTRAPP viscosity model
		REFVISPD	reference viscosity for liquid at the boiling point for Pedersen viscosity model
		REFVISTW	reference viscosity for liquid at the boiling point for Two model

Argument name	Type (size)	Description	
		REFVISLB	reference viscosity for liquid at the boiling point for LBC model
		LJEVISC	Lennard-Jones ε/k parameter for viscosity (K)
		LJBVISC	Lennard-Jones σ parameter (m)
		EOSC	Corresponding states reference equation code
		TYPE	Type of component
		HDATUM	Options for calculating enthalpy 1 for compound 2 for elements 3 for standard enthalpy (298.15K)
		SDATUM	Options for calculating entropy 1 for compound 2 for elements 3 for standard entropy (298.15K, 1 bar)
		MCRKS1	First Mathias Copeman parameter with EoS RKSA
		MCRKS2	Second Mathias Copeman parameter with EoS RKSA
		MCRKS3	Third Mathias Copeman parameter with EoS RKSA
		MCPR1	First Mathias Copeman parameter with EoS PRA
		MCPR2	Second Mathias Copeman parameter with EoS PRA
		MCPR3	Third Mathias Copeman parameter with EoS PRA
		HYDOC	Hydrate cavity occupation codes
		HYD1	Hydrate parameter 1
		HYD2	Hydrate parameter 2
		HYD3	Hydrate parameter 3
		ASSBETA	CPA β parameter
		ASSEPSILON	CPA ε_{AB} parameter (J mol^{-1})
		ASSGAMMA	CPA γ_{AB} parameter
		ASSDELTA	CPA δ_{AB} parameter (K^{-1})
		ASSFF	Number of association sites (CPA)
		ASSAC60	CPA a_c parameter (J $m^3 mol^{-2}$)
		ASSBC	CPA b parameter (m^3 mol^{-1})

Argument name	Type (size)	Description
		ASSKAPPA — CPA κ parameter SAFTKAPPA — PC-SAFT κ parameter SAFTEPSILON — PC-SAFT ε parameter SAFTGAMMA — PC-SAFT γ parameter SAFTFF — Number of association sites (PC-SAFT) SAFTEK — PC-SAFT ε/k parameter SAFTSIGMA — PC-SAFT σ parameter SAFTLAMBDA — PC-SAFT λ parameter SAFTM — PC-SAFT m parameter SAFTQ — Charge of per segment SAFTFRQ — Fraction of charge per segment VSRKS1 — The first volume shift parameter with EoS RKSA VSRKS2 — The second volume shift parameter with EoS RKSA VSRKS3 — The third volume shift parameter with EoS RKSA VSPR1 — The first volume shift parameter with EoS PRA VSPR2 — The second volume shift parameter with EoS PRA VSPR3 — The third volume shift parameter with EoS PRA CPIDLIQ — Set to 1 to use the pure liquid enthalpy as reference for excess energy models, set to 2 if you intend to use the gas activity coefficient model. VCLBC — Critical volume for LBC viscosity model **Temperature-dependent properties:** Cpideal — Ideal gas Cp correlation (J mol^{-1} K^{-1}) Psat — Saturated vapour pressure (Pa) Hvap — Enthalpy of vaporisation correlation (J mol^{-1}) Ldens — Saturated liquid density correlation (mol m^{-3}) Lvisc — Liquid viscosity correlation (Pa s) Vvisc — Vapour viscosity correlation (Pa s)

<table>
<tr><th>Argument name</th><th>Type (size)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>Lthcond</td><td>Liquid thermal conductivity correlation (W m^{-1} K^{-1})</td></tr>
<tr><td>Vthcond</td><td>Vapour thermal conductivity correlation (W m^{-1} K^{-1})</td></tr>
<tr><td>Stension</td><td>Surface tension correlation (N m^{-1})</td></tr>
<tr><td>Cpliquid</td><td>Liquid Cp correlation (J mol^{-1} K^{-1})</td></tr>
<tr><td>Sdens</td><td>Solid density correlation (mol m^{-3})</td></tr>
<tr><td>Cpsolid</td><td>Solid Cp correlation (J mol^{-1} K^{-1})</td></tr>
<tr><td>Virialcoeff</td><td>Second virial coefficient correlation (m^3 mol^{-1})</td></tr>
</table>
</td></tr>
<tr><td>offset</td><td>Integer</td><td>The offset used for viewing the equation number, coefficients and temperature limits of a temperature-dependent property correlation. For temperature-independent properties, this option can be omitted.
Note: For the ideal gas correlation (Cpideal) only, the offset values 1 and 2 return the enthalpy and entropy constants of integration, such that gas enthalpy and entropy are zero at 298.15 K and 1 bar. The coefficients and temperature limits are available from offset value 3 onwards.
For example, if you are using the saturated vapour pressure equation 1 from the Multiflash Models and Physical Properties User Guide, the offset value to return the required equation detail is shown in the following table. For equations with a different number of coefficients, use offset values appropriate for the number of coefficients in the equation; T_{min} and T_{max} are always after the coefficients.
<table>
<tr><th>Offset value</th><th>Returns</th></tr>
<tr><td>0</td><td>The equation number, as specified in the Multiflash Models and Physical Properties User Guide.</td></tr>
<tr><td>1</td><td>Coefficient a_1</td></tr>
<tr><td>2</td><td>Coefficient a_2</td></tr>
<tr><td>3</td><td>Coefficient a_3</td></tr>
<tr><td>4</td><td>T_{min}</td></tr>
<tr><td>5</td><td>T_{max}</td></tr>
</table>
</td></tr>
<tr><td>init_script</td><td>String array (255)</td><td>The initialization script that defines the Multiflash problem.</td></tr>
</table>

Temperature-dependent component properties (MF_PDAT_T)

Returns the value of the specified temperature-dependent property of a pure component or petroleum fraction.

=MF_PDAT_T(comp_no, comp_prop, temperature, init_script)

Arguments

Argument name	Type (size)	Description
comp_no	Integer	The number of the component in your fluid.
comp_prop	String (40)	The required temperature-dependent physical property. Cpideal — Ideal gas Cp correlation (J mol^{-1} K^{-1}) Psat — Saturated vapour pressure (Pa) Hvap — Enthalpy of vaporisation correlation (J mol^{-1}) Ldens — Saturated liquid density correlation (mol m^{-3}) Lvisc — Liquid viscosity correlation (Pa s) Vvisc — Vapour viscosity correlation (Pa s) Lthcond — Liquid thermal conductivity correlation (W m^{-1} K^{-1}) Vthcond — Vapour thermal conductivity correlation (W m^{-1} K^{-1}) Stension — Surface tension correlation (N m^{-1}) Cpliquid — Liquid Cp correlation (J mol^{-1} K^{-1}) Sdens — Solid density correlation (mol m^{-3}) Cpsolid — Solid Cp correlation (J mol^{-1} K^{-1}) Virialcoeff — Second virial coefficient correlation (m^3 mol^{-1})
temperature	Real (double)	The temperature at which you want the property evaluated, specified in the current input units.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Fixed phase fraction functions

Fixed phase fraction flash at fixed pressure (MF_PFRACF)

Performs a fixed phase fraction flash at constant pressure. Optional arguments are shown in italics.

=MF_PFRACF(phase_wanted, prop_wanted, comp_no or phase_wanted, fixed_phase, basis, pressure, fraction, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Fixed phase fraction flash at fixed temperature (MF_TFRACF)

Performs a fixed phase fraction flash at constant temperature. Optional arguments are shown in italics.

=MF_PFRACF(phase_wanted, prop_wanted, comp_no or phase_wanted, fixed_phase, basis, pressure, fraction, amounts, init_script, *stream_type*, *starting_press*)

For a description of the arguments, see *Flash function arguments* on page 35.

Fixed wax phase fraction flash at fixed pressure (MF_PWAT)

Performs a fixed wax phase fraction flash at constant pressure. Optional arguments are shown in italics. The basis option is related to the first argument for the phase_wanted. If phase_wanted is specified as “overall”, it means that the fraction for the wax phase required is related to the total fluid. If phase_wanted is specified as the name of the liquid phase, the fixed phase fraction of wax is related to the liquid+wax phase.

=MF_PWAT(phase_wanted, prop_wanted, comp_no/phase_wanted, basis, pressure, fraction, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Tolerance calculation for fixed phase fraction (MF_FRACT)

Performs a tolerance calculation for a fixed phase fraction at constant pressure and temperature. The tolerance calculation takes an initial fluid and adds a second fluid until the fixed phase fraction is satisfied at the defined pressure and temperature. Optional arguments are shown in italics.

=MF_FRACT(phase_wanted, prop_wanted, comp_no, fixed_phase, basis, pressure, temperature, fraction, amounts(1) , amounts(2) , init_script, *stream_type*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash functions

Flash at fixed pressure and temperature (MF_PTF)

Performs a flash at constant pressure and temperature. Optional arguments are shown in italics.

=MF_PTF(phase_wanted, prop_wanted, comp_no, pressure, temperature, amounts, init_script, *stream_type*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed pressure and enthalpy (MF_PHF)

Performs a flash at fixed pressure and enthalpy. Optional arguments are shown in italics.

=MF_PHF(phase_wanted, prop_wanted, comp_no, pressure, enthalpy, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed temperature and enthalpy (MF_THF)

Performs a flash at fixed temperature and enthalpy. Optional arguments are shown in italics.

=MF_THF(phase_wanted, prop_wanted, comp_no, temperature, enthalpy, amounts, init_script, *stream_type*, *starting_press*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed pressure and entropy (MF_PSF)

Performs a flash at fixed pressure and entropy. Optional arguments are shown in italics.

=MF_PTF(phase_wanted, prop_wanted, comp_no, pressure, entropy, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed temperature and entropy (MF_TSF)

Performs a flash at fixed temperature and entropy. Optional arguments are shown in italics.

=MF_TSF(phase_wanted, prop_wanted, comp_no, temperature, entropy, amounts, init_script, *stream_type*, *starting_press*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed pressure and volume or density (MF_PVF)

Performs a flash at fixed pressure and volume/density. Optional arguments are shown in italics.

=MF_PVF(phase_wanted, prop_wanted, comp_no, pressure, volume, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed temperature and volume or density (MF_TVF)

Performs a flash at fixed temperature and volume/density. Optional arguments are shown in italics.

=MF_TVF(phase_wanted, prop_wanted, comp_no, temperature, volume, amounts, init_script, *stream_type*, *starting_press*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed pressure and internal energy (MF_PUF)

Performs a flash at fixed pressure and internal energy. Optional arguments are shown in italics.

=MF_PUF(phase_wanted, prop_wanted, comp_no, pressure, internalenergy, amounts, init_script, *stream_type*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed temperature and internal energy (MF_TUF)

Performs a flash at fixed temperature and internal energy. Optional arguments are shown in italics.

=MF_TUF(phase_wanted, prop_wanted, comp_no, temperature, internalenergy, amounts, init_script, *stream_type*, *starting_press*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed internal energy and volume or density (MF_UVF)

Performs a flash at fixed internal energy and volume/density. Optional arguments are shown in italics.

=MF_UVF(phase_wanted, prop_wanted, comp_no, internalenergy,volume, amounts, init_script, *stream_type*, *starting_press*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed enthalpy and volume or density (MF_HVF)

Performs a flash at fixed enthalpy and volume/density. Optional arguments are shown in italics.

=MF_UVF(phase_wanted, prop_wanted, comp_no, enthalpy,volume, amounts, init_script, *stream_type*, *starting_press*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed enthalpy and entropy (MF_HSF)

Performs a flash at fixed enthalpy and entropy. Optional arguments are shown in italics.

=MF_HSF(phase_wanted, prop_wanted, comp_no, enthalpy, entropy, amounts, init_script, *stream_type*, *starting_press*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash at fixed entropy and volume or density (MF_SVF)

Performs a flash at fixed entropy and volume/density. Optional arguments are shown in italics.

=MF_SVF(phase_wanted, prop_wanted, comp_no, entropy, volume, amounts, init_script, *stream_type*, *starting_press*, *starting_temp*)

For a description of the arguments, see *Flash function arguments* on page 35.

Flash function arguments

The arguments shown in the table are used by the Multiflash functions.

Argument name	Type (size)	Description
phase_wanted	String (40)	The name of the phase for which the calculated property is required. The name must correspond to one of those used to define the phases in the Multiflash initialization script. Infochem normally uses the following convention, although you can define your own phase names:

GAS	gas/vapour phase
LIQUID1	first hydrocarbon liquid phase
LIQUID2	second hydrocarbon liquid phase
WATER	aqueous liquid phase
ICE	ice phase
HYDRATE1	hydrate structure I phase
HYDRATE2	hydrate structure II phase
HYDRATEH	hydrate structure H phase
OVERALL	the total value for all phases added together

<table>
<tr><th>Argument name</th><th>Type (size)</th><th>Description</th></tr>
<tr><td>prop_wanted</td><td>Integer</td><td>The required property of a stream. The available properties are:
<table>
<tr><td>ACTCO</td><td>activity coefficient of a component in a phase.</td></tr>
<tr><td>AMOUNTS</td><td>concentration of a component in the phase requested as total amount</td></tr>
<tr><td>AVMOLWT[3]</td><td>average molecular weight</td></tr>
<tr><td>COMPRESSIBILITY</td><td>isothermal compressibility</td></tr>
<tr><td>CP</td><td>heat capacity at constant pressure</td></tr>
<tr><td>CV</td><td>heat capacity at constant volume</td></tr>
<tr><td>ENTHALPY</td><td>enthalpy</td></tr>
<tr><td>ENTHALPY_ELEM</td><td>enthalpy relative to elements</td></tr>
<tr><td>ENTROPY</td><td>entropy</td></tr>
<tr><td>ENTROPY_ELEM</td><td>entropy relative to elements</td></tr>
<tr><td>EXPANSIVITY</td><td>isobaric expansivity</td></tr>
<tr><td>FRACTIONS[3]</td><td>concentration of a component in the phase requested as the mass or mole fraction</td></tr>
<tr><td>FUGCO</td><td>fugacity coefficient of a component in a phase.</td></tr>
<tr><td>GIBBS</td><td>Gibbs energy</td></tr>
<tr><td>GIBBS_ELEM</td><td>Gibbs energy relative to elements</td></tr>
<tr><td>INTERNALENERGY</td><td>internal energy</td></tr>
<tr><td>INTERNALENERGY_ELEM</td><td>internal energy relative to elements</td></tr>
<tr><td>JTCOEFF</td><td>the value of the Joule-Thompson coefficient.</td></tr>
<tr><td>PHASE</td><td>total amount of that phase</td></tr>
<tr><td>PRESSURE</td><td>pressure</td></tr>
<tr><td>SSOUND</td><td>speed of sound</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Argument name</th><th>Type (size)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>STATUS[3]</td><td>success/error status of the flash calculation[2]</td></tr>
<tr><td>SURTEN[1]</td><td>surface tension</td></tr>
<tr><td>TEMPERATURE</td><td>temperature</td></tr>
<tr><td>THCOND</td><td>thermal conductivity</td></tr>
<tr><td>VISCOSITY</td><td>Viscosity</td></tr>
<tr><td>VOLUME (or DENSITY)</td><td>volume/density</td></tr>
<tr><td>ZFACTOR</td><td>compressibility factor</td></tr>
</table>
</td></tr>
<tr><td>comp_no[1]</td><td>Integer</td><td>The number of the component in your fluid. For flash calculations where the prop_wanted value is not ‘AMOUNTS’ or ‘FRACTIONS’, comp_no is ignored and can be left blank in either the cell or calling argument list.</td></tr>
<tr><td>fixed_phase</td><td>String (40)</td><td>The phase with the fixed fraction. The phase must be identified by the name in the Multiflash initialization script. The Infochem name conventions can be used, as for phase_wanted, with the exception of ‘OVERALL’.</td></tr>
</table>

<table>
<tr><th>Argument name</th><th>Type (size)</th><th>Description</th></tr>
<tr><td>basis</td><td>String (40)</td><td>The type of calculation:
<table>
<tr><td>NORMAL</td><td>Default for mole fraction</td></tr>
<tr><td>LOWER_NORMAL</td><td>Lower normal</td></tr>
<tr><td>UPPER_NORMAL</td><td>Upper normal</td></tr>
<tr><td>LOWER_
RETROGRADE</td><td>Lower retrograde</td></tr>
<tr><td>UPPER_
RETROGRADE</td><td>Upper retrograde</td></tr>
<tr><td>UNSPECIFIED</td><td>Unspecified</td></tr>
<tr><td>NORMAL_
MASSFRACTION</td><td>Normal for mass fraction</td></tr>
<tr><td>LOWER_NORMAL_
MASSFRACTION</td><td>Lower normal for mass fraction</td></tr>
<tr><td>UPPER_NORMAL_
MASSFRACTION</td><td>Upper normal for mass fraction</td></tr>
<tr><td>LOWER_
RETROGRADE_
MASSFRACTION</td><td>Lower retrograde for mass fraction</td></tr>
<tr><td>UPPER_
RETROGRADE_
MASSFRACTION</td><td>Upper retrograde for mass fraction</td></tr>
<tr><td>UNSPECIFIED_
MASSFRACTION</td><td>Unspecified for mass fraction</td></tr>
<tr><td>NORMAL_
VOLUMEFRACTION</td><td>Normal for volume fraction</td></tr>
<tr><td>LOWER_NORMAL_
VOLUMEFRACTION</td><td>Lower normal for volume fraction</td></tr>
<tr><td>UPPER_NORMAL_
VOLUMEFRACTION</td><td>Upper normal for volume fraction</td></tr>
<tr><td>LOWER_
RETROGRADE_
VOLUMEFRACTION</td><td>Lower retrograde for volume fraction</td></tr>
<tr><td>UPPER_
RETROGRADE_
VOLUMEFRACTION</td><td>Upper retrograde for volume fraction</td></tr>
</table></td></tr>
</table>

Argument name	Type (size)	Description
		UNSPECIFIED_VOLUMEFRACTION: Unspecified for volume fraction NORMAL_NUCLEATION: Normal for nucleation LOWER_NORMAL_NUCLEATION: Lower normal for nucleation UPPER_NORMAL_NUCLEATION: Upper normal for nucleation LOWER_RETROGRADE_NUCLEATION: Lower retrograde for nucleation UPPER_RETROGRADE_NUCLEATION: Upper retrograde for nucleation UNSPECIFIED_NUCLEATION: Unspecified for nucleation
init_var	String (40)	The quantity to fix to define the starting point of the phase envelope trace. Can be set to either 'TEMPERATURE' or 'PRESSURE'.
direction	String (4)	The initial direction of the temperature or pressure along the phase envelope. Can be set to 'UP' or 'DOWN'.
pressure	Real (double)	The pressure, specified in the current input units.
temperature	Real (double)	The temperature, specified in the current input units.
enthalpy	Real (double)	The enthalpy, specified in the current input units.
entropy	Real (double)	The entropy, specified in the current input units.
volume	Real (double)	The volume or density, specified in the current input units.
internalenergy	Real (double)	The internal energy, specified in the current input units.
fraction	Real (double)	The phase fraction specified.
amounts	Real (double)	The array of quantities of all components in the fluid, specified in the current input units.
amounts(1)	Real (double)	The array of quantities of all components in the first fluid of the tolerance calculation, specified in the current input units.

Argument name	Type (size)	Description
amounts(2)	Real (double)	The array of quantities of all components in the second fluid of the tolerance calculation, specified in the current input units.
no_points	Integer	The maximum number of points that the phase envelope function is to return.
skip_no	Integer	The number of calculated phase envelope points that should be skipped between points that are returned by the function. This number is designed to reduce the number of points plotted if less detail is required. If 'skip_no' is set to zero, all calculated points are returned.
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.
starting_press	Real (double)	The initial value of the pressure, specified in the current input units, for a calculation that involves a pressure search.
starting_temp	Real (double)	The initial value of the temperature, specified in the current input units, for a calculation that involves a temperature search.

[1]If the requested property (prop_wanted) is SURTEN, the second phase name must be provided in the comp_no field, using one of the values specified for phase_wanted.

[2]If the calculation is successful, STATUS returns the value 'OK', otherwise it displays the Multiflash error numbers (with the limitation that Excel can only display up to 255 characters in any one cell). Positive numbers are Multiflash errors and negative numbers are Multiflash warnings. The meaning of each error code can be found by selecting **Multiflash Add-In > Error Codes** from the Excel **Add-Ins** ribbon.

[3]If units are requested for this dimensionless quantity, the cell specifying the unit is set blank.

Phase envelope functions

Phase envelope (MF_PHENV)

Traces a phase line or boundary for fixed phase fraction. Optional arguments are shown in italics.

This is an array function, see *Setting up the phase envelope function*.

=MF_PHENV(fixed_phase, basis, init_var, direction, pressure, temperature, fraction, amounts, no_points, skip_no, init_script, *stream_type*)

For a description of the arguments, see *Flash function arguments* on page 35.

Related Topics

» Setting up the phase envelope function

The column headings for phase envelope (MF_PHENV_COLS)

Returns the column headings for phase envelope routines. Optional arguments are shown in italics.

This is an array function, see *Setting up the phase envelope function*.

=MF_PHENV_COLS(init_script, *stream_type*)

Arguments

Argument name	Type (size)	Description
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Related Topics

» Setting up the phase envelope function

Fixed enthalpy line (MF_HPHENV)

Traces a line of fixed enthalpy. Optional arguments are shown in italics.

This is an array function, see *Setting up the phase envelope function*.

=MF_HPHENV(init_var, direction, pressure, temperature, enthalpy, amounts, no_points, skip_no, init_script, stream_type)

Arguments

Argument name	Type (size)	Description
init_var	String (40)	The quantity to fix to define the starting point of the phase envelope trace. Can be set to either 'TEMPERATURE' or 'PRESSURE'.
direction	String (4)	The initial direction of the temperature or pressure along the phase envelope. Can be set to 'UP' or 'DOWN'.
pressure	Real (double)	The pressure, specified in the current input units.
temperature	Real (double)	The temperature, specified in the current input units.
amounts	Real (double)	The array of quantities of all components in the fluid, specified in the current input units.
no_points	Integer	The maximum number of points that the phase envelope function is to return.
skip_no	Integer	The number of calculated phase envelope points that should be skipped between points that are returned by the function. This number is designed to reduce the number of points plotted if less detail is required. If 'skip_no' is set to zero, all calculated points are returned.

Argument name	Type (size)	Description
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Related Topics

» Setting up the phase envelope function

Fixed entropy line (MF_SPHENV)

Traces a line of fixed entropy. Optional arguments are shown in italics.

This is an array function, see *Setting up the phase envelope function*.

=MF_SPHENV(init_var, direction, pressure, temperature, entropy, amounts, no_points, skip_no, init_script, stream_type)

Arguments

Argument name	Type (size)	Description
init_var	String (40)	The quantity to fix to define the starting point of the phase envelope trace. Can be set to either 'TEMPERATURE' or 'PRESSURE'.
direction	String (4)	The initial direction of the temperature or pressure along the phase envelope. Can be set to 'UP' or 'DOWN'.
pressure	Real (double)	The pressure, specified in the current input units.
temperature	Real (double)	The temperature, specified in the current input units.
entropy	Real (double)	The entropy, specified in the current input units.
amounts	Real (double)	The array of quantities of all components in the fluid, specified in the current input units.
no_points	Integer	The maximum number of points that the phase envelope function is to return.
skip_no	Integer	The number of calculated phase envelope points that should be skipped between points that are returned by the function. This number is designed to reduce the number of points plotted if less detail is required. If 'skip_no' is set to zero, all calculated points are returned.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Argument name	Type (size)	Description
stream_type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Related Topics

» Setting up the phase envelope function

Fixed internal energy line (MF_UPHENV)

Traces a line of fixed internal energy. Optional arguments are shown in italics.

This is an array function, see *Setting up the phase envelope function*.

=MF_UPHENV(init_var, direction, pressure, temperature, internalenergy, amounts, no_points, skip_no, init_script, *stream_type*)

Arguments

Argument name	Type (size)	Description
init_var	String (40)	The quantity to fix to define the starting point of the phase envelope trace. Can be set to either 'TEMPERATURE' or 'PRESSURE'.
direction	String (4)	The initial direction of the temperature or pressure along the phase envelope. Can be set to 'UP' or 'DOWN'.
pressure	Real (double)	The pressure, specified in the current input units.
temperature	Real (double)	The temperature, specified in the current input units.
internalenergy	Real (double)	The internal energy, specified in the current input units.
amounts	Real (double)	The array of quantities of all components in the fluid, specified in the current input units.
no_points	Integer	The maximum number of points that the phase envelope function is to return.
skip_no	Integer	The number of calculated phase envelope points that should be skipped between points that are returned by the function. This number is designed to reduce the number of points plotted if less detail is required. If 'skip_no' is set to zero, all calculated points are returned.
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Related Topics

» Setting up the phase envelope function

Fixed volume line (MF_VPHENV)

Traces a line of fixed volume. Optional arguments are shown in italics.

This is an array function, see *Setting up the phase envelope function*.

=MF_VPHENV(init_var, direction, pressure, temperature, volume, amounts, no_points, skip_no, init_script, *stream_type*)

Arguments

Argument name	Type (size)	Description
init_var	String (40)	The quantity to fix to define the starting point of the phase envelope trace. Can be set to either 'TEMPERATURE' or 'PRESSURE'.
direction	String (4)	The initial direction of the temperature or pressure along the phase envelope. Can be set to 'UP' or 'DOWN'.
pressure	Real (double)	The pressure, specified in the current input units.
temperature	Real (double)	The temperature, specified in the current input units.
volume	Real (double)	The volume or density, specified in the current input units.
amounts	Real (double)	The array of quantities of all components in the fluid, specified in the current input units.
no_points	Integer	The maximum number of points that the phase envelope function is to return.
skip_no	Integer	The number of calculated phase envelope points that should be skipped between points that are returned by the function. This number is designed to reduce the number of points plotted if less detail is required. If 'skip_no' is set to zero, all calculated points are returned.
init_script	String array (255)	The initialization script that defines the Multiflash problem.
stream_type	String (40)	The Multiflash stream type. If only one stream is used in your problem, you can omit this argument.

Related Topics

» Setting up the phase envelope function

Salt calculator functions

For the original salt component model, one of the components in the INFODATA databank is called “salt component”. This is used to represent a salt pseudo component, based on a sodium chloride equivalence, for use in calculating freezing point depression or hydrate inhibition.

For the electrolyte model, which is based on defining ions and not a salt pseudo component, the ions Na+, K+, Ca++, Cl- and Br- can be defined from the INFODATA databank.

To determine the inhibition effect of salt, using either the salt component model or the electrolyte model, you must provide a description of the salt content of the aqueous stream; however, the data for the salt, brine or formation and production water ion analysis is often not available to you in the appropriate units. To help generate the data in the form required by Multiflash, you can use the salt calculator functions to calculate the equivalent amount of the “salt component” or the ions.

Before you can use the salt calculator functions, the following must be defined:

- Water component
- Water amount
- “Salt component” (for the salt component model) or the required ions (electroyte model).

Tip: Use the Inhibitor Calculator in the Multiflash GUI to define these components, and then save the initialization script in Excel.

Multiflash ion names (MF_ION_INAM)

Returns the array of names of cations or anions recognised by the electrolyte model.

=MF_ION_INAM(ion_type, init_script)

Arguments

Argument	Type (size)	Description
ion_type	String (40)	Specifies which kind of ion to return, can be set to either ‘CATIONS’ or ‘ANIONS’.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Ion ratio calculator for defined salinity – ion concentrations as input (MF_ION_IC)

Calculates the ion ratio to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with ion concentrations as specified. Returns the ion ratio to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ions to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_ION_IC(ion, ion_ids, ion_concs, sp_gravity, init_script)

Arguments

Argument	Type (size)	Description
ion	String	Returns the ion ratio calculated by Multiflash functions. Only Na^+ and Cl^- ions are calculated; other ion ratios are returned as zero.

<table>
<tr><th>Argument</th><th>Type (size)</th><th>Description</th></tr>
<tr><td>ion_ids</td><td>String (array)</td><td>Each string identifies a particular ion. The complete list of possible ion identifiers is as follows:
<table>
<tr><td>Na+</td><td>sodium ion</td></tr>
<tr><td>Ca++</td><td>calcium ion</td></tr>
<tr><td>Mg++</td><td>magnesium ion</td></tr>
<tr><td>K+</td><td>Potassium ion</td></tr>
<tr><td>Sr++</td><td>Strontium ion</td></tr>
<tr><td>Ba++</td><td>barium ion</td></tr>
<tr><td>Fe++</td><td>ferrous ion</td></tr>
<tr><td>Cl-</td><td>chloride ion</td></tr>
<tr><td>SO4--</td><td>sulphate ion</td></tr>
<tr><td>HCO3-</td><td>bicarbonate ion</td></tr>
<tr><td>Br-</td><td>Bromine ion</td></tr>
</table></td></tr>
<tr><td>ion_concs</td><td>Real (double, array)</td><td>Gives each ion concentration in mg/litre. The list of concentrations must correspond to and be the same length as the list of ion identifiers passed in ‘ion_ids’.</td></tr>
<tr><td>TDS</td><td>Real (double)</td><td>Total dissolved salt in mg/litre.</td></tr>
<tr><td>sp_gravity</td><td>Real (double)</td><td>The specific gravity or density in g/cm^3 of the solution. This argument can be left undefined, in which case the value is estimated on the basis of the solution’s composition.</td></tr>
<tr><td>init_script</td><td>String array (255)</td><td>The initialization script that defines the Multiflash problem.</td></tr>
</table>

Ion ratio calculator for defined salinity - salt mass fractions as input (MF_ION_MASS_FR)

Calculates the ion ratio to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with salt mass (weight) fractions as specified. Returns the ion ratio to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ions to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_ION_MASS_FR(ion, salt_ids, salt_mass_fracs, init_script)

Arguments

Argument	Type (size)	Description
ion	String	Returns the ion ratio calculated by Multiflash functions. Only Na^+ and Cl^- ions are calculated; other ion ratios are returned as zero.
salt_ids	String (array)	Each string identifies a particular salt. The complete list of possible salt identifiers is as follows: NaCl — sodium chloride CaCl2 — calcium chloride MgCl2 — magnesium chloride KCl — Potassium chloride SrCl2 — Strontium chloride BaCl2 — barium chloride FeCl2 — ferrous chloride Na2SO4 — sodium sulphate NaHCO3 — sodium bicarbonate NaBr — Sodium bromide
salt_ mass_ fracs	Real (double, array)	Gives each salt mass (weight) fraction. The list of salt mass fractions must correspond to and be the same length as the list of salt identifiers passed in 'salt_ids'.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Ion ratio calculator for defined salinity - salt molalities as input (MF_ION_MOLAL)

Calculates the ion ratio to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with salt molalities as specified. Returns the ion ratio to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ions to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_ION_MOLAL(ion, salt_ids, salt_molalities, init_script)

Arguments

Argument	Type (size)	Description
ion	String	Returns the ion ratio calculated by Multiflash functions. Only Na^+ and Cl^- ions are calculated; other ion ratios are returned as zero.

<table>
<tr><th>Argument</th><th>Type (size)</th><th>Description</th></tr>
<tr><td>salt_ids</td><td>String (array)</td><td>Each string identifies a particular salt. The complete list of possible salt identifiers is as follows:
<table>
<tr><td>NaCl</td><td>sodium chloride</td></tr>
<tr><td>CaCl2</td><td>calcium chloride</td></tr>
<tr><td>MgCl2</td><td>magnesium chloride</td></tr>
<tr><td>KCl</td><td>Potassium chloride</td></tr>
<tr><td>SrCl2</td><td>Strontium chloride</td></tr>
<tr><td>BaCl2</td><td>barium chloride</td></tr>
<tr><td>FeCl2</td><td>ferrous chloride</td></tr>
<tr><td>Na2SO4</td><td>sodium sulphate</td></tr>
<tr><td>NaHCO3</td><td>sodium bicarbonate</td></tr>
<tr><td>NaBr</td><td>Sodium bromide</td></tr>
</table></td></tr>
<tr><td>salt_molal-ities</td><td>Real (double, array)</td><td>Gives each salt molality. The list of salt molalities must correspond to and be the same length as the list of salt identifiers passed in ‘salt_ids’.</td></tr>
<tr><td>init_script</td><td>String array (255)</td><td>The initialization script that defines the Multiflash problem.</td></tr>
</table>

Ion ratio calculator for defined salinity - salt mole fractions as input (MF_ION_MOLE_FR)

Calculates the ion ratio to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with salt mole fractions as specified. Returns the ion ratio to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ions to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_ION_MOLE_FR(ion, salt_ids, salt_mole_fracs init_script)

Arguments

Argument	Type (size)	Description
ion	String	Returns the ion ratio calculated by Multiflash functions. Only Na^+ and Cl^- ions are calculated; other ion ratios are returned as zero.

<table>
<tr><th>Argument</th><th>Type (size)</th><th>Description</th></tr>
<tr><td>salt_ids</td><td>String (array)</td><td>Each string identifies a particular salt. The complete list of possible salt identifiers is as follows:
<table>
<tr><td>NaCl</td><td>sodium chloride</td></tr>
<tr><td>CaCl2</td><td>calcium chloride</td></tr>
<tr><td>MgCl2</td><td>magnesium chloride</td></tr>
<tr><td>KCl</td><td>Potassium chloride</td></tr>
<tr><td>SrCl2</td><td>Strontium chloride</td></tr>
<tr><td>BaCl2</td><td>barium chloride</td></tr>
<tr><td>FeCl2</td><td>ferrous chloride</td></tr>
<tr><td>Na2SO4</td><td>sodium sulphate</td></tr>
<tr><td>NaHCO3</td><td>sodium bicarbonate</td></tr>
<tr><td>NaBr</td><td>Sodium bromide</td></tr>
</table></td></tr>
<tr><td>salt_ mole_ fracs</td><td>Real (double, array)</td><td>Gives each salt mole fraction. The list of salt mole fractions must correspond to and be the same length as the list of salt identifiers passed in ‘salt_ ids’.</td></tr>
<tr><td>init_script</td><td>String array (255)</td><td>The initialization script that defines the Multiflash problem.</td></tr>
</table>

Ion ratio calculator for defined salinity - total dissolved salt as input (MF_ION_TDS)

Calculates the ion ratio to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with total dissolved salt as specified. Returns the ion ratio to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ions to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_ION_TDS(ion, TDS, sp_gravity, init_script)

Arguments

Argument	Type (size)	Description
ion	String	Returns the ion ratio calculated by Multiflash functions. Only Na^+ and Cl^- ions are calculated; other ion ratios are returned as zero.
TDS	Real (double)	Total dissolved salt in mg/litre.

Argument	Type (size)	Description
sp_gravity	Real (double)	The specific gravity or density in g/cm^3 of the solution. This argument can be left undefined, in which case the value is estimated on the basis of the solution's composition.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Salinity calculator – ion concentrations as input (MF_SAL_IC)

Calculates the amount of 'salt component' to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with ion concentrations as specified. Returns the number of moles of 'salt component' to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of 'salt component' to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_SAL_IC(ion_ids, ion_concs, sp_gravity, init_script)

Arguments

Argument	Type (size)	Description
ion_ids	String (array)	Each string identifies a particular ion. The complete list of possible ion identifiers is as follows: Na+ — sodium ion Ca++ — calcium ion Mg++ — magnesium ion K+ — Potassium ion Sr++ — Strontium ion Ba++ — barium ion Fe++ — ferrous ion Cl- — chloride ion SO4-- — sulphate ion HCO3- — bicarbonate ion Br- — Bromine ion
ion_concs	Real (double, array)	Gives each ion concentration in mg/litre. The list of concentrations must correspond to and be the same length as the list of ion identifiers passed in 'ion_ids'.
sp_gravity	Real (double)	The specific gravity or density in g/cm^3 of the solution. This argument can be left undefined, in which case the value is estimated on the basis of the solution's composition.

Argument	Type (size)	Description
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Salinity calculator – salt mass fractions as input (MF_SAL_MASS_FR)

Calculates the amount of 'salt component' to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with salt mass (weight) fractions as specified.Returns the number of moles of 'salt component' to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of 'salt component' to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_SAL_MASS_FR(salt_ids, salt_mass_fracs, init_script)

Arguments

Argument	Type (size)	Description
salt_ids	String (array)	Each string identifies a particular salt. The complete list of possible salt identifiers is as follows: NaCl – sodium chloride CaCl2 – calcium chloride MgCl2 – magnesium chloride KCl – Potassium chloride SrCl2 – Strontium chloride BaCl2 – barium chloride FeCl2 – ferrous chloride Na2SO4 – sodium sulphate NaHCO3 – sodium bicarbonate NaBr – Sodium bromide
salt_mass_fracs	Real (double, array)	Gives each salt mass (weight) fraction. The list of salt mass fractions must correspond to and be the same length as the list of salt identifiers passed in 'salt_ids'.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Salinity calculator – salt molalities as input (MF_SAL_MOLAL)

Calculates the amount of 'salt component' to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with salt molalities as specified. Returns the number of

moles of ‘salt component’ to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ‘salt component’ to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_SAL_MOLAL(salt_ids, salt_molalities, init_script)

Arguments

Argument	Type (size)	Description
salt_ids	String (array)	Each string identifies a particular salt. The complete list of possible salt identifiers is as follows: NaCl — sodium chloride CaCl2 — calcium chloride MgCl2 — magnesium chloride KCl — Potassium chloride SrCl2 — Strontium chloride BaCl2 — barium chloride FeCl2 — ferrous chloride Na2SO4 — sodium sulphate NaHCO3 — sodium bicarbonate NaBr — Sodium bromide
salt_molal-ities	Real (double, array)	Gives each salt molality. The list of salt molalities must correspond to and be the same length as the list of salt identifiers passed in ‘salt_ids’.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Salinity calculator – salt mole fractions as input (MF_SAL_MOLE_FR)

Calculates the amount of ‘salt component’ to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with salt mole fractions as specified. Returns the number of moles of ‘salt component’ to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of ‘salt component’ to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_SAL_MOLE_FR(salt_ids, salt_mole_fracs, init_script)

Arguments

Argument	Type (size)	Description
salt_ids	String (array)	Each string identifies a particular salt. The complete list of possible salt identifiers is as follows: NaCl — sodium chloride CaCl2 — calcium chloride MgCl2 — magnesium chloride KCl — Potassium chloride SrCl2 — Strontium chloride BaCl2 — barium chloride FeCl2 — ferrous chloride Na2SO4 — sodium sulphate NaHCO3 — sodium bicarbonate NaBr — Sodium bromide
salt_ mole_ fracs	Real (double, array)	Gives each salt mole fraction. The list of salt mole fractions must correspond to and be the same length as the list of salt identifiers passed in 'salt_ ids'.
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Salinity calculator – total dissolved salt as input (MF_SAL_TDS)

Calculates the amount of 'salt component' to add to water that will give the same inhibition effect on gas hydrate or ice formation as a saline solution with total dissolved salt as specified. Returns the number of moles of 'salt component' to be added per mole of water if the input units for amounts are currently set to molar units; returns the mass (weight) of 'salt component' to be added per unit mass (weight) of water if the input units for amounts are currently set to mass units.

=MF_SAL_TDS(TDS, sp_gravity, init_script)

Arguments

Argument	Type (size)	Description
TDS	Real (double)	Total dissolved salt in mg/litre.
sp_gravity	Real (double)	The specific gravity or density in g/cm^3 of the solution. This argument can be left undefined, in which case the value is estimated on the basis of the solution's composition.

Argument	Type (size)	Description
init_script	String array (255)	The initialization script that defines the Multiflash problem.

Units and conversion functions

Input units (MF_IU)

Returns the name of the current input units for a given property.

=MF_IU(property, init_script)

Arguments

<table>
<tr><th>Argument</th><th>Type</th><th>Description</th></tr>
<tr><td>property</td><td>String (40)</td><td>The property whose unit of measurement you are requesting. The allowed values are:
<table>
<tr><td>AMOUNTS</td><td>phase/component amounts</td></tr>
<tr><td>TEMPERATURE</td><td>Temperature</td></tr>
<tr><td>PRESSURE</td><td>Pressure</td></tr>
<tr><td>VOLUME (or DENSITY)</td><td>volume/density</td></tr>
<tr><td>ENTHALPY(or INTERNALENERGY) (or GIBBS)</td><td>enthalpy/internal energy/Gibbs energy</td></tr>
<tr><td>ENTROPY (or Cp or Cv)</td><td>entropy/Cp/Cv</td></tr>
<tr><td>SSOUND</td><td>speed of sound</td></tr>
<tr><td>VISCOSITY</td><td>Viscosity</td></tr>
<tr><td>THCOND</td><td>thermal conductivity</td></tr>
<tr><td>SURTEN</td><td>surface tension</td></tr>
</table></td></tr>
<tr><td>init_script</td><td>String (255)</td><td>The initialization script that defines the Multiflash problem.</td></tr>
</table>

Output units (MF_OU)

Returns the name of the current output units for a given property.

=MF_OU(property, init_script)

Arguments

Argument	Type	Description
property	String (40)	The property whose unit of measurement you are requesting. The allowed values are: AMOUNTS — phase/component amounts TEMPERATURE — Temperature PRESSURE — Pressure VOLUME (or DENSITY) — volume/density ENTHALPY(or INTERNALENERGY) (or GIBBS) — enthalpy/internal energy/Gibbs energy ENTROPY (or Cp or Cv) — entropy/Cp/Cv SSOUND — speed of sound VISCOSITY — Viscosity THCOND — thermal conductivity SURTEN — surface tension
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for density (MF_UND)

Returns the value of the converted density for the requested units.

=MF_UND(quantity, convert_from, convert_to, molecular_weight, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of density in the current units.

Argument	Type	Description
convert_ from	String (40)	The current unit for density. The available units are (not case sensitive): ■ mol/m3 ■ kg/m3 ■ lb/ft3 ■ lbmol/ft3 ■ mol/cm3 ■ g/cm3 ■ kmol/m3 ■ m3/mol ■ m3/kg ■ ft3/lb ■ ft3/lbmol ■ cm3/mol ■ cm3/g ■ m3/kmol
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for enthalpy (MF_UNH)

Returns the value of the converted enthalpy for the requested units.

=MF_UNH(quantity, convert_from, convert_to, molecular_weight, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of enthalpy in the current units.

Argument	Type	Description
convert_ from	String (40)	The current unit for enthalpy. The available units are (not case sensitive): ■ J/mol ■ kJ/mol ■ MJ/mol ■ kJ/kg ■ cal/mol ■ kcal/mol ■ kcal/lbmol ■ BTU/lbmol ■ cal/g ■ kcal/lb ■ BTU/lb ■ J/kg ■ J/kmol
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
molecular_ weight	Real (double)	The molecular weight used in the current units.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for entropy (MF_UNS)

Returns the value of the converted entropy for the requested units.

=MF_UNS(quantity, convert_from, convert_to, molecular_weight, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of entropy in the current units.

Argument	Type	Description
convert_ from	String (40)	The current unit for entropy. The available units are (not case sensitive): ■ J/mol/K ■ J/kmol/K ■ kJ/kmol/K ■ MJ/mol/K ■ GJ/kmol/K ■ J/kg/K ■ kJ/kg/K ■ cal/mol/K ■ kcal/lbmol/F ■ BTU/lbmol/F ■ kcal/lb/F ■ BTU/lb/F
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
molecular_ weight	Real (double)	The molecular weight used in the current units.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for pressure (MF_UNP)

Returns the value of the converted pressure for the requested units.

=MF_UNP(quantity, convert_from, convert_to, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of pressure in the current units.

Argument	Type	Description
convert_ from	String (40)	The current unit for pressure. The available units are (not case sensitive): ■ Pa ■ kPa ■ Mpa ■ bar ■ barg ■ atm ■ atmg ■ psi ■ psig ■ mmHg ■ inwg ■ at ■ atg
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for surface tension (MF_UNST)

Returns the value of the converted surface tension for the requested units.

=MF_UNST(quantity, convert_from, convert_to, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of surface tension in the current units.
convert_ from	String (40)	The current unit for surface tension. The available units are (not case sensitive): ■ N/m ■ mN/m ■ dyne/cm
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for temperature (MF_UNT)

Returns the value of the converted temperature for the requested units.

=MF_UNT(quantity, convert_from, convert_to, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of temperature in the current units.
convert_ from	String (40)	The current unit for temperature. The available units are (not case sensitive): ■ K ■ degC ■ degF ■ degR
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for thermal conductivity (MF_UNTC)

Returns the value of the converted thermal conductivity for given units.

=MF_UNTC(quantity, convert_from, convert_to, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of thermal conductivity in the current units.
convert_ from	String (40)	The current unit for thermal conductivity. The available units are (not case sensitive): ■ W/m/K ■ kW/m/K ■ kJ/hr/m/K ■ BTU/hr/ft/F ■ kcal/hr/ft/F ■ cal/s/cm/K ■ kcal/hr/cm/K ■ mW/cm/K ■ W/cm/K ■ mW/m/K
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.

Argument	Type	Description
init_script	String (255)	The initialization script that defines the Multiflash problem.

Unit conversion for viscosity (MF_UNVIS)

Returns the value of the converted viscosity for the requested units.

=MF_UNVIS(quantity, convert_from, convert_to, init_script)

Arguments

Argument	Type	Description
quantity	Real (double)	The value of viscosity in the current units.
convert_ from	String (40)	The current unit for viscosity. The available units are (not case sensitive): ■ Pas ■ cP ■ mPAs ■ microPas ■ dynes/cm2 ■ microP
convert_to	String (40)	The requested unit of the property. The available units are the same as those for convert_from.
init_script	String (255)	The initialization script that defines the Multiflash problem.

Continue
to the next
section

Troubleshooting

When you use the Multiflash Excel functions, a potential problem is most often identified by one of these values in a cell:

- #N/A - indicates that the phase wanted does not exist under the conditions specified. This is not necessarily an error. For example, if you are using the isothermal flash function (MF_PTF) then, depending on the conditions chosen, your stream might be a single phase and calculation of amounts or properties for other phases is not possible.
- #NUM! - indicates that the flash or other calculation has failed.
- #VALUE! - indicates that there is an error in the function definition.
- #NULL! - indicates that an unrecognised property or phase name has been entered.

Other types of error can occur, for example, where the calculated values are not what you expect or do not appear to change when you expect them to.

#N/A

#N/A (not available) is generated by Multiflash functions to indicate that the phase wanted does not exist under the conditions specified. This is a common situation when using any of the flash functions. Depending on the conditions chosen, your fluid can be all gas or all liquid and calculation of amounts or properties of the other phase will not be possible.

Tip: You can verify that the problem is a non-existent phase by adding an extra line to your calculations. Use PHASE for the property argument of the flash function that you are using. If the phase does not exist, the value returned is zero.

#NULL!

#NULL! is returned by a flash or chemical equilibrium function if the property or phase requested is not recognised. #NULL! can be returned even if the flash calculation has converged.

If you see this error, you must check that the property you have requested in the 'prop_wanted' argument is one of the valid values and not mis-spelt. You must also check that the phase you have requested in the 'phase_wanted' argument corresponds to one of the phases defined in the initialization script, for example, GAS or LIQUID1.

#NUM!

#NUM! is generated by the Multiflash functions when a calculation fails.

Single occurrences of #NUM!

If #NUM! occurs for one temperature or pressure in a series, it is likely to be because of a convergence failure. You can verify that the problem is a convergence failure by adding an extra line to your calculations. Use STATUS for the property argument of the flash function that you are using.

Requirement: You must define the flash function for the status with all other function arguments matching those of the main calculation.

If a converged solution has been obtained, the value returned is OK. If there has been an error or a warning issued, the Multiflash error codes are returned. You can look up the meaning of an error code by choosing **Add-Ins > Multiflash Add-In > Error Codes** in the Excel interface.

Tip: A common cause of the #NUM! error is when you use the function wizard and fail to complete the function definition. This happens if you do not scroll down in the function wizard and enter all required arguments.

It may be that there is no solution, but you can investigate further by doing one or more of the following:

- Change the basis or type of solution.
- Provide a starting guess in the function definition.
- Define a key component for each liquid phase by adding the following to your initialization script:

  key liquid1 heaviest;
  key liquid2 lightest;

  Rationale: In most cases, you have defined the model in the initialization script using a model definition file (.mfl) or created the script using Multiflash for Windows. When you do this, there are four phase descriptors allocated: GAS, LIQUID1, LIQUID2 and WATER. There are no key components defined for LIQUID1 and LIQUID2 (only "not water"); defining a key component for one of these liquid phases can sometimes help find a converged solution. The script changes shown make LIQUID1 and LIQUID2 heavier and lighter liquids.

  Remember to include these additional cells in the init_script argument in your function definition. You need to be aware that if compositions change markedly in a spreadsheet, labelling of liquid phases may also change.

Multiple occurrences of #NUM!

If #NUM! is reported for a complete series of calculations there can be several causes. It is worth making sure that they are genuine errors for the current calculation as follows:

1. Select **Add-Ins > Multiflash Add-In > Clear** to remove all existing calculations. This ensures that no errors are inadvertently retained from previous function definitions.
2. Recalculate the spreadsheet by clicking on the initialization script, placing the cursor on the formula bar, and then pressing enter.

If problems persist, try reinitialising Multiflash as follows:

- Select **Add-Ins > Multiflash Add-In > Clear**, and then specify the cell range of the initialization script.

  Result: Multiflash is reinitialized and any problems with the script are displayed.

#VALUE!

#VALUE! is generated by Excel to indicate that there is an error in the function definition.

If you see the #VALUE! error code, check your function definition, preferably with the function wizard, so that you can see the values that are being specified for the arguments.

Troubleshooting other problems

Some problems can occur with the flash functions that do not generate specific error messages. The most common problems and possible resolutions are summarised below.

I changed my initialization script but my results have not changed.

Previous calculation results are remembered until you reinitialize the script or use the Clear command. Force the spreadsheet to recalculate, see *Updating worksheets* on page 17.

If the values obtained still appear incorrect, perform the same calculation using the full Multiflash application.

All my calculated values are the same.

One of the most common problems is to reference arguments incorrectly. This usually manifests itself as the same calculated value for a series of different inputs. For example, when you are calculating the bubble point temperature for a series of pressures, if you inadvertently define the pressure argument with an absolute reference, and then copy the function to other cells, the calculation always uses the same value for pressure. When you are generating multiple values in this way, the pressure reference must be a mixed reference, where only the row or column is fixed, so that the copied functions use the correct pressure value.

I created a new script using the Multiflash GUI, but my calculations fail when I reference the new script.

When you use the Multiflash GUI to generate a new initialization script for use with other calculation functions, problems can occur if the new initialization script is inappropriate for the types of calculation that you are performing in the worksheet. If you think this might be the case, try performing a sample calculation using the initialization script in the full Multiflash application.

Tip: You can call the Multiflash GUI again from Excel, and specify the range of the script that you want to use in your sample calculation.

All my functions are returning #NUM! or #NULL! even though similar calculations work in Multiflash.

The most usual cause of this problem is when you use the 'include' command in your script but you have not used the full path, including drive letter, for specifying the location of the file. You must always enclose the full path within double quotes and include the drive letter, for example:

include "C:\Program Files (x86)\Infochem\MF61\MFL Files\rks.mfl";

Continue
to the next
section

Calculation of physical properties

The Multiflash Excel interface is a very useful tool for generating tables of data.

The constant or temperature dependent physical properties of a pure component or petroleum fraction are calculated using the functions MF_PDAT and MF_PDAT_T.

The properties of a pure component in a fluid are calculated using the many flash functions available in the Multiflash Excel interface.

Example Excel files referred to are available in the MFL Files folder of your Multiflash installation, by default C:\Program Files\Infochem\MF61\ for 32-bit systems and C:\Program Files (x86)\Infochem\MF61\ for 64-bit systems.

Calculating pure component properties

You can calculate the properties of a pure component, such as thermodynamic and transport properties, under specified conditions by using the Multiflash flash functions in the Excel interface.

As an example, the file pure.xls calculates the liquid properties on the saturation line and the vapour properties of pure octane, by using the bubble point calculation function MF_TBUBF and isothermal flash MF_PTF. When you look at the file, notice the following additional features:

- The required properties are written in rows 9 and 27 so they can be referenced by the functions
- The input and output unit functions are used to display the units in rows 11 and 29.
- Constant values are entered in worksheet cells and referenced by the functions, for example, pressure in cell D23. This enables you to repeat a sequence of calculations using a different constant value.

Related Topics

» Bubble point at fixed temperature (MF_TBUBF)
» Flash at fixed pressure and temperature (MF_PTF)

Constant physical properties and temperature-dependent correlation coefficients

You can calculate the constant physical properties and temperature-dependent correlation coefficients of a pure component or petroleum fraction by using the Multiflash function MF_PDAT.

As an example, the file phyprops.xls shows how to display the constant properties and the correlation cofficients of the temperature-dependent properties. When you look at the file, notice the following additional features:

- The 'offset' argument is not required for constant properties and is omitted or given a zero value.
- For the constant properties, the #NUM! values in the table occur when a particular value is not stored for the component.
- For the temperature-dependent coefficients, the #NULL! values in the table occur because there are not enough values to fill the table.

Related Topics

» Component properties (MF_PDAT)

Temperature-dependent properties of a pure component or petroleum fraction

You can calculate the temperature-dependent properties of a pure component or a petroleum fraction for any component in a mixture, for example calculating the saturated vapour pressure of component 1 at a given temperature, by using the Multiflash function MF_PDAT_T.

For a full description of the equations and the number of correlation coefficients used for each property, see the *Multiflash Models and Physical Properties Manual*.

Related Topics

» Temperature-dependent component properties (MF_PDAT_T)

Example Excel spreadsheets

Example Excel spreadsheets

For any large set of flash calculations involving model comparison, or evaluation of the impact of operating conditions on the overall production design, the best approach is to use the Multiflash Excel functions. Example calculations are generating tables of pure component data, using linked flashes, and recycles for simple flowsheet calculations.

Example Excel files referred to are available in the MFL Files folder of your Multiflash installation, by default C:\Program Files\Infochem\MF61\ for 32-bit systems and C:\Program Files (x86)\Infochem\MF61\ for 64-bit systems.

Note: Some users want to fit their own experimental data for components not included in their version of Multiflash. To help users to do this, we have available a series of spreadsheets for fitting pure component data. They are not issued with the standard installation but we supply them on request.

Pure component properties

What does it do?

The spreadsheet pure.xls calculates the liquid properties on the saturation line and the vapour properties of pure octane, by using the bubble point calculation function MF_TBUBF and isothermal flash MF_PTF.

How does it do it?

1. The included file, ideal.mfl, defines Infodata as the source of the pure component data, the ideal model for gas and liquid phases, and the requirement for both thermodynamic and transport property data. The remainder of the initialization script specifies the required component, and the physical property level setting.

   Rationale: The physical property level default in the Excel interface encompasses heat capacity and transport properties, but not activity and fugacity coefficients.

2. The required properties are written in rows 9 and 27 so they can be referenced by the functions. The input and output unit functions are used to display the units in rows 11 and 29. Constant values are entered in worksheet cells and referenced by the functions, for example, pressure in cell D23. This enables you to repeat a sequence of calculations using a different constant value. The table of temperatures is fixed in cells H12 to H21 for liquid properties, and cells H30 to H39 for vapour properties.
3. The Multiflash functions are specified in I12 (MF_TBUBF) and I30 (MF_PTF), taking their arguments from the worksheet cells, using absolute and relative cell addressing as appropriate. Each function is then copied to fill the table where results are required.

Why is it useful?

The worksheet calculates a complete set of properties over a range of temperatures in one go. To calculate and view all the results in the Multiflash interface would be a much greater effort. You can obtain the same information for another component by replacing octane in the initialization script.

Related Topics

» Bubble point at fixed temperature (MF_TBUBF)

» Flash at fixed pressure and temperature (MF_PTF)

Stored properties of a pure component

What does it do?

The spreadsheet phyprops.xls calculates the constant physical properties and temperature-dependent correlation coefficients of a pure component or petroleum fraction by using the Multiflash function MF_PDAT.

How does it do it?

1. The included file, rksa.mfl, defines the model used. The remaining script line specify the components and petroleum fractions for which you want to calculate properties.

   Tip: You can change the model used by referencing another .mfl file that defines a different model.

2. For the constant properties, the required properties are in column F. The Multiflash function is specified in H5 and M5 (MF_PDAT) and the unit function is specified in I5 and N5 (MF_OU) to return the values for the requested units. Each function is then copied to fill the table where results are required.

   Note: The error #NUM! in the example sheet means that the properties are not available for the component.

3. For the correlation coefficients, the required properties are in column A. The Multiflash function is specified in B8 (MF_PDAT) and then copied to fill the table where results are required.

   Note: The error #NULL! in the example sheet means that there is no value for that equation.

4. For the BIPs, the component pair, temperature dependence and order are specified for the function (MF_BDAT) to use as arguments. The function is specified in B13 and then copied to fill the table where results are required.

   Note: The error #NULL! in the example sheet means that there is no value for that BIP type.

Why is it useful?

The worksheets return a complete set of properties, correlation coefficients and BIPs in one go. To view all the results in the Multiflash interface would be a much greater effort. You can obtain the same information for another component by using different component names in the initialization script.

Related Topics

» Component properties (MF_PDAT)
» Output units (MF_OU)
» BIP parameter (MF_BDAT)

Phase envelope

What does it do?

The spreadsheet phenv.xls investigates a phase envelope by using the array function MF_PHENV.

How does it do it?

1. The included file, pra.mfl, defines the model, in this case the PRA model. The remaining lines of the script specify the components and measurement units.

   Tip: You can change the model used by referencing another .mfl file that defines a different model.

2. The component names are written in column C using the function MF_COMP. The amounts of each component are entered in column E. Argument values and input conditions are entered in columns G and I. Input and output units are written in cells K7 and L7 using the functions MF_IU and MF_OU.
3. The array function MF_PHENV is specified in the block of cells K8 to M34. The three columns return the calculated temperature, pressure and status of the phase envelope calculation.

Why is it useful?

It is particularly useful to plot the phase envelope as you investigate its characteristics, because you can see when you are approaching a cricondenbar, cricondentherm or critical point. In this example, the phase line passes through a critical point, temperature as shown in cell K19 and pressure as shown in cell L19. It is marked by a 'C' in cell M19. Also, the line crosses another phase boundary showing a resultant discontinuity in gradient. This occurs at the point given in cells K26:L26 marked by a 'D' in cell M26.

Related Topics

- » Phase envelope (MF_PHENV)
- » Component name (MF_COMP)
- » Input units (MF_IU)
- » Output units (MF_OU)

Linked flashes

What does it do?

The Linked flash worksheet in the spreadsheet flash.xls investigates a simple flowsheet of two linked flashes. A schematic of the flow is shown in the worksheet.

How does it do it?

1. The included file, pra.mfl, defines the model, in this case the PRA model. The remaining lines of the script specify the components and measurement units.

   Tip: You can change the model used by referencing another .mfl file that defines a different model.

2. The component names are written in cells C7 to C12 using the function MF_COMP. The initial amounts of each component are entered in cells D7 to D12. Initial argument values and input conditions are entered in cells C17, C18 and A22 to A28. Input and output units are written in cells D17 and D18 using the functions MF_IU and MF_OU.
3. The function MF_PTF is specified as an array function in the block of cells C22 to E28, to give results for the high pressure separator.
4. The amounts for the liquid and water phases are combined to produce the input feed for the second flash in cells G22 to G28. New temperature and pressure values are entered in cells I17

and I18.

Tip: When combining streams for linking flashes or recycles, it is easier to define them in terms of total amounts rather than fractions.

Alternative: If you also wanted to remove water after the HP separator, the feed to the LP separator could be defined in the function as the LIQUID1 compositions only, that is, cells D22 to D28.

5. The function MF_PTF is specified as an array function in the block of cells H22 to J28, to give results for the low pressure separator.

Why is it useful?

Having each stage of a linked flash visible on the worksheet enables you to see at a glance the effect of different starting conditions for each stage of the overall process. Using the Multiflash GUI involves copying the output of one stage to the input of the next, and to repeat under different conditions means having to restore the original feed values.

Related Topics

- Flash at fixed pressure and temperature (MF_PTF)
- Defining a flash function as an array function in Excel
- Component name (MF_COMP)
- Input units (MF_IU)
- Output units (MF_OU)

Stream types

What does it do?

A stream type is a set of components in your fluid, consisting of one or more components. The Stream type worksheet in the spreadsheet flash.xls demonstrates how you can specify stream types within your fluid and assign a different model to this stream. This feature is most useful when using Multiflash in connection with a third party application such as a simulator. However, in this example we have extended the linked flashes example to indicate how the concept works. We have added triethylene glycol (TEG) to the fluid and amended our simple flowsheet so that the water from the high pressure separator is removed and then recombined with the water stream from the low pressure separator.

How does it do it?

1. User-defined model and phase descriptors are defined in the initialization script.

   Requirement: When you specify second and subsequent models, you must use different model and phase descriptor names.

2. A stream type, named hydrocarbon, including all the components, is defined in the initialization script for use with the high and low pressure separator flashes. The model used for this stream is the advanced Peng-Robinson.

   Tip: If you do not list the components in the stream type definition, the default is that all components are included.

3. A stream type, named TEG, including the components water and TEG, is defined so that we can look at the regeneration of TEG by boiling off water after the two water outputs are recombined. This allows us to use a different model, in this case UNIFAC, because an activity model is more appropriate for this polar aqueous stream.

4. The component names are written in cells C24 to C30 using the function MF_COMP. The initial amounts of each component are entered in cells D24 to D30. Initial argument values and input conditions are entered in rows 32 to 37. Input and output units are written using the functions MF_IU and MF_OU.
5. The function MF_PTF is specified in the block of cells C39 to E46, to give results for the high pressure separator.
6. The high pressure liquid phase is used as input to the low pressure separator.
7. The high pressure liquid phase is used as input to the low pressure separator.

   Tip: When combining streams for linking flashes or recycles, it is easier to define them in terms of total amounts rather than fractions.
8. The function MF_PTF is specified in the block of cells H39 to J46, to give results for the low pressure separator.
9. The water phases from both separators are combined to use as input for the TEG regeneration.
10. The function MF_PTF is specified in the block of cells M43 to N46, to give results for the TEG regeneration.

Why is it useful?

Having each stage of a linked flash visible on the worksheet enables you to see at a glance the effect of different starting conditions for each stage of the overall process. Additionally, defining stream types enables you to isolate specific components and phases to use in the individual flash calculations.

Related Topics

- » Flash at fixed pressure and temperature (MF_PTF)
- » Component name (MF_COMP)
- » Input units (MF_IU)
- » Output units (MF_OU)

UNIFACFIT.xls

What does it do?

This spreadsheet enables you to generate binary interaction parameters (BIPs) for activity models if UNIFAC group structures are available for your chosen components. The BIP values are adjusted so that predicted values fit the UNIFAC values.

How does it do it?

1. UNIFAC is used to calculate the phase equilibria of the binary pair.
2. You use the Excel solver tool to fit the data and generate BIPs for the binary pair. You can minimise the sum of squares of the differences between given and predicted temperature or pressure, or the differences in gas composition.

Why is it useful?

There are times when you want to use an activity model that has no stored BIPs for one or more binary pairs in your fluid. This spreadsheet is useful when you do not have the time to search for experimental data for the missing pairs, or are unable to find any.

Using the spreadsheet

The spreadsheet has several worksheets.

UNIFAC

This is the worksheet where you generate the phase equilibria data (liquid and gas phase compositions and temperature or pressure) that you are going to fit.

You specify, by entering information in the appropriate red cells:

- The databank to act as the source of pure component data, either Infodata or DIPPR if you have it.
- The names of the two components for your binary pair.
- Whether you wish to generate data for an isotherm or isobar and your chosen temperature or pressure.

The required data is generated by the Multiflash functions when the calculation in the spreadsheet is updated. It includes both column headings and plots. The composition range is fixed and the units are SI. There is no need to change these, although you can if you want.

You can see the status of the data generation in the spreadsheet. This should be OK if the UNIFAC group structures are available for your chosen components. An error status of –13201 means that the structures are missing for one or both components.

After the data is generated, select the worksheet for the model you want to use, WilsonE, UNIQUAC VLE or NRTL VLE.

Activity model worksheets

All the model worksheets function in a similar manner. The component names, conditions, temperature or pressure, and phase equilibria data are copied automatically from the UNIFAC worksheet. You then enter an initial guess for the BIPs (a useful default is to start with 0.0 for both) and use the Excel Solver tool to start the fitting procedure. The calculated BIPs appear in the cells used for the initial guesses and in the cells reporting the fitted BIPs. The equilibrium data are plotted for comparison with the UNIFAC generated data, enabling you to decide easily whether the solution is acceptable.

If the solution is not acceptable you can try one or more of the following:

- Start with a different initial estimate for the BIPs.
- Use a different criterion for minimisation. The default setting is to minimise the sum of squares of the differences between given and predicted temperature or pressure. You can also minimise differences in gas composition by changing the target cell in Excel Solver.
- Change the constraints on the values for BIPs when fitting. We have set limits on the values the BIPs can take as part of the Excel Solver tool.

After you have generated acceptable BIPs, you can store them in the Multiflash .mfl file that you are using for your fluid calculations.

VLEFIT.xls

What does it do?

This spreadsheet enables you to generate binary interaction parameters (BIPs) for activity models by adjusting the BIP values so that predicted values fit experimental data.

How does it do it?

1. Enter your experimental phase equilibria data for the binary pair.
2. You use the Excel solver tool to fit the data and generate BIPs for the binary pair. You can minimise the sum of squares of the differences between given and predicted temperature or pressure, or the differences in gas composition.

Why is it useful?

There are times when you want to use an activity model that has no stored BIPs for one or more binary pairs in your fluid. This spreadsheet is useful when you do not have the time to search for experimental data for the missing pairs, or are unable to find any, but you do have experimental equilibrium data for the required binary pairs.

Using the spreadsheet

The spreadsheet has several worksheets.

Experimental

This is the worksheet where you enter the phase equilibria data (liquid and gas phase compositions and temperature or pressure) that you are going to fit.

You specify, by entering information in the appropriate red cells:

- The databank to act as the source of pure component data, either Infodata or DIPPR if you have it.
- The names of the two components for your binary pair.
- Whether you wish to generate data for an isotherm or isobar and your chosen temperature or pressure.
- The choice of temperature and pressure units, from the list in blue below the temperature and pressure input values.
- Your experimental data.

  Note: If you have more data points than we have used, and are familiar with Excel, you can extend the range, remembering to change the cell references in the dependent activity model worksheets. Otherwise you should limit the data by choosing suitable points from the data available. If you have fewer data points, or missing data, enter #N/A in the cells which would otherwise be empty.

After you enter the data, select the worksheet for the model you want to use, WilsonE, UNIQUAC VLE or NRTL VLE.

Activity model worksheets

All the model worksheets function in a similar manner. The component names, conditions, temperature or pressure, and phase equilibria data are copied automatically from the experimental worksheet. You then enter an initial guess for the BIPs (a useful default is to start with 0.0 for both) and use the Excel Solver tool to start the fitting procedure. The calculated BIPs appear in the cells used for the initial guesses and in the cells reporting the fitted BIPs. The equilibrium data are plotted for comparison with the experimental data, enabling you to decide easily whether the solution is acceptable.

Note: If you have not used #N/A for missing data, the plots for x, y will not be relevant and should be ignored.

If the solution is not acceptable you can try one or more of the following:

- Start with a different initial estimate for the BIPs.
- Use a different criterion for minimisation. The default setting is to minimise the sum of squares of the differences between given and predicted temperature or pressure. You can also minimise differences in gas composition by changing the target cell in Excel Solver.
- Change the constraints on the values for BIPs when fitting. We have set limits on the values the BIPs can take as part of the Excel Solver tool.

After you have generated acceptable BIPs, you can store them in the Multiflash .mfl file that you are using for your fluid calculations.

Interfacial tension

What does it do?

The spreadsheet SurTen.xls enables you to calculate and plot interfacial tension between each phase of your fluid. It is also an example of how more than one initialization script can be used in worksheet calculations when you want to compare results from different models. This example has general applicability for any type of comparison that you want to perform between models.

How does it do it?

1. The initialization scripts are defined in column S.

   Tip: Use the Multiflash GUI to set up each model and fluid whose results you want to compare.
2. Pure component properties are tabulated in the block E5 to M19, to illustrate the use of the functions MF_PDAT and MF_PDAT_T.
3. Experimental data for interfacial tension is entered in cells A28 to D42.
4. Two tables are defined for calculated interfacial tension, in cells G28 to I42 and cells L28 to N42, using the function MF_PTF.

   Note: For the case of property "SURTEN", the third argument for the MF_PTF function is the second phase to consider in the interaction.

Why is it useful?

This type of worksheet enables you to compare results from multiple models, with tabulated data that can be turned into plots using the usual Excel features.

Related Topics

» Flash at fixed pressure and temperature (MF_PTF)
» Component properties (MF_PDAT)
» Temperature-dependent component properties (MF_PDAT_T)

PVT analysis

What does it do?

The spreadsheet analpvt.xls enables you to perform a PVT analysis by generating an initialization script using the Multiflash GUI. This example has general applicability for any type of calculation that involves characterization or model tuning.

How does it do it?

1. The starting initialization script is defined in cells A1 to A6.
2. You launch the Multiflash GUI, perform your required PVT analysis and write the script into the worksheet.

   Tip: Refer to the Multiflash online help to guide you through the available options.
3. Use the generated script as the initialization script for your subsequent flash calculations.

Why is it useful?

This type of worksheet enables you to characterize fluids and tune models. You can perform multiple characterizations of the original script, or tune a model using different experimental data, and use the saved scripts as the basis to perform flash calculations and compare different characterized fluids and tuned models. Your tabulated data can be plotted using all the features of Excel.

Related Topics

- » Creating an initialization script using the Multiflash GUI
- » Flash at fixed pressure and temperature (MF_PTF)
- » Component properties (MF_PDAT)
- » Temperature-dependent component properties (MF_PDAT_T)

Continue
to the next
section

Index

#

#N/A 63

#NULL! 63

#NUM! 63

#VALUE! 64

A

Absolute cell references 20

Array functions 20, 22

- defining in Excel 22
- setting up 20

B

Binary interaction parameter function 23

BIP function (MF_BDAT) 23

Bubble point at fixed pressure (MF_PBUBF) 24

Bubble point at fixed temperature (MF_TBUBF) 24

C

Cell references 20

- absolute 20
- mixed 20
- relative 20

Component feed composition (MF_FEED) 25

Component name (MF_COMP) 25

Component properties (MF_PDAT) 25

D

Databank path 2

Dew point at fixed pressure (MF_PDEWF) 24

Dew point at fixed temperature (MF_TDEWF) 25

E

Error

- #N/A 63
- #NULL! 63
- #NUM! 63
- #VALUE! 64

Example spreadsheets

- BIP generation using Excel solver tool 74
- calculating interfacial tension 76
- linked flashes 71
- model tuning 76
- phase envelope 70
- pure component properties 69
- PVT Analysis 76
- stored properties of a pure component 70
- stream types 72
- UNIFAC BIP generation 73

Excel

- BIP generation using Excel solver tool example 74
- calculating interfacial tension example 76
- example spreadsheets 69
- linked flashes example 71
- model tuning example 76
- Multiflash functions 69
- phase envelope 70
- pure component properties example 69
- PVT analysis example 76
- stored properties of a pure component example 70
- stream types 72
- UNIFAC BIP generation example 73

F

Fixed enthalpy line (MF_HPHENV) 41

Fixed entropy line (MF_SPHENV) 42

Fixed internal energy line (MF_UPHENV) 43

Fixed phase fraction flash at fixed pressure (MF_PFRACF) 32
Fixed phase fraction flash at fixed temperature (MF_TFRACF) 32
Fixed volume line (MF_VPHENV) 44
Fixed wax phase fraction flash at fixed pressure (MF_PWAT) 33
Flash at fixed enthalpy and entropy (MF_HSF) 34
Flash at fixed enthalpy and volume or density (MF_HVF) 34
Flash at fixed entropy and volume or density (MF_SVF) 35
Flash at fixed internal energy and volume or density (MF_UVF) 34
Flash at fixed pressure and enthalpy (MF_PHF) 33
Flash at fixed pressure and entropy (MF_PSF) 33
Flash at fixed pressure and internal energy (MF_PUF) 34
Flash at fixed pressure and temperature (MF_PTF) 33
Flash at fixed pressure and volume or density (MF_PVF) 34
Flash at fixed temperature and enthalpy (MF_THF) 33
Flash at fixed temperature and entropy (MF_TSF) 34
Flash at fixed temperature and internal energy (MF_TUF) 34
Flash at fixed temperature and volume or density (MF_TVF) 34
Flash function arguments 35
Function wizard 18
Functions
flash function arguments 35
MF_BDAT 23
MF_COMP 25
MF_FEED 25
MF_FRACT 33
MF_HPHENV 41
MF_HSF 34
MF_HVF 34
MF_ION_IC 45
MF_ION_INAM 45
MF_ION_MASS_FR 46
MF_ION_MOLAL 47
MF_ION_MOLE_FR 48
MF_ION_TDS 49
MF_IU 54
MF_OU 54
MF_PBUBF 24
MF_PDAT 25
MF_PDAT_T 31
MF_PDEWF 24
MF_PFRACF 32
MF_PHENV 40
MF_PHENV_COLS 41
MF_PHF 33
MF_PSF 33
MF_PTF 33
MF_PUF 34
MF_PVF 34
MF_PWAT 33
MF_SAL_IC 50
MF_SAL_MASS_FR 51
MF_SAL_MOLAL 51
MF_SAL_MOLE_FR 52
MF_SAL_TDS 53
MF_SPHENV 42
MF_SUMSQN 24
MF_SVF 35
MF_TBUBF 24
MF_TDEWF 25
MF_TFRACF 32
MF_THF 33
MF_TSF 34
MF_TUF 34
MF_TVF 34
MF_UND 55

MF_UNH 56
MF_UNP 58
MF_UNS 57
MF_UNST 59
MF_UNT 59
MF_UNTC 60
MF_UNVIS 61
MF_UPHENV 43
MF_UVF 34
MF_VERS 23
MF_VPHENV 44

H

Hardware requirements 1
Help 5

I

Initialization scripts 15
creating in Excel 16
creating with Multiflash GUI 17
updating worksheets 17
Input units (MF_IU) 54
Installation
Excel interface 3
Installing KBC SLM security 1
Installing the RLM software scurity system 2
Multiflash program 1
USB dongle security device 2
Ion ratio calculator for defined salinity – ion concentrations as input (MF_ION_IC) 45
Ion ratio calculator for defined salinity - salt mass fractions as input (MF_ION_MASS_FR) 46
Ion ratio calculator for defined salinity - salt molalities as input (MF_ION_MOLAL) 47
Ion ratio calculator for defined salinity - salt mole fractions as input (MF_ION_MOLE_FR) 48
Ion ratio calculator for defined salinity - total dissolved salt as input (MF_ION_TDS) 49

M

Mixed cell references 20
Multiflash
Excel functions 69
Multiflash ion names (MF_ION_INAM) 45
Multiflash version number (MF_VERS) 23

O

Output units (MF_OU) 54

P

Phase diagram See also Phase envelope
Phase envelope See also Plots
example 70
Phase envelope (MF_PHENV) 40
Phase envelope functions
defining 22
Plots See also Phase envelope
Properties
constant physical properties 67
Pure component properties in a mixture 67
temperature-dependent correlation coefficients 67
temperature-dependent properties 68

R

Relative cell references 20

S

Salinity calculator – ion concentrations as input (MF_SAL_IC) 50
Salinity calculator – salt mass fractions as input (MF_SAL_MASS_FR) 51
Salinity calculator – salt molalities as input (MF_SAL_MOLAL) 51
Salinity calculator – salt mole fractions as input (MF_SAL_MOLE_FR) 52
Salinity calculator – total dissolved salt as input (MF_SAL_TDS) 53
Spreadsheets See Excel

Stream types 72

example 72

Sum of squares of N cells (MF_SUMSQN) 24

T

Temperature-dependent component properties (MF_PDAT_T) 31

The column headings for phase envelope (MF_PHENV_COLS) 41

Tolerance calculation for fixed phase fraction (MF_FRACT) 33

Troubleshooting

#N/A 63

#NULL! 63

#NUM! 63

#VALUE! 64

other problems 65

U

Unit conversion for density (MF_UND) 55

Unit conversion for enthalpy (MF_UNH) 56

Unit conversion for entropy (MF_UNS) 57

Unit conversion for pressure (MF_UNP) 58

Unit conversion for surface tension (MF_UNST) 59

Unit conversion for temperature (MF_UNT) 59

Unit conversion for thermal conductivity (MF_UNTC) 60

Unit conversion for viscosity (MF_UNVIS) 61

Updating worksheets 17