Prode Properties

Properties of pure fluids and mixtures

User's Manual rel. 1.2
Microsoft Windows version

PRODE
www.prode.com

Contents

License agreement

Agreement made between Prode "Prode" and "User".

- Prode is the owner of the product "Prode Properties" including , but not limited to, dynamic link libraries, static libraries, header files, sample programs, utility programs, together with the accompanying documentation collectively known as the "software",
- User desires to obtain the right to utilize the software, the parties hereby agree as follows

Personal license

A version with limited features is available for personal use at home or in educational establishments for teaching purposes, all other applications, without first obtaining a commercial license from Prode, are expressly prohibited.

Commercial license

Upon full payment of the license fee the User has full right to utilize the purchased number of units of the software, a unit is defined as one copy of the software or any portion thereof installed on one stand-alone computer, for networked computers one unit shall be applied for each user having concurrent access and one unit shall be applied for the server.

For all applications

- Prode grants the nonexclusive, nontransferable right to use the software.
- User has a royalty free right to reproduce and distribute the software as available from Prode Internet server (personal licence) provided that User doesn't remove or alter any part of the software or of the licensing codes and threat the software as a whole unit.
- You cannot decompile, disassemble or reverse engineer the files containing the licensed software, or any backup copy, in whole or in part.
- You cannot rent, lease or sublicense the Licensed Software without express agreement by Prode.
- The software is provided "as is, where is" , Prode does not warrant that software is free from defects, or that any technical or support services provided by Prode will correct any defects which might exist.
- Prode shall not be liable for any damages that may result directly or indirectly from the use of these software programs including any loss of profits, loss of revenues, loss of data, or any incidental or consequential damages that may arise out of use of these software.
- Your license is effective upon your acceptance of this agreement and installing the Licensed Software.
- This license agreement shall remain in effect until the Licensed Software will be in use.
- You may terminate it at any time by destroying the Licensed Software together with all copies. It will also terminate if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy all copies of the Licensed Software in any form in your possession or under your control.

Customer support

Prode will provide the licensee with limited technical support by telephone, or by electronic media for a period of 60 days after delivery of the product.

How to contact Prode

you can contact Prode by phone, web page or email, the details are available at http://www.prode.com

How to obtain technical support

we welcome your comments or suggestions about our products , while the program has been tested carefully to ensure proper operation, it still may be possible for an unusual situation to result in an error. We will have a much greater chance of fixing or assisting with errors and problems if they are provided to us in a form that is repeatable.

In reporting a problem to us, the following information should be given:

- customer reference
- the version of the software
- a copy of the procedure you are running and if possible the input data
- a detailed description of what you were doing (sequence of operations) when the problem occurred
- any additional information you think may describe the problem

Introduction

Prode Properties includes a comprehensive collection of procedures to solve problems such as :

- Physical Properties Data
- Heat / Material Balance
- Process Simulation
- Process Control
- Equipment Design
- Separations
- Instrument Design
- And more ....

Technical features overview

Entirely written in C++ Prode Properties for Windows (different versions for Android, Linux etc. are available) is released in form of Dynamic Library (DLL, Active X) for direct access from Windows applications (Microsoft Excel, MATLAB, MathCad, Visual Studio applications including NET etc.).

- Support for Windows 7, 8, 10 (both 32 and 64 bit versions of library are included)
- Allows up to 500 different streams with up to 100 components per stream (user can redefine)
- Several compilations of chemical data and BIPs are available, the user can add new components and BIPs
- Comprehensive set of thermodynamic models
- Complete set of flash operations T-P, H-P, H-T, S-P, S-T, V-P, V-T, H-V, S-V, H-S, constant energy, phase-fraction...
- ‘ Functions for calculating specific properties of mixtures (critical point, Cricodentherm, Cricondenbar, cloud point etc.)
- Functions for calculating values and derivatives of fugacities, enthalpy, entropy, volume vs. temperature, pressure, composition
- Functions for calculating equilibrium lines at specified phase fractions (generation of phase diagrams)
- Functions for solving operating blocks as mixer, gas separator, liquid separator, distillation column, compressor, piping
- Functions for calculating stream properties as density, conductivity, viscosity (gaseous and liquid phases) surface tension, speed of sound, Joule Thomson etc.

Dynamic Link Libraries

A dynamic-link library is a binary file that acts as a shared library of functions that can be used simultaneously by multiple applications. these libraries are compatible with almost all Microsoft Windows applications and being compiled code they run very fast. They also integrate tightly with your application, allowing it to run as an autonomous program unit rather than being dependent on external modules of a different application.
Prode Properties includes file I/O , graphical interfaces etc. for a total of about 200000 lines of code, all the code (compiled with Microsoft Visual Studio) resides in compact libraries, ppp.dll (size 7-10 Mbytes depending from version), it’s a very compact and efficient code, easy to distribute with third part applications.

Reference Literature

Although Prode Properties may appear easy to utilize also for people without a background in chemical engineering a basic knowledge in this area is useful for selecting the proper methods and critically evaluate the results, as general introduction to the matter we suggest :

- Introduction to Chemical Engineering Thermodynamics by Smith, Van Ness, Abbott
- The Properties of Gases & Liquids, by Reid, Prausnitz, Poling
- Phase Equilibria in Chemical Engineering by Walas

What’s new

Release 1.1 [1994]
First distribution of Prode Properties as independent product, previously included in Prode Calculator, author Roberto Paron

Release 1.1c [1997]
inclusion of multi-phase equilibria (vapor, liquid, solid, phases)

Release 1.2 [2003]
inclusion of several solvers for columns and reactors

Release 1.2a-d [2013-2016]
maintenance versions for porting to different platforms (Windows, Linux, Android, IOS)

Features available vs. Versions	Personal	Base	Extended (**)
Limited number of components and features	x		
Database with 1650 (or 2300) chemicals		x	x
Database with more than 30000 BIPs		x	x
Complete set of thermodynamic models		x	x
SAFT models			EX
Asphaltene models			EX
Wax models			EX
GERG (2008)			EX
Electrolytes			EX
Hydrates Std. model	x	x	x
Hydrates Complex model			EX
Derivatives vs. P,T,W of Fg, H, S, V	x	x	x
Properties of fluids and mixtures	x	x	x
Multiphase flash with specified T, P, H, S, V	x	x	x
Additional flash operations			EX
Vapor-Liquid-liquid phase diagrams	x	x	x
Vapor-Liquid-Solid phase diagram			EX
VLE-LLE-SLE data regression	x	x	x
Raw data regression utility	x	x	x
Characterization of petroleum fractions			EX
Multiphase (gas,liquid) pipeline with heat transfer			EX
Isentropic nozzle HEM . HNE	x	x	x
Isentropic nozzle HNE-DS , NHNE			EX
Polytropic stage, vapor-liquid (gas+liquid)		x	x
Distillation (vapor-liquid)	x	x	x
Distillation (vapor-liquid-liquid, liquid-liquid)			EX
Distillation batch (and dynamic)			EX
Depressuring unit (blow-down)			EX
Reactions (different reactor types)			EX

(**) extended versions available with distribution license

Installing the program

this paraghaph provides information about system requirements, procedures for installing Prode Properties software (Windows versions) and upgrading from previous versions.

Sistem requirements

- Microsoft Windows 7, 8, 10 (32 or 64 bit)
- 2 GB of RAM installed (if used in union with Microsoft Excel or other applications)
- 20 MB of available hard-disk space

Installation procedure

1) Uninstall previous versions : if there are previous versions installed
1a) login as admin and uninstall Prode Properties
1b) make sure to delete all copies of ppp.dll in system folders (use Windows search tool to locate files)

2) download the last version of Prode Properties from this page,
http://www.prode.com/en/download.htm
and follow the instructions provided in the page

Obtain the license

Prode Properties is copy-protected, your personal copy has limited features and to access all the features you must obtain a license from Prode, different types of licenses are available,

- software copy protection (distributed via email, installation on a single computer identified by an installation code)
- hardware (dongle) copy protection (we ship the dongle, installation on single or multiple computers)
- network installation

Order a software copy protection license

the license file is based on the installation code which the program generates automatically.

- Run an applications which does access Prode PROPERTIES, once in the Properties editor the license page will show the installation code ID (see below, it's the string **2RJWAXV**)
- with order please specify the installation code

Order a hardware copy protection license

There are versions for stand-alone computer and network-connected computers, please contact Prode for details

Prode Properties Quick Start

With Prode Properties you can solve complex problems with only minor programming effort. Much of the functionality is provided by the library. In this chapter you will learn step by step how to access Properties from your favourite application. This chapter is for those of you that want to skip the tutorial and immediately start using Properties. In the following sections, you will learn how to utilize the samples provided with Properties. When you run the samples you will get a broad overview of the possibilities available from using Properties

Locating the sample files

As default the sample files, including data files, project files, and other associated files are supplied with the program and placed in subdirectories under Prode main directory.

IMPORTANT

The installation procedure creates a directory \Prode\ and different subdirectories

\Prode\C	includes definitions and code for C / C++ applications
\Prode\Excel	includes samples for Microsoft Excel
\Prode\LIB	includes the versions of the library
\Prode\MATLAB	includes definitions and code for MATLAB applications
\Prode\MATHCAD	includes definitions and code for MATHCAD applications
\Prode\Fortran	includes definitions and sample code for Fortran applications
\Prode\NET\VBprops	includes definitions and samples for Microsoft NET VB applications
\Prode\NET\C#-props	includes definitions and samples for Microsoft NET C# applications

Data files folder

Prode Properties stores several files in a directory \Prode\ in user space, the exact path depends from Windows version and settings, in Windows 7, 8, 10 the folder is C:\ProgramData\prode

the list of files includes

chem.dat
pseudo.dat
bips.dat
mod.dat
def.ppp
res.lan
lic.dat
.....

do not remove or rename these files, if Prode Properties cannot access these files (for example because they have been disseminated in different directories) an error message “Corrupted file, error reading data file” will be generated.

Make sure all users can access data files folder

IMPORTANT

When installing Prode Properties for users without full administrative rights make sure all users have read/write rights to data files folder.
if a user has no read/write rights on data files folder the program can generate errors and stop working.

Avoid errors in read / write operations

If a user doesn’t receive full read / write permissions on data files folder the program can generate a error when saving def.ppp or chem.dat files,
if you see this error you can
1) login as admin, and run Prode Properties
2) immediately before to save def.ppp or chem.dat (from Prode Properties) , with Windows File Manager manually delete the file which you wish to overwrite (def.ppp or chem.dat)
3) (from Prode Properties) save the file

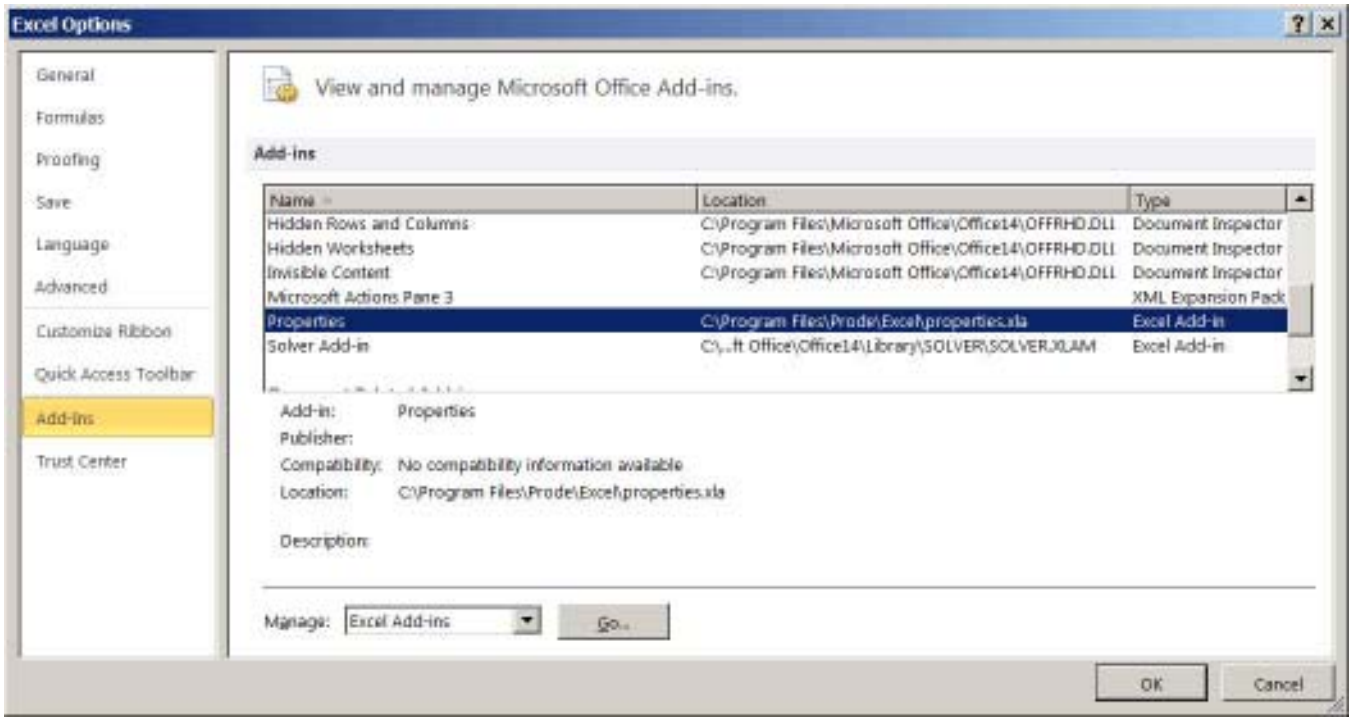
Getting Started from Microsoft Excel, part 1

Prerequisites

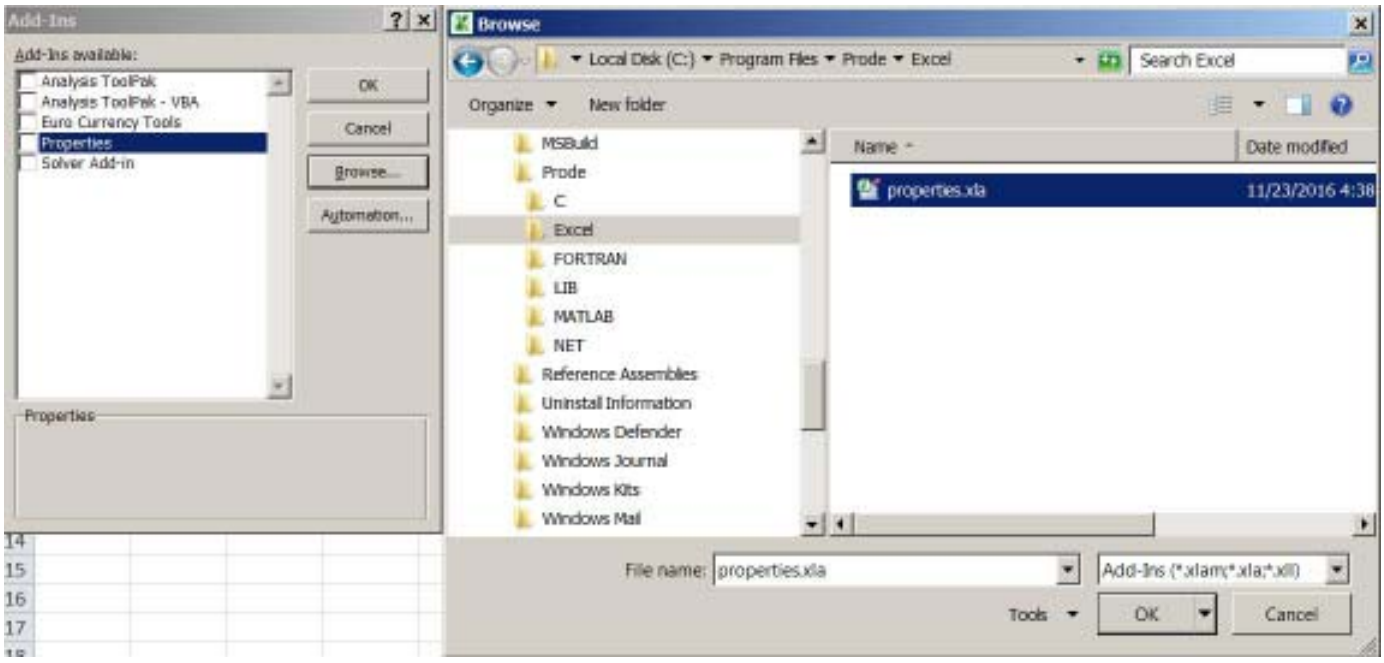
1) The different versions (32 or 64 bit) of Excel require different versions of Prode dll library, (Excel 32 requires Prode dll 32 bit while Excel 64 requires Prode dll 64 bit), when installing Prode Properties make sure to install the version suitable for your copy of Excel.
2) Before to run Prode Properties you should verify the separator in Regional Settings, by default Excel threats commas as separators and you should enter a macro as =EStrGD(1,300,1.0E5), if you wish to utilize a different separator, for example =EStrGD(1;300;1.0E5) you need to edit and modify the Regional Settings

Install Prode Properties add-in

before to use Excel you must load the add-in (file properties.xla) which instructs Excel about the methods included in Prode Properties library, you need to go through this procedure only once
In Excel 2010, 2013, 2016 open File menu, choose Options item and then Add-Ins



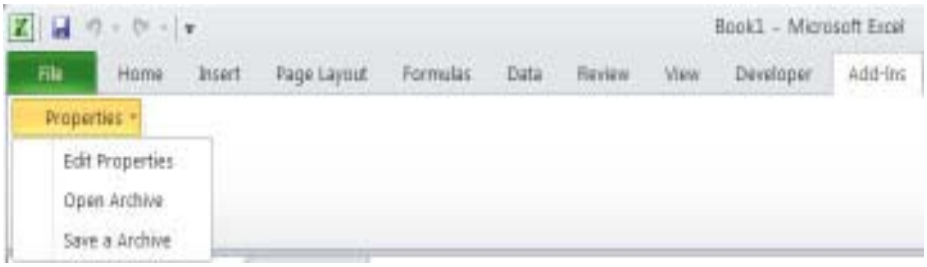
on the bottom select Manage Excel Add-Ins and click Go, you'll see a list of add-ins, some checked, some not checked. If Prode Properties isn't listed (and it won't be unless you went through this procedure earlier) browse for the properties.xla file (by default installed in C:\Program Files\Prode\Excel\) then back your way out.



Now Prode Properties should be listed in the list of add-ins, its box should be checked, click Ok to exit Excel Add-ins dialog, Prode Properties add-in provides instructions for Excel to access Prode methods and create a new menu.

Prode Properties Add-In menu

The menu for Prode Properties is available under Add-Ins tab in Microsoft Excel



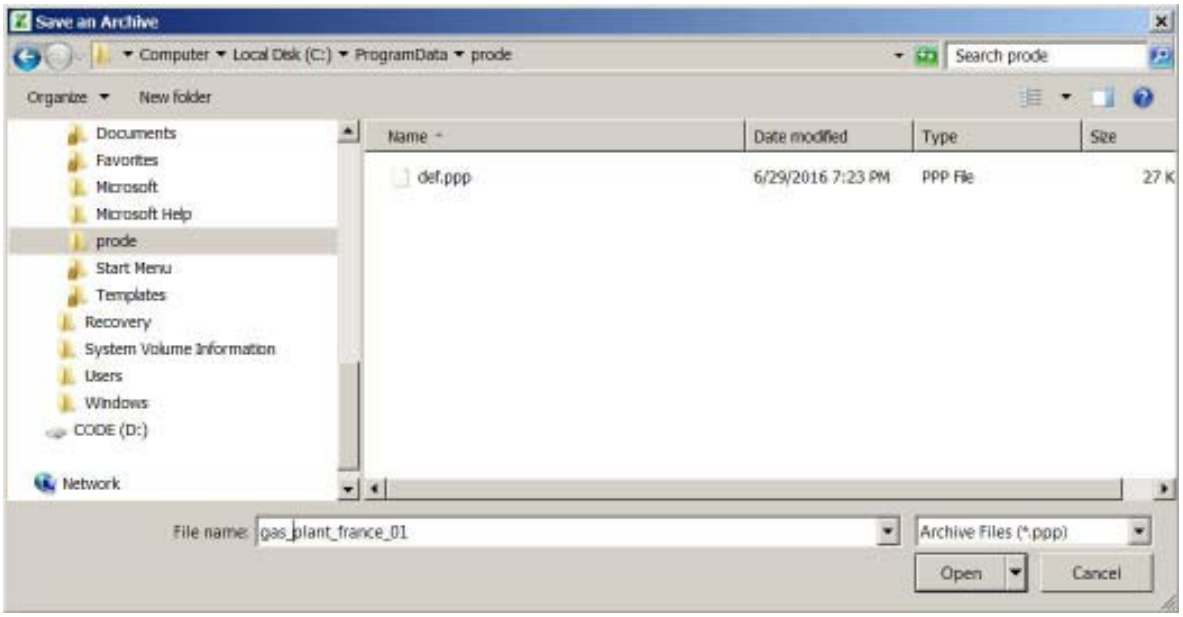
Edit Properties : open Prode Editor
Open Archive : open existing files
Save a Archive : store data in Files

Working with archives

Prode Properties stores data in memory and you may wish to save in files so that the new information will not be lost when you end the program,
each archive contains a copy of all streams (compositions, models, BIPs), plus units etc.
read the paragraph "Working with archives, save and load data, default settings" for additiona information

to Save data to archive, in Properties add-in menu select Save a Archive

then select the folder and the name of file

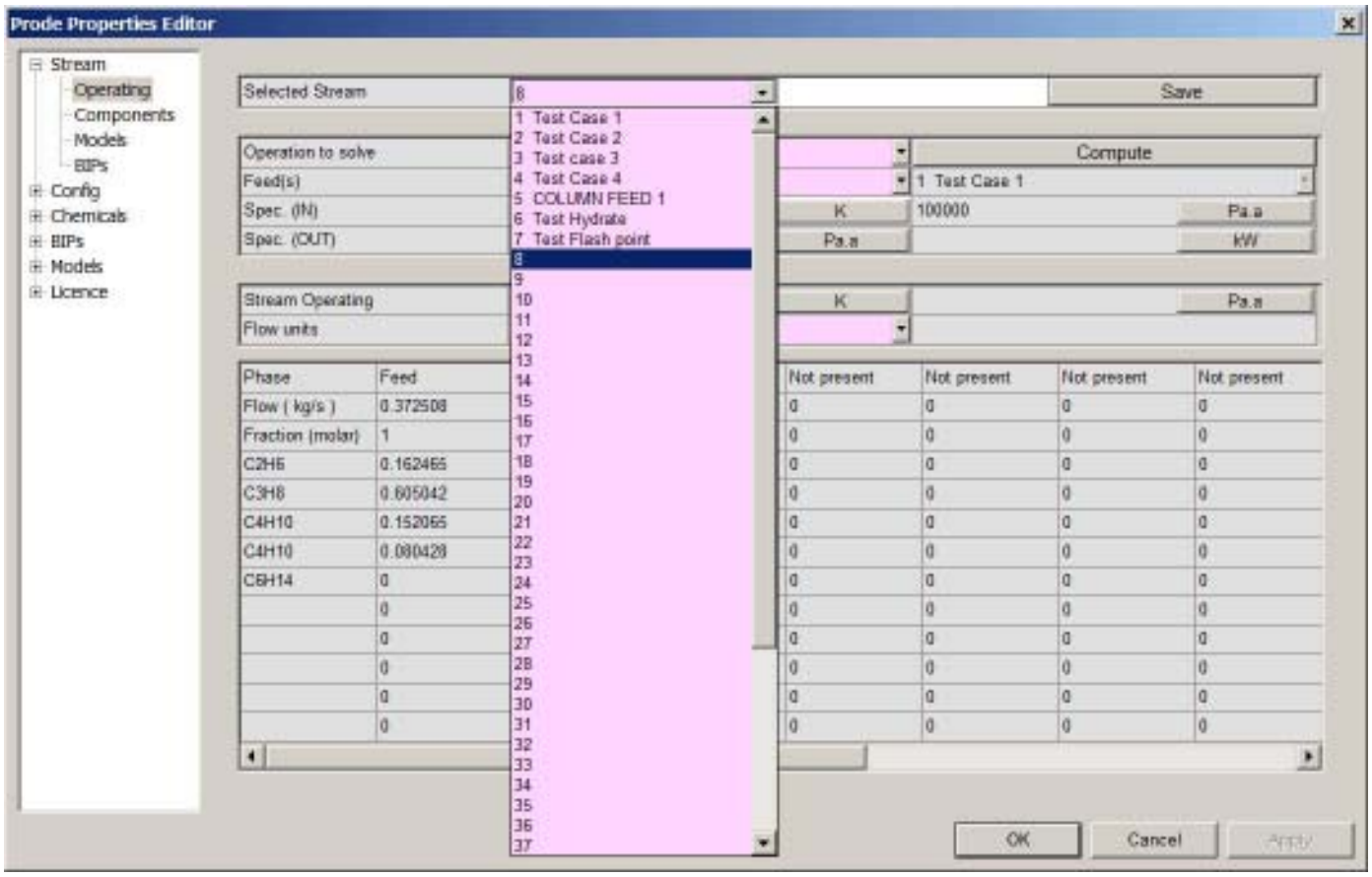


to Restore data, in Properties add-in menu select Open Archive

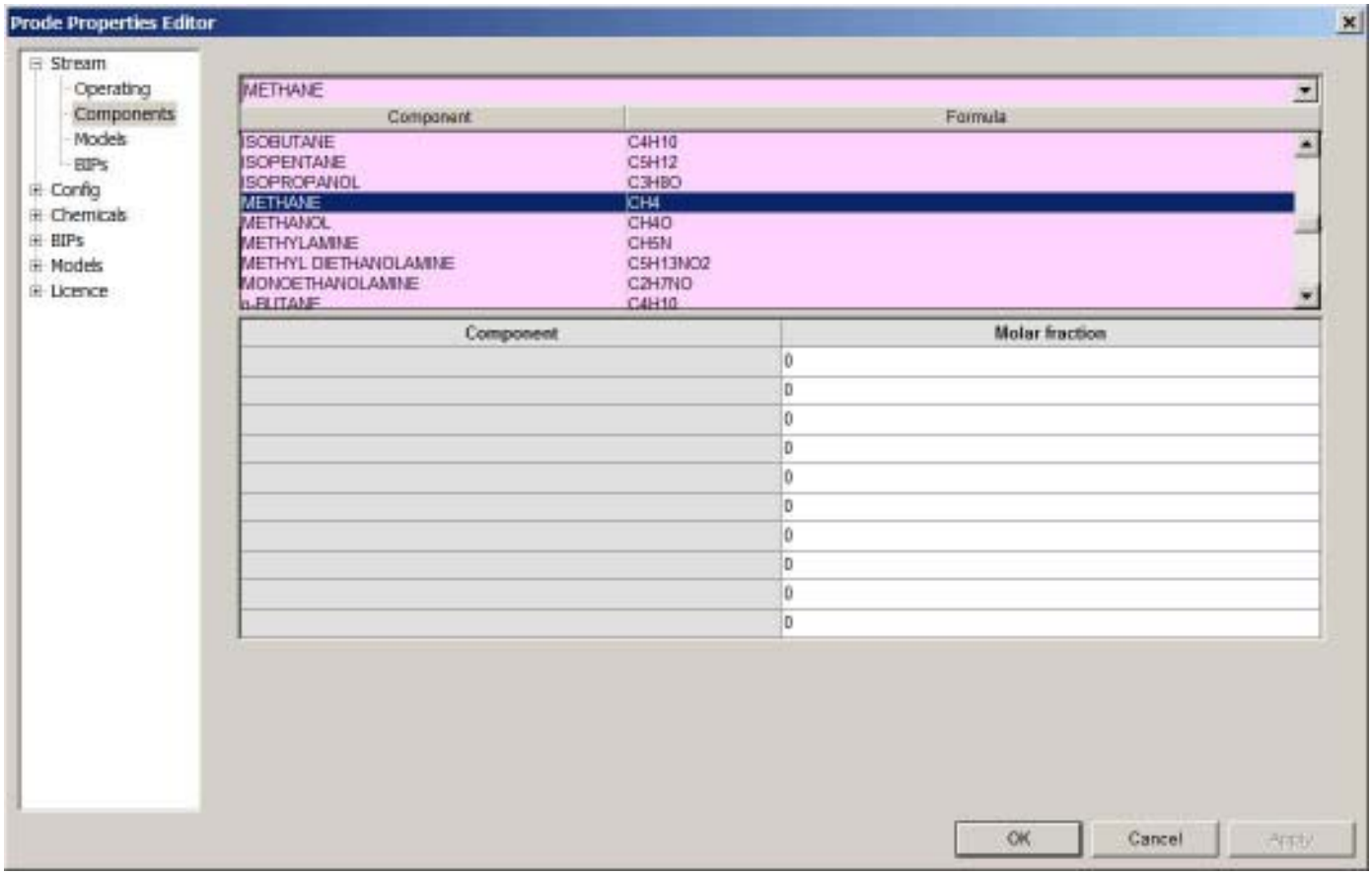
and select the archive to load

Getting Started from Microsoft Excel, part 2 operations with editor

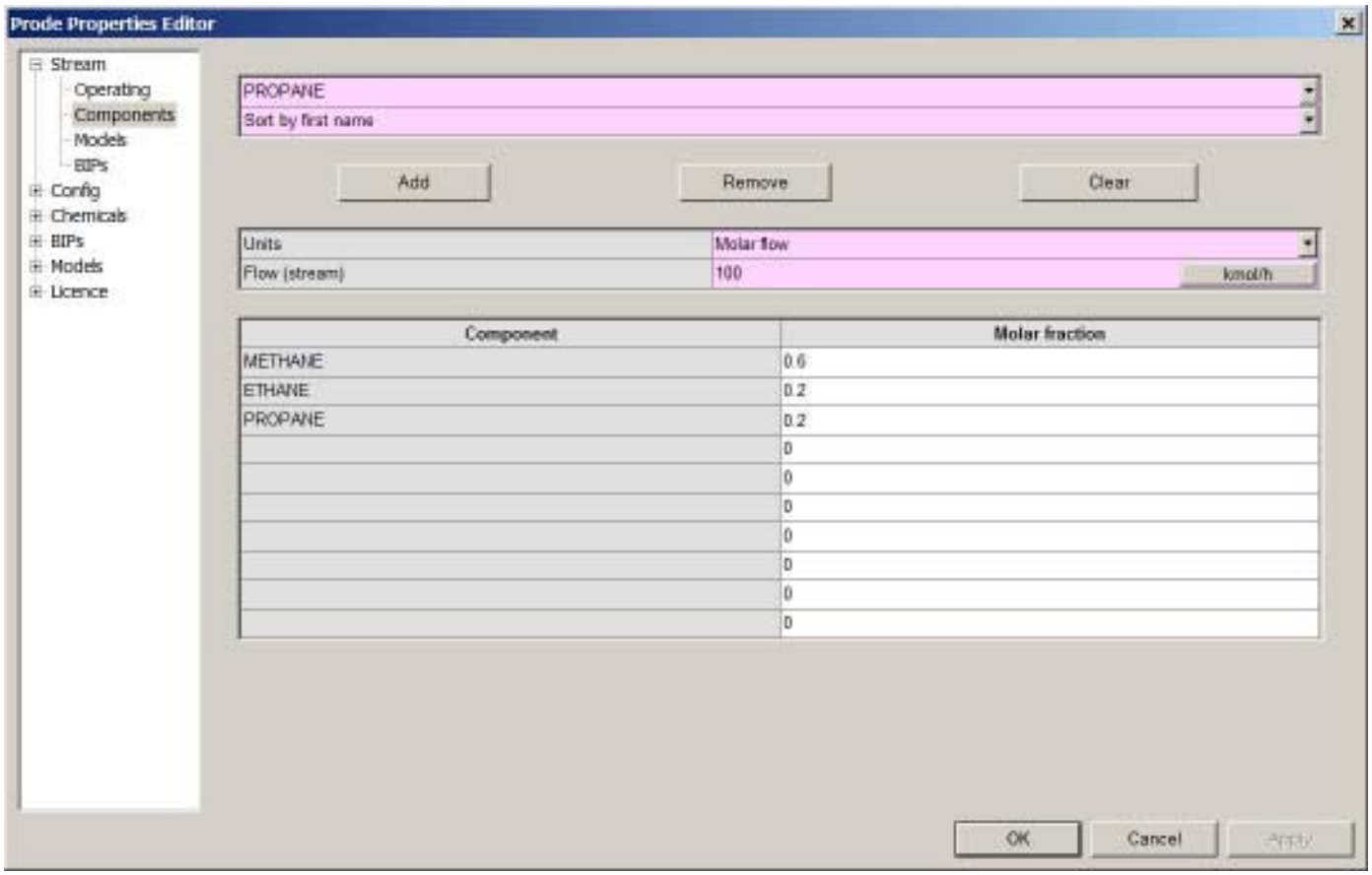
In Excel open a new page, File -> New, select a blank workbook,
in add-in menu select Edit Properties to open Prode Properties Editor, then select Operating tab and stream 8



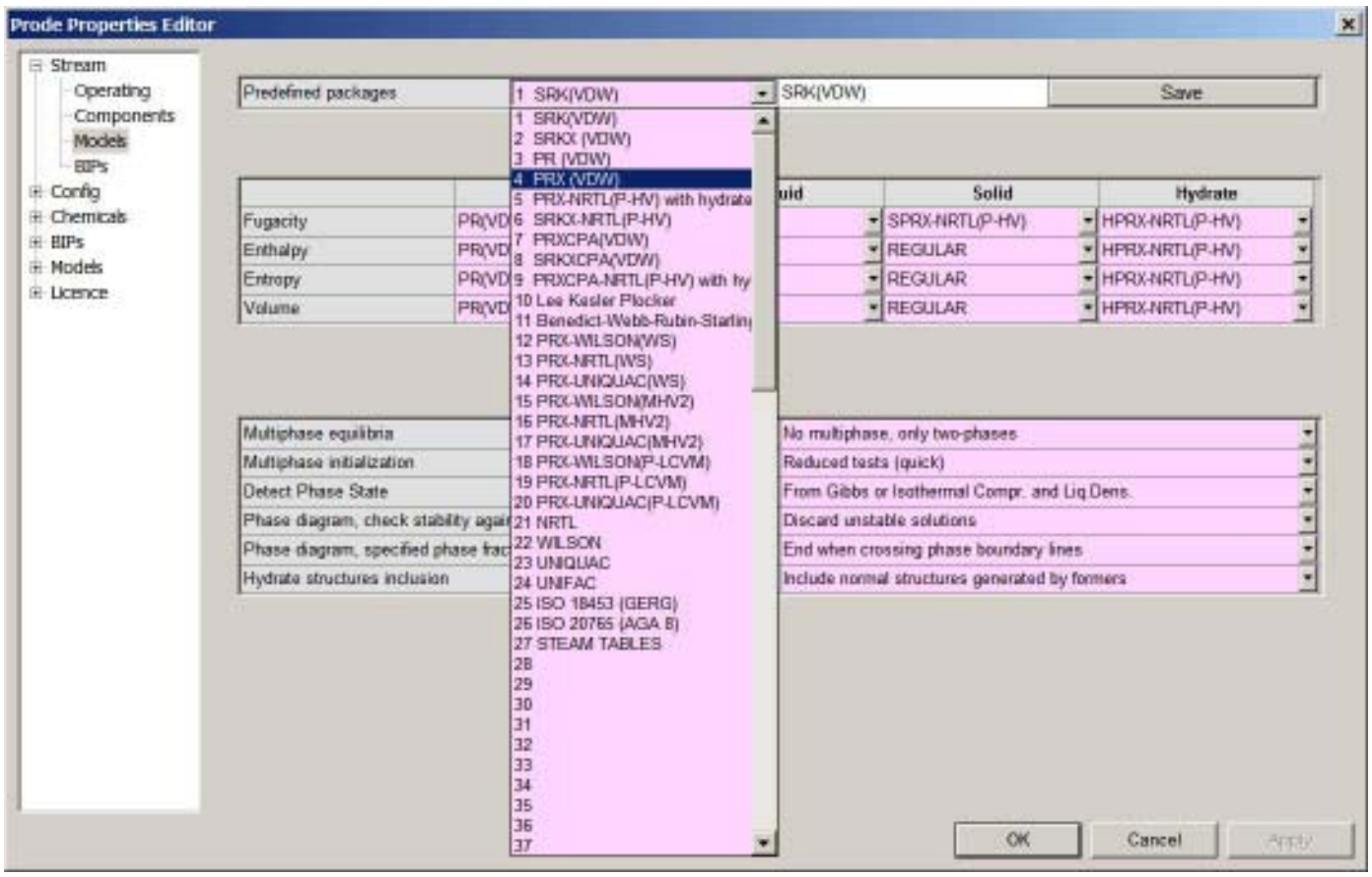
select Components tab and click Clear button (in case you see a composition defined), set units to molar flow and flow to 100 Kmol/h, from the first list select Methane as first component



then click Add button and define 0.6 as molar fraction for methane, continue and add ethane 0.2 mole fraction and propane 0.2 mole fraction



now we must define the models, goto Models tab and select PRX (PR Extended) in the list of predefined packages



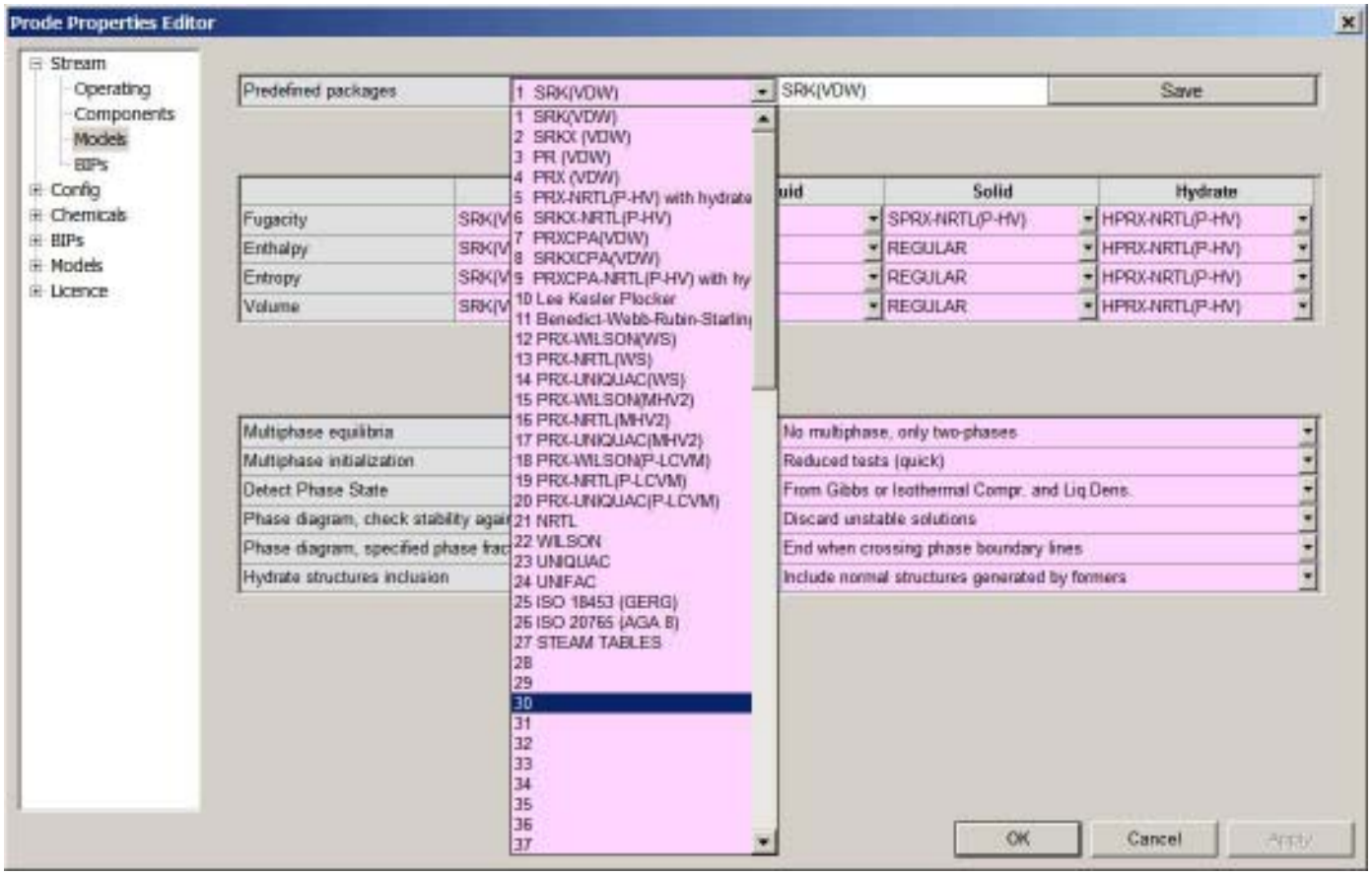
this defines Peng Robinson Extended. for all properties (fugacity, enthalpy, entropy, volume) of vapor and liquid phases.

in Models dialog you can define the different options available for phase equilibria

Multiphase equilibria	Multiphase vapor-liquid
Multiphase initialization	No multiphase, only two-phases Multiphase vapor-liquid Multiphase vapor-liquid-solid Multiphase vapor-liquid-solid-hydrate
Detect Phase State	
Phase diagram, check stability against feed	
Phase diagram, specified phase fraction lines	End when crossing phase boundary lines
Hydrate structures inclusion	Include normal structures generated by formers

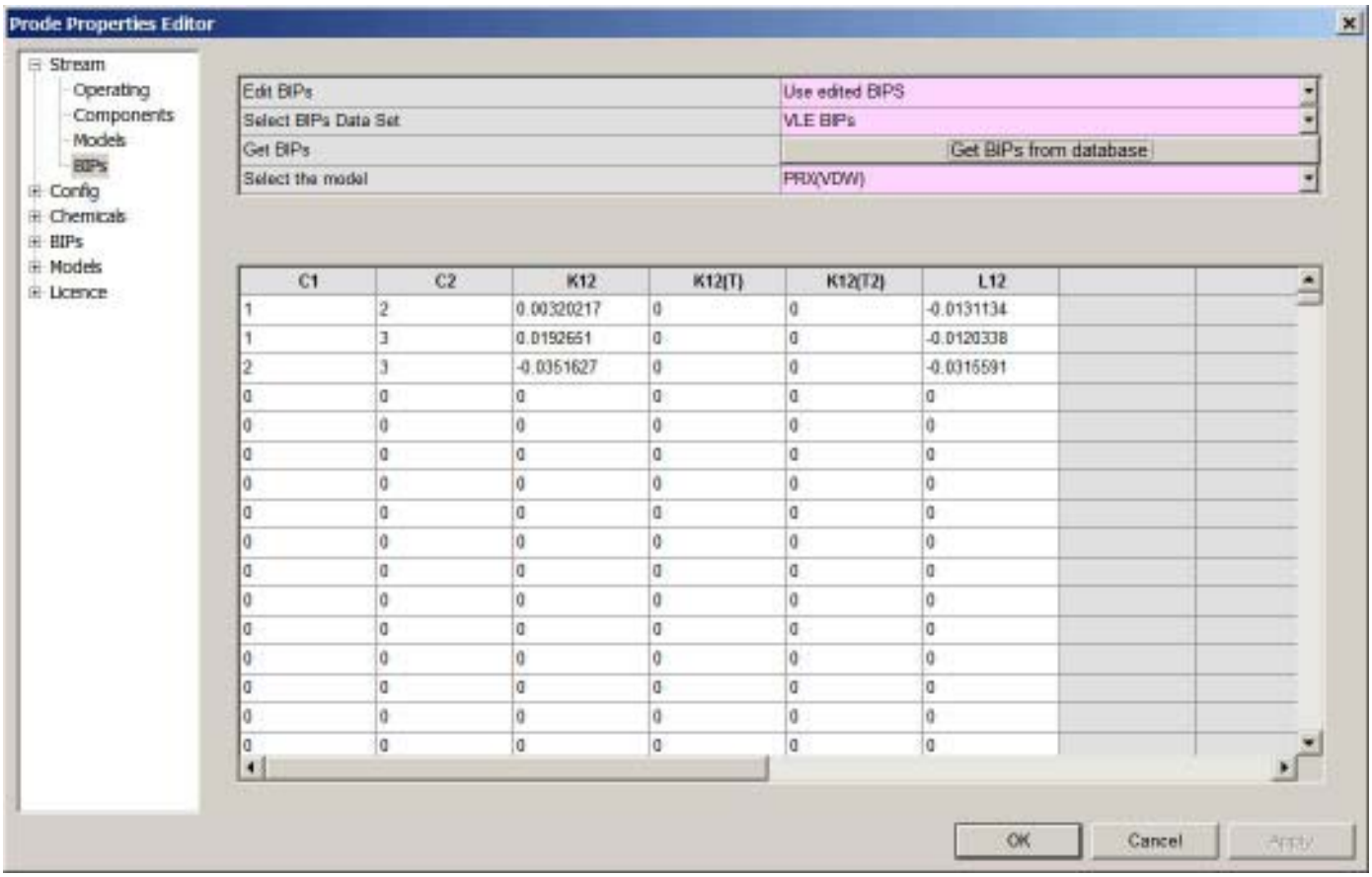
Multiphase equilibria instructs the procedure to calculate Vapor-Liquid, Vapor-Liquid-Liquid, Vapor-Liquid-Solid or Vapor-Liquid-Solid-Hydrate phase equilibria (for this stream)
Multiphase initialization and Detect phase state allow to reduce (in some cases) calculation time and improve identification of state in difficult cases
Phase diagram, check stability against feed and Phase diagram, specified phase fraction lines are specific for phase diagrams
Hydrate structures inclusion allows to include all the strcutures which may be generated by formers

You can edit / modify / define new thermo packages, to modify or create a new thermo package select a element in the list (for example 30) , then define the models for the different properties and options, enter a name (for example Test) and finally click on Save button to store your package in memory



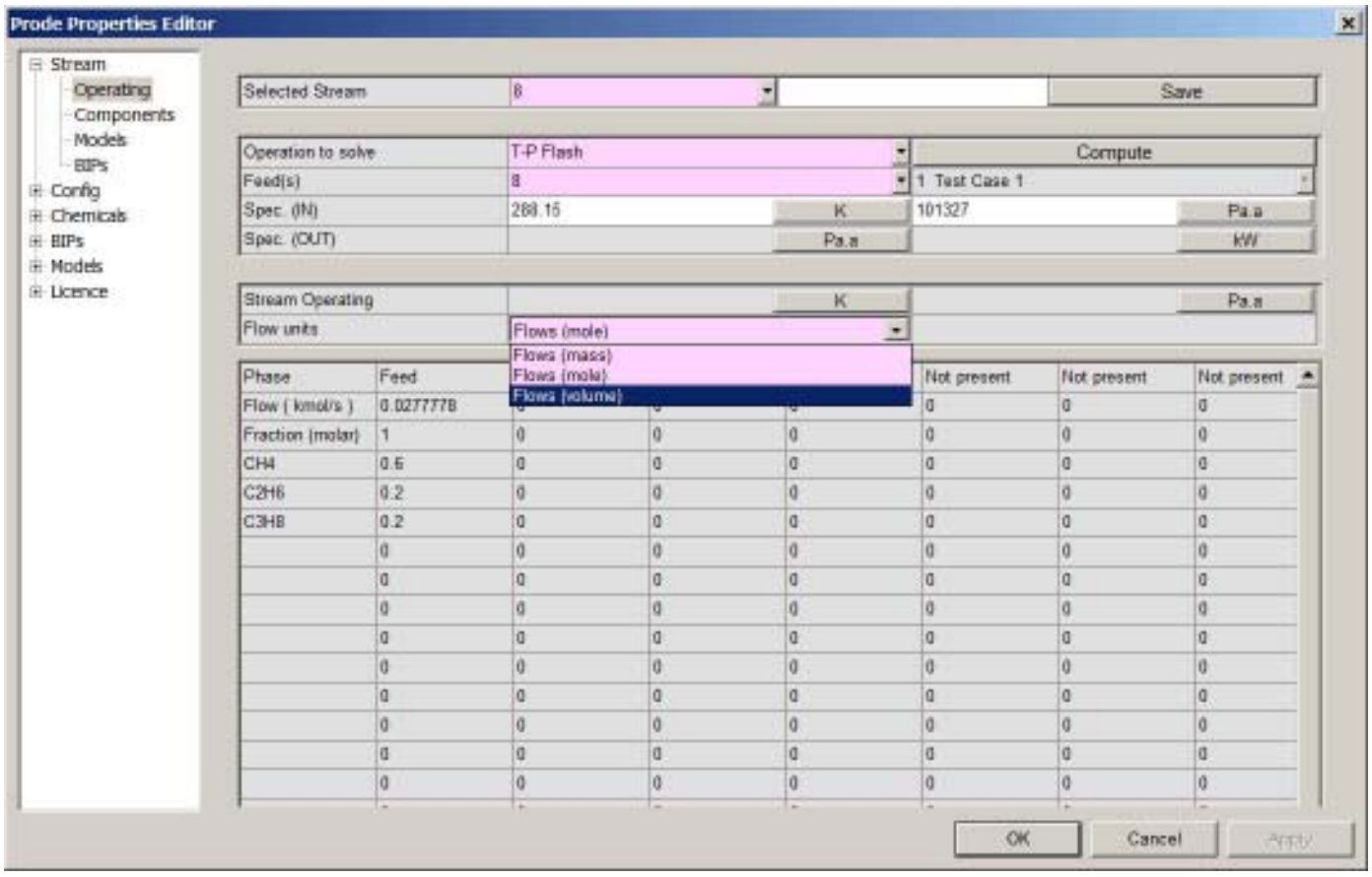
Next step, define BIPs

Many models available in Prode Properties require BIPs, Prode Properties stores BIPs in 4 different files (VLE, LLE, SLE and Hydrate) To load BIps from these files, in BIPs tab select VLE BIPs as data set and click on Get BIPs from Database to load BIPs

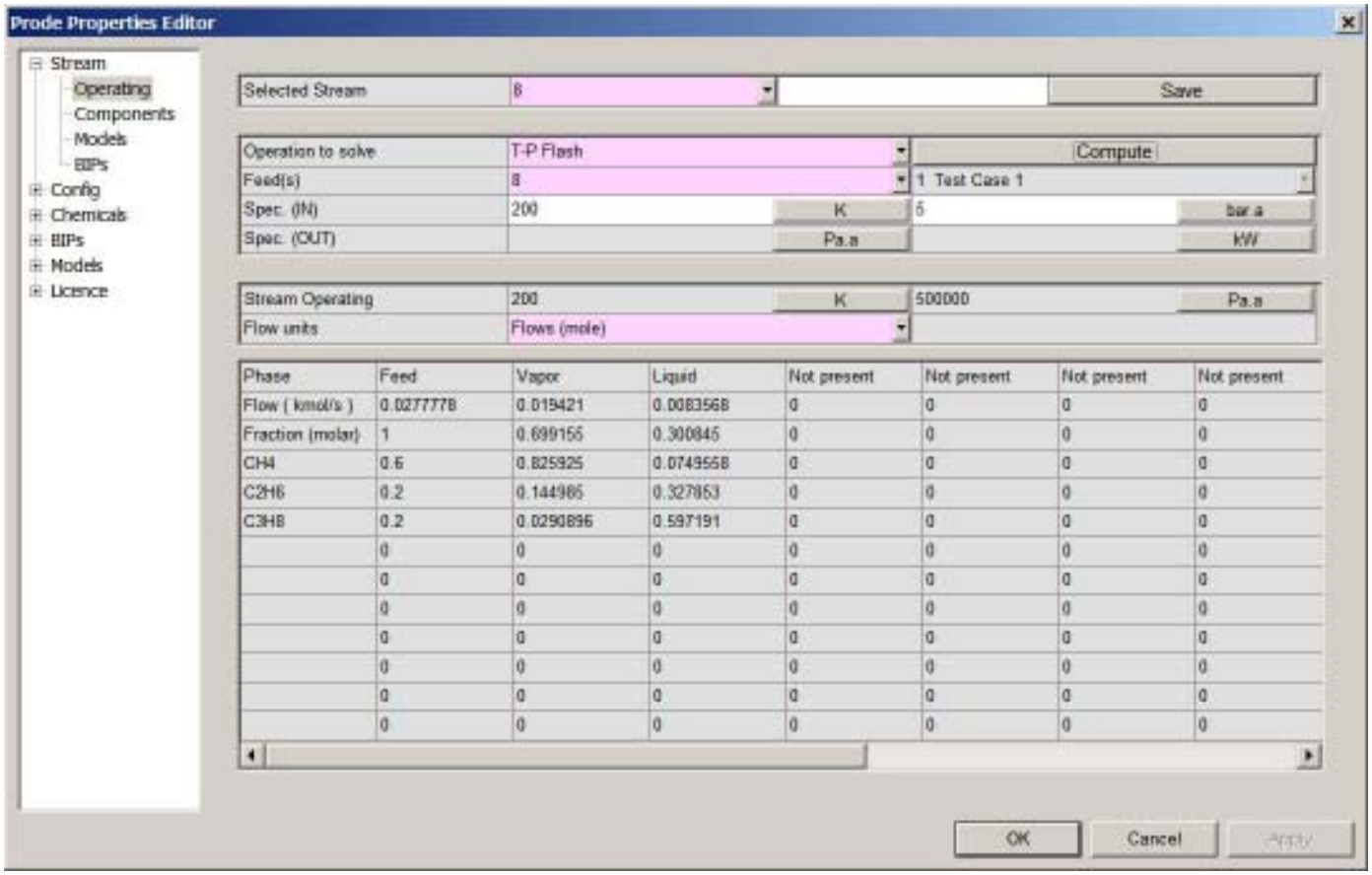


Notice that each row requires to define C1 (first component in binary) C2 (second component) and one or more BIPs depending from selected thermodynamic model ,
read “Binary Interaction Parameters (BIP)” and “Regress VLE-LLE-SLE data” for additional information about how to calculate BIPs, store BIPs in database and retrieve BIPs
Next step shows how to save stream data

To save the stream, goto Operating tab, in first grid you can define a name for this stream, for example test and then click on Save button, the page now shows the new composition C1 0.6 C2 0.2 C3 0.2



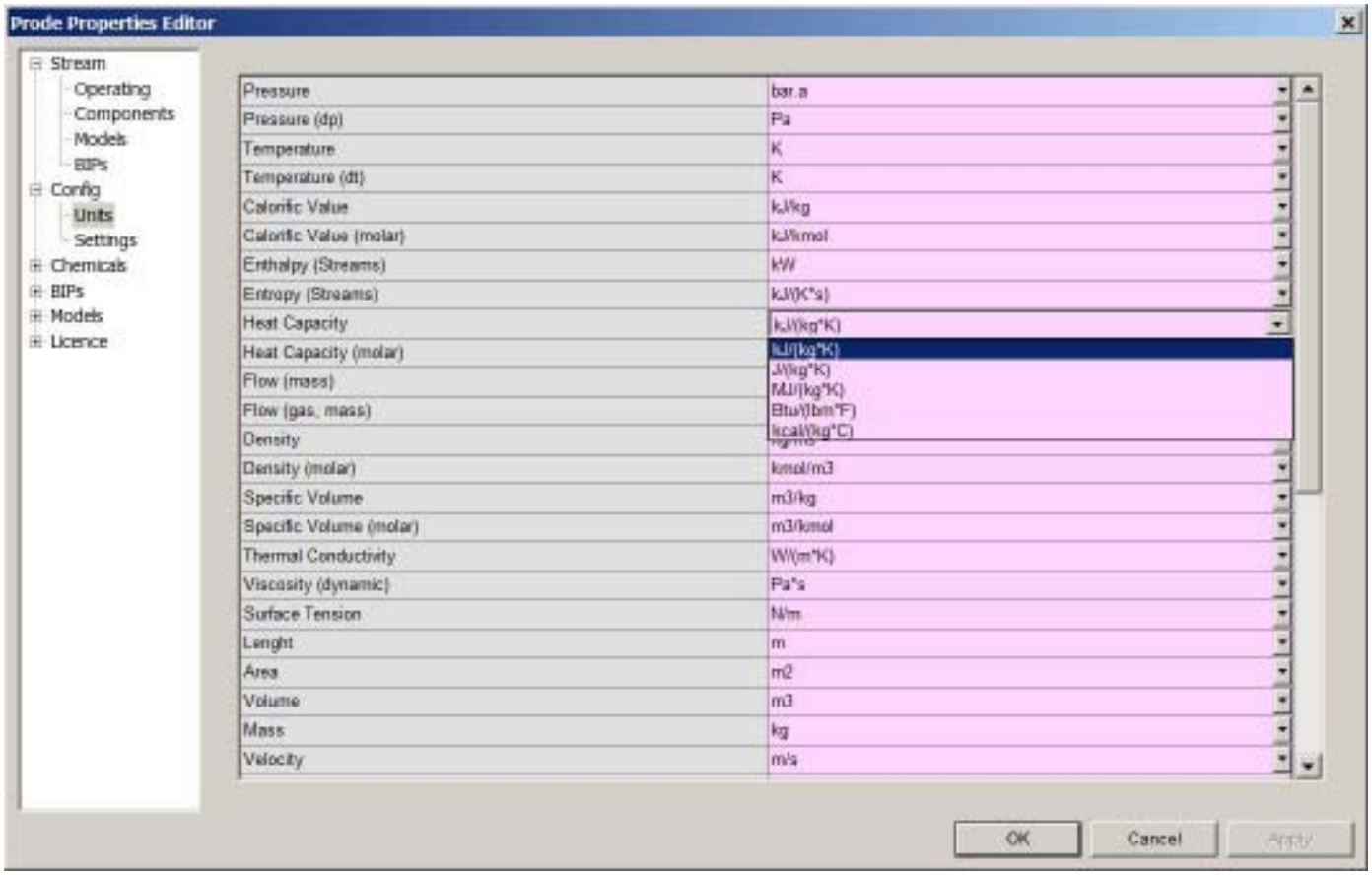
from Operating tab it is possible to solve different operations, for solving a isothermal flash in second grid select TP-Flash (the operation to solve) stream 8 as feed (remember, results will be stored in stream selected in first row, in this case stream 8) define 200 K and 5 Bar.a as operating conditions and click on Compute button



At these conditions Prode Properties (Peng Robinson Extended) calculates vapor+liquid equilibria.

Select the units :

most values require to specify unit, Prode Properties allows to define a set of predefined units,
in Prode Editor select Config and Units, define Bar.a as unit for pressure, K for temperature, Kg/m3 for density, KJ Kg/K for heat capacity



click on Ok button to leave Prode Properties editor.

Calling Prode Properties methods from Excel cells
if you are not interested goto “Getting Started from Microsoft Excel, part 3 working with predefined pages”

Once streams and units have been defined you can calculate different properties directly in Excel, we utilize the methods discussed in paragraph “Extended methods for accessing stream’s properties”, these methods allows to calculate properties at specified conditions, you may wish to read the paragraph for additional information.
In B1 we enter 150 as temperature (remember we have K as unit) and in B2 we enter 5 as pressure (remember we have Bar.a as unit), the units of calculated values are Kg/m3 for density, and Kj Kg / K for heat capacity
in B3 enter the macro =EStrLf(8,B1,B2) for calculating liquid fraction of stream 8 at temperature specified in B1 and pressure specified in B2
in B4 enter the macro =EStrLD(8,B1,B2) for calculating density of liquid fraction,
in B5 enter the macro =EStrLcp(8,B1,B2) for calculating heat capacity of liquid fraction,
in B6 enter the macro =EStrGD(8,B1,B2) for calculating density of vapor fraction,
in B7 enter the macro =EStrGcp(8,B1,B2) for calculating heat capacity of vapor fraction.

Important : you may obtain different (more accurate) values with recent versions of software

B3 =EStrLf(8,B1,B2)

	A	B	C	D
1	Temperature	150		
2	Pressure	5		
3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965		

B4 =EStrLD(8,B1,B2)

	A	B	C	D
1	Temperature	150		
2	Pressure	5		
3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965		

B5 =EStrLcp(8,B1,B2)

	A	B	C	D
1	Temperature	150		
2	Pressure	5		
3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965		

B6 =EStrGD(8,B1,B2)

	A	B	C	D
1	Temperature	150		
2	Pressure	5		
3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965		

B7 =EStrGcp(8,B1,B2)

	A	B	C	D
1	Temperature	150		
2	Pressure	5		
3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965		

In addition to the specific methods discussed in paragraph “Extended methods for accessing stream’s properties”, with Excel you can utilize all the methods exported by Prode Properties library, the list includes methods to define streams, calculate a complete set of properties and solve complex operations such as columns, reactors etc.
With different methods there are different ways to define operating conditions, for example, if your units are K (temperature) and Bar.a (pressure) you can set 150 K and 5 bar.a as operating conditions in stream 1 with the macro

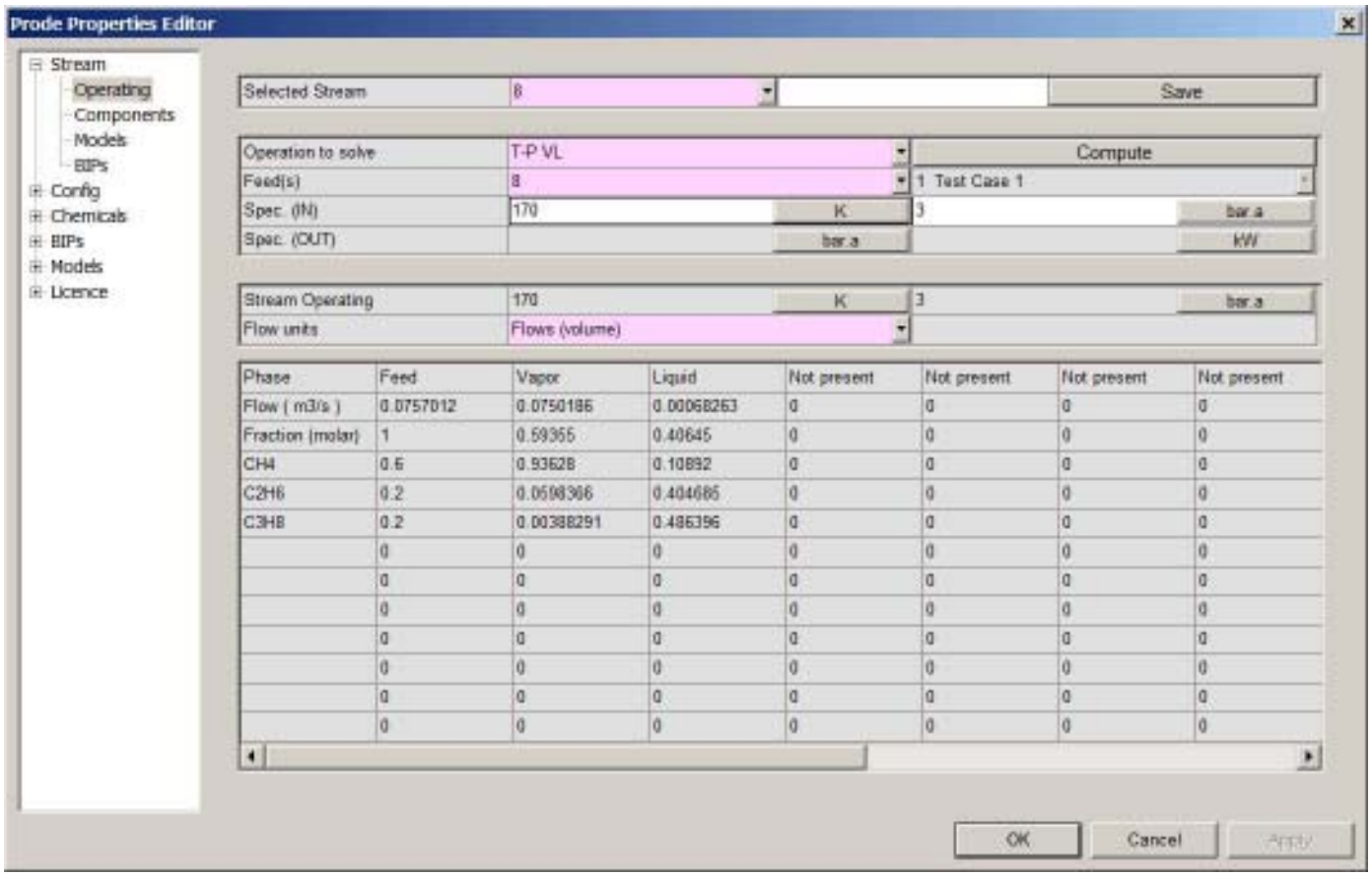
=setOp(8,150,5)

in the same way you can define the operating conditions as result, for example, of a H-P operation where you specify final pressure and enthalpy (in this case 15 Bar.and 500 kW)

= HPF(1,15,500,0)

we’ll see different applications in following examples

this example shows how to define operating conditions solving a isothermal flash in Prode Properties editor (as in previous example) and setOp() method, open Properties Editor and select the stream 8 defined in previous example with composition C1 0.6 C2 0.2 C3 0.2 , as operation to solve select T-P VL (isothermal flash vapor-liquid) , enter 170 K as operating temperature and 3 Bar.a as operating pressure and click on Compute to obtain the results.



Notice that you can obtain the same results with the macro

=setOp(8,170,5)

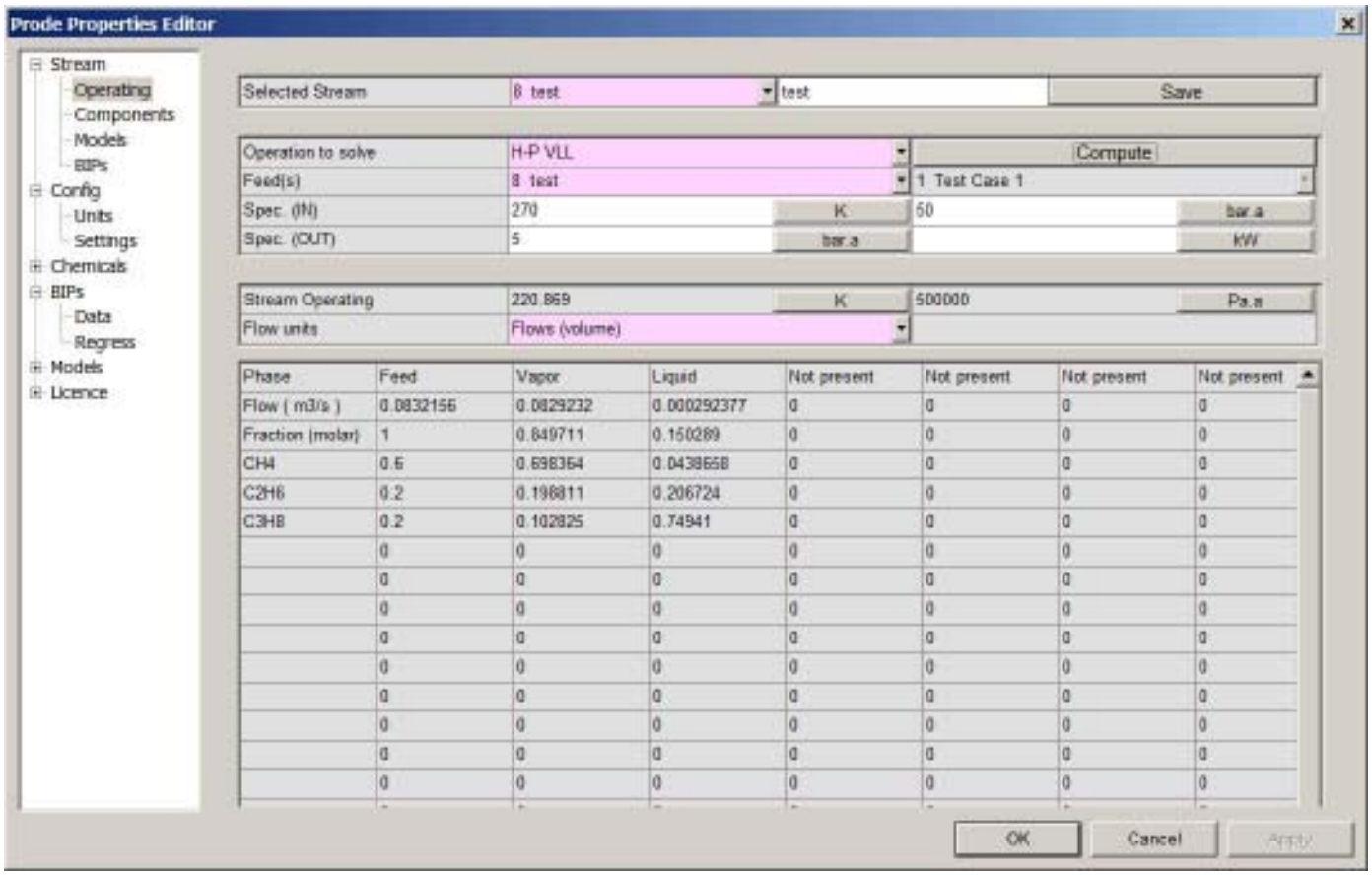
as previously examined,
in this example we adopt methods which do not include the values of temperature and pressure as input

D1 =getT(8)

	A	B	C	D
1	Temperature	150		170
2	Pressure	3		3
3	Liquid Fraction	0.51733		0.40645
4	Liquid Density	576.8338		584.905
5	Liquid Heat Capacity	2.275305		2.214435
6	Gas Density	4.086361		3.734277
7	Gas Heat Capacity	2.134471		2.051359
8				

in cell D1 enter =getT(8) to obtain operating temperature for stream 8
in cell D2 enter =getP(8) to obtain operating pressure
in cell D3 enter =StrLf(8) to obtain liquid fraction
in cell D4 enter =StrLD(8) to obtain density of liquid fraction
in cell D5 enter =StrLcp(8) to obtain heat capacity of liquid fraction
in cell D6 enter =StrGD(8) to obtain density of vapor fraction
in cell D7 enter =StrGcp(8) to obtain heat capacity of vapor fraction

Next example shows how to define the conditions solving a H-P flash operation from Prode Properties editor,
Open Prode Properties editor, in operating tab, first row select stream (stream 8 defined in previous example with composition C1 0.6 C2 0.2 C3 0.2), as Operation to solve select H-P VL , as feed stream 8, the following two rows allow to specify input conditions (temperature and pressure) and output conditions (pressure) plus the heat to add or remove, we wish to model a control valve (no heat added / removed) with inlet conditions 270 K 50 Bar.a and outlet pressure 5 bar., enter values and click on Compute button to get the results



the procedure predicts an output temperature of about 220 K with Vapor+Liquid
As in previous example you can obtain equivalent results by entering the macros

=HPF(8,5,EStrH(8,270,50),0)

where macro HPF solves the H-P operation with specified pressure 5 Bar.a and enthalpy calculated with macro EStrH(8,270,50)

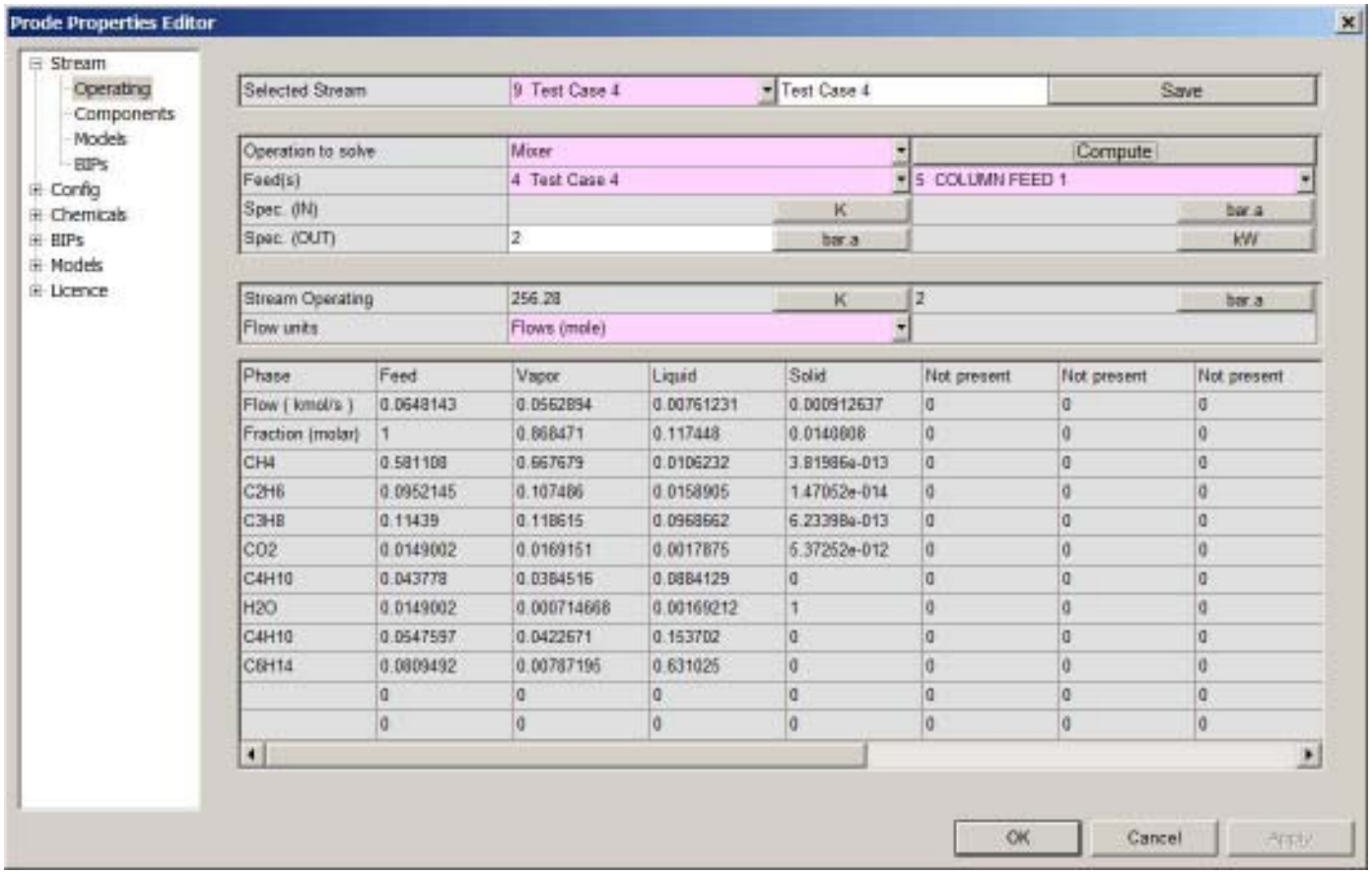
we can now repeat the previous example to obtain the properties in Excel with new operating conditions

B1 =getT(8)

	A	B	C	D
1	Temperature	220.8691		
2	Pressure	5		
3	Liquid Fraction	0.150289		
4	Liquid Density	570.6584		
5	Liquid heat Capacity	2.240764		
6	Gas Density	6.181223		
7	Gas Heat capacity	1.846163		

in cell D1 enter =getT(8) to obtain operating temperature for stream 1
in cell D2 enter =getP(8) to obtain operating pressure
in cell D3 enter =StrLf(8) to obtain liquid fraction
in cell D4 enter =StrLD(8) to obtain density of liquid fraction
in cell D5 enter =StrLcp(8) to obtain heat capacity of liquid fraction
in cell D6 enter =StrGD(8) to obtain density of vapor fraction
in cell D7 enter =StrGcp(8) to obtain heat capacity of vapor fraction

Following a similar procedure it is possible to solve, from Prode Properties editor operations as separators and mixers.
Differently from other operations, where the specifications define initial and final conditions, the mixer requires to specify the operating conditions for the two feeds before to solve the mixer operation, it is possible to define the operating conditions for the two feeds solving, for example, two isothermal flash operations.
Supposing we wish to add stream 4 to stream 5 and obtain the results (new composition and operating conditions) in stream 9, we can follow this procedure
1) select stream 4, define T-P VL operation with T = 300 P = 5 Bar.a click on Compute to solve
2) select stream 5, define T-P VL operation with T = 270 P = 2 Bar.a click on Compute to solve
3) select stream 9 (where the results will be stored) , define Mixer, select streams 4 and 5 as feeds and specify 2 Bar.a as output pressure, click on Compute to solve



as in previous example note that you can obtain equivalent results with macro

=MIXF(9,4,5,2,0)

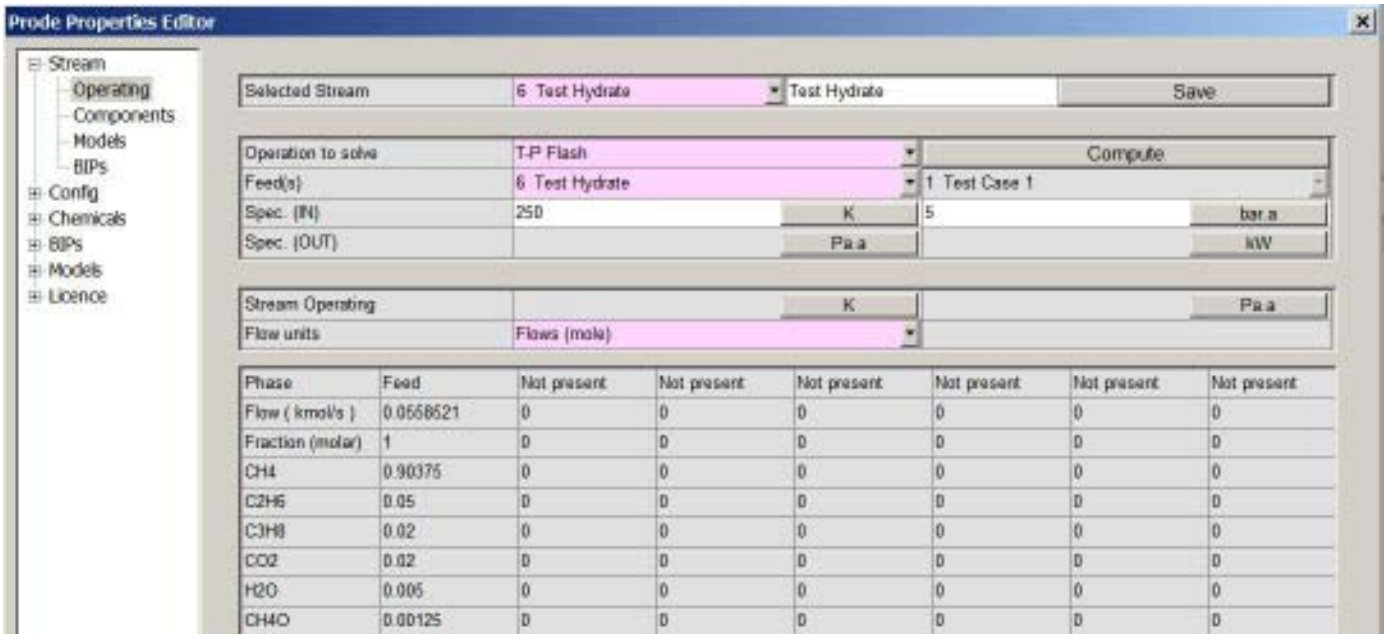
which solves a mixer operation, mixing streams 4 and 5 (results in stream 9) with specified ouput pressure 2 Bar.a
Also note that Mixer operation adopts the multiphase settings given for first stream (in this case stream 4) , if you invert sequence =MIXF(9,5,4,2,0) it shows a Vapor-liquid solution (no multiphase)

next example shows how to evaluate hydrate formation for a given mixture.

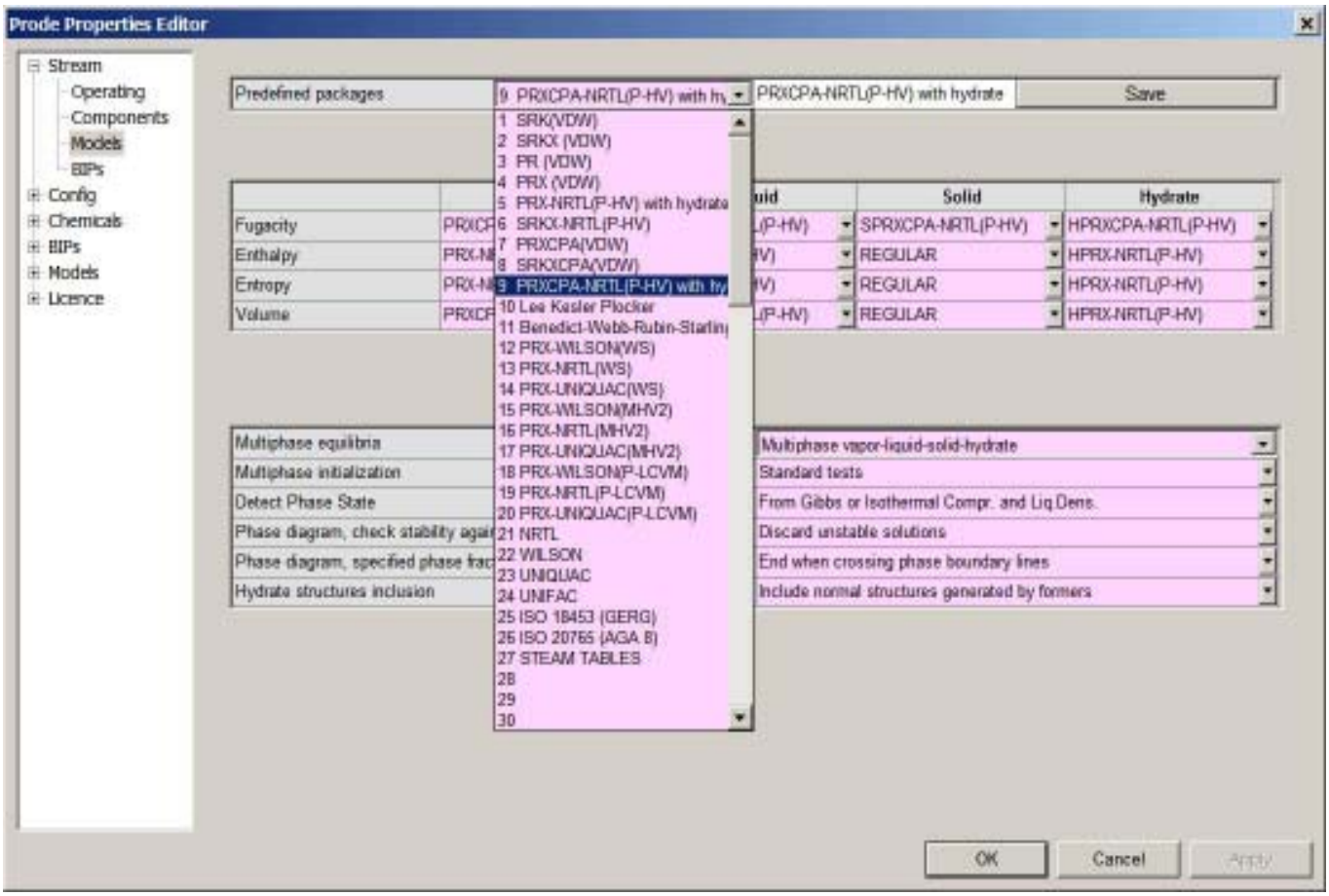
Important : Hydrate models have a limited range of temperatures and pressures
Standard Hydrate Method 220 K < T < 310 K 0.1 Bar.a < P < 1000 Bar.a
Complex Hydrate Method 220 K < T < 320 K 0.1 Bar.a < P < 3000 Bar.a

the different operations can be solved only when t, p conditions are within the allowed range, outside this range the methods can return a "cannot converge calc's loop" error meaning that procedure cannot calculate one (or more) point(s)

To calculate phase equilibria with hydrates you must include in stream one or more fomers plus water.
In Properties Editor select stream "6 Test Hydrate", this stream includes a predefined composition C1 0.90375 C2 0.05 C3 0.02 CO2 0.02 H2O 0.005 CH4O 0.00125

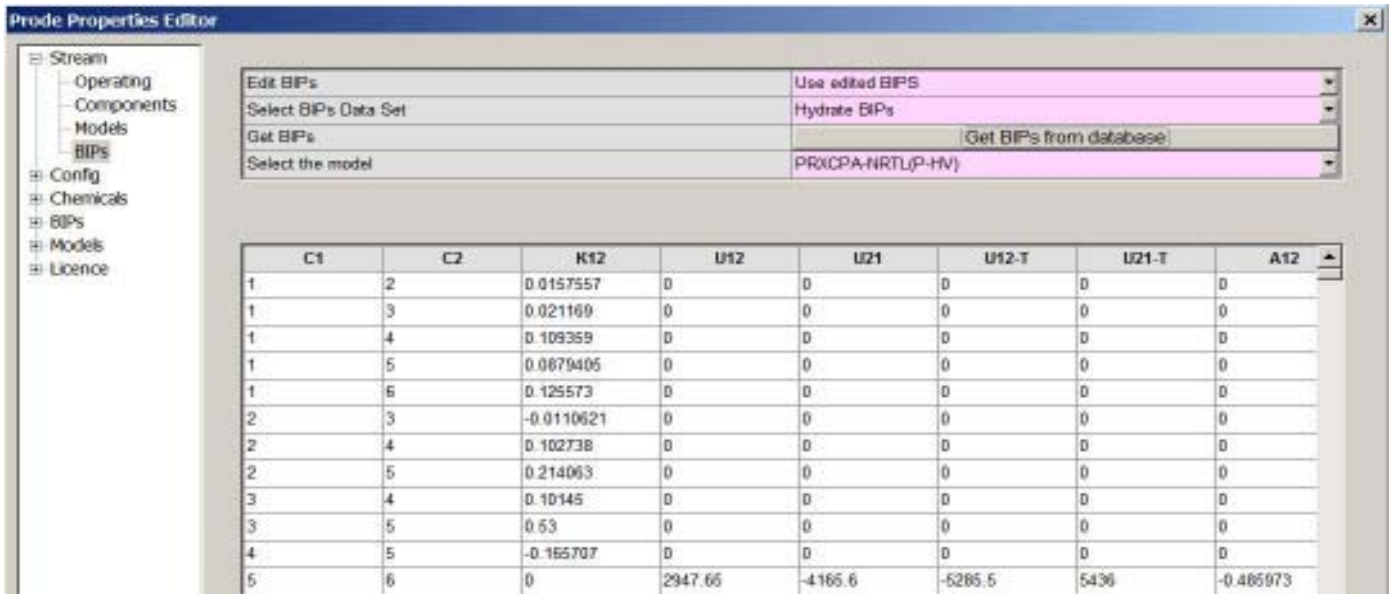


When solving phase equilibria with solids and hydrates make sure to select the correct models for vapor, liquid, solid and hydrate phases, For hydrates Prode Properties Base includes a model based on PRXCPA-NRTL(P-HV) and one based on PRX-NRTL(P-HV),
- PRXCPA-NRTL(P-HV) adopts PRXCPA-NRTL for vapor and liquid , SPRXCPA -NRTL for solid , HPRXCPA -NRTL for hydrate
- PRX-NRTL(P-HV) adopts PRX-NRTL for vapor and liquid , SPRX -NRTL for solid and HPRXCPA -NRTL for hydrate
in this example select the PRXCPA-NRTL(P-HV) package which automatically define the correct models



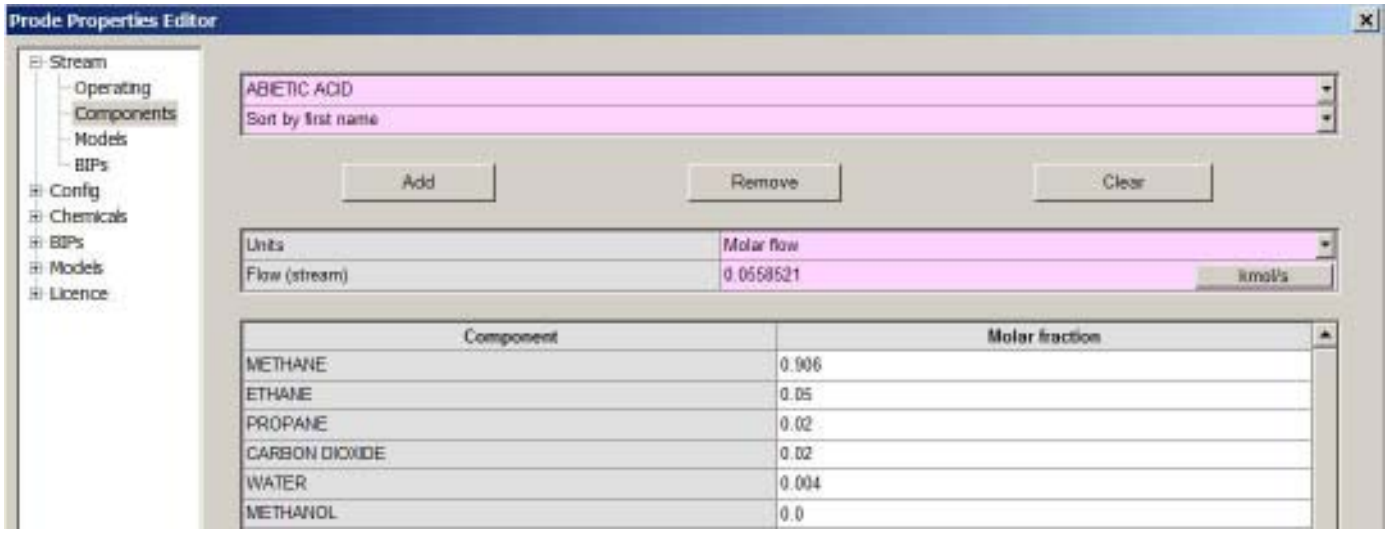
select as Multiphase equilibria option Multiphase Vapor-Liquid-Solid Hydrate

When calculating phase equilibria with solids (hydrates) to avoid large errors you must define BIPs,
for this example select Hydrate BIPs as Data Set and click on Get BIPs from Database button to load the values



C1	C2	K12	U12	U21	U12-T	U21-T	A12
1	2	0.0157557	0	0	0	0	0
1	3	0.021169	0	0	0	0	0
1	4	0.109359	0	0	0	0	0
1	5	0.0879405	0	0	0	0	0
1	6	0.125573	0	0	0	0	0
2	3	-0.0110621	0	0	0	0	0
2	4	0.102738	0	0	0	0	0
2	5	0.214063	0	0	0	0	0
3	4	0.10145	0	0	0	0	0
3	5	0.53	0	0	0	0	0
4	5	-0.165707	0	0	0	0	0
5	6	0	2947.65	-4165.6	-5285.5	5436	-0.485973

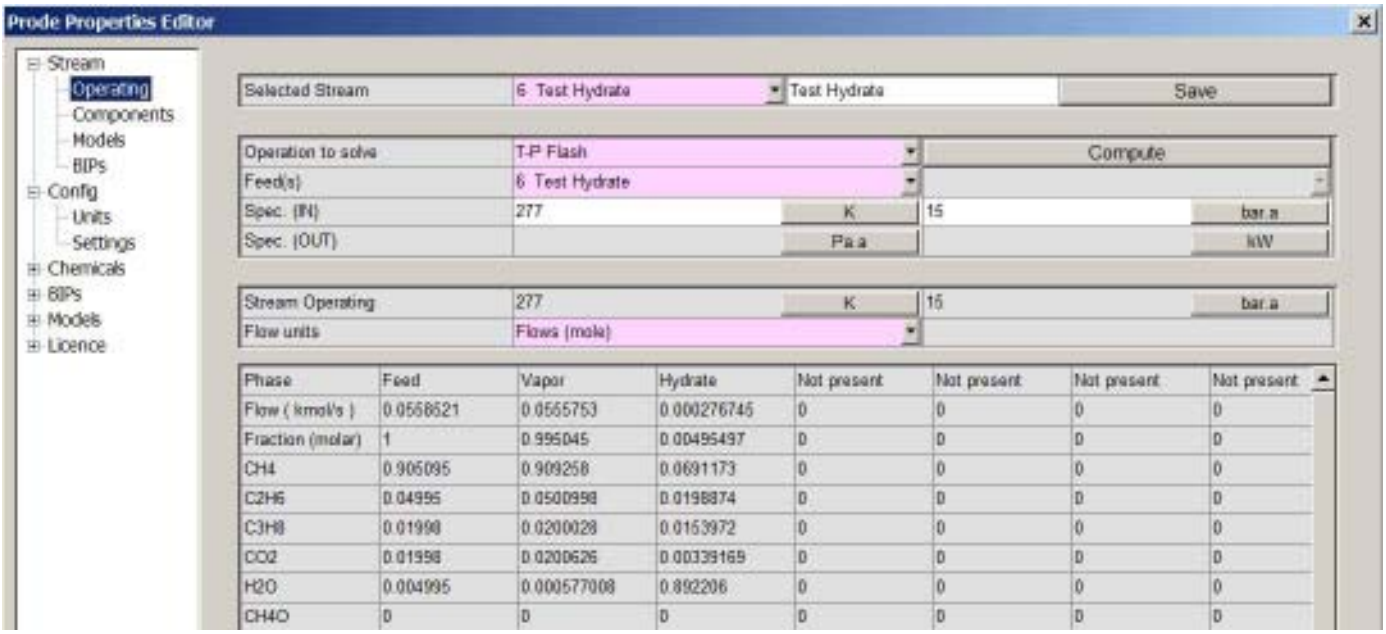
now, in Components tab define the composition C1 0.906 C2 0.05 C3 0.02 CO2 0.02 H2O 0.004 CH4O 0



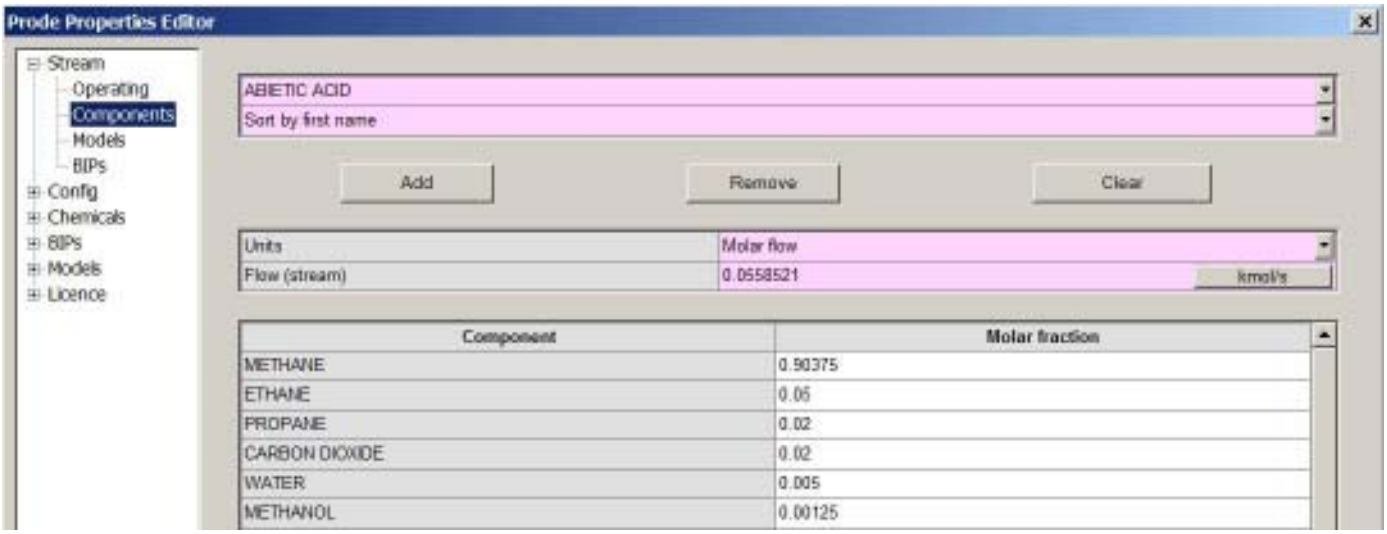
Component	Molar fraction
METHANE	0.906
ETHANE	0.05
PROPANE	0.02
CARBON DIOXIDE	0.02
WATER	0.004
METHANOL	0.0

then, back to Operating tab and click on Save button to store the values in Prode Properties

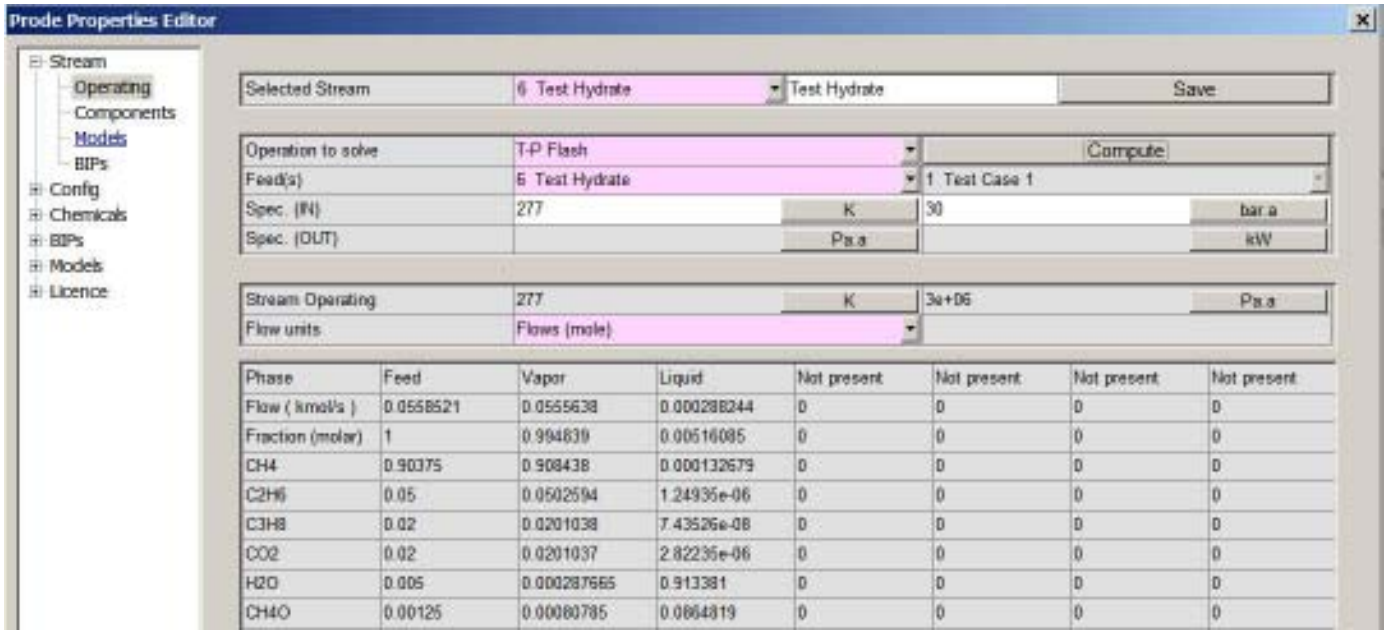
Once saved you can calculate hydrate phase equilibria immediately selecting the TP-Flash operation, setting temperature (277 K) and Pressure (15 Bar.a) , click on Compute button to see the results, at specified conditions the model indicates that hydrates can form



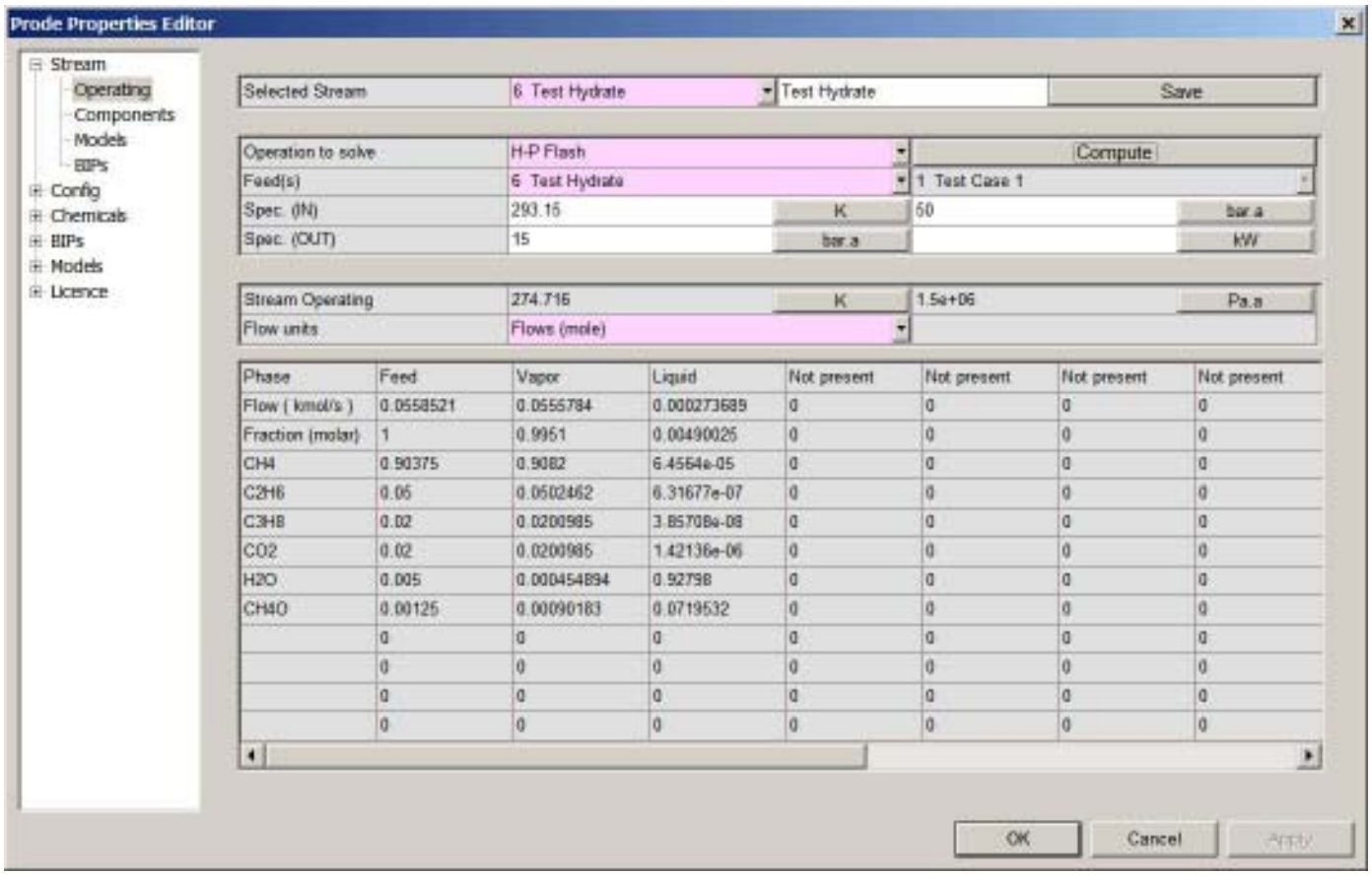
you may decide to adopt methanol as inhibitor to avoid the formation of hydrates
We will consider a methanol (molar) fraction of about 1/4 or 0.00125 methanol vs. 0.0050 water
In component's tab define the compositon C1 0.90375 C2 0.05 C3 0.02 CO2 0.02 H2O 0.005 CH4O 0.00125



in the Operating tab click on Save button to store the new composition
you can solve a TP-Flash operation to find the predicted hydrate formation pressure
in this case we test 277 K 30 Bar.a without finding hydrate formation



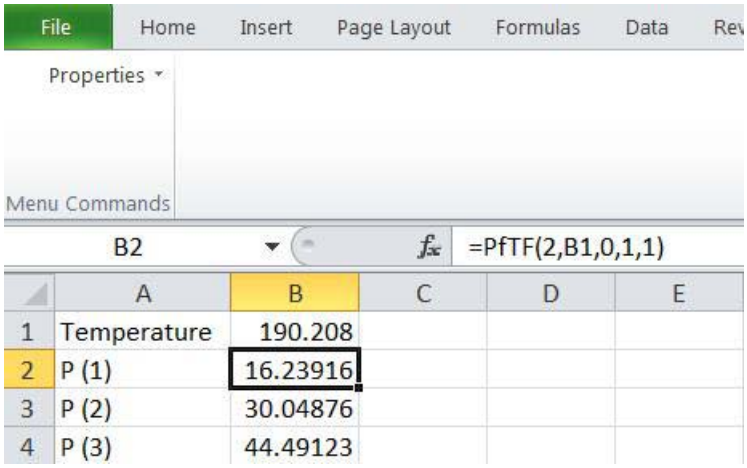
in a similar way it is possible to simulate a valve with the H-P Flash operation, in this case (on the same stream 6) we simulate a valve with Tin 293.15 K Pin 50 Bar.a and Pout 15 Bar.a , the procedure calculates as final temperature about 275 K (without hydrate phase)



Prode Properties includes several methods for calculating equilibrium points, see the paragraph “Methods for thermodynamic calc’ s” for additional information
LfPF() and LfTF() as the name says are based on a liquid fraction specification, they returns the first point (along the specified liquid fraction line) at the specified pressure (or temperature)
PfPF() and PfTF() instead can accept a gas, liquid or solid fraction and calculate up to 5 points (at specified pressure or temperature) along the equilibrium line,

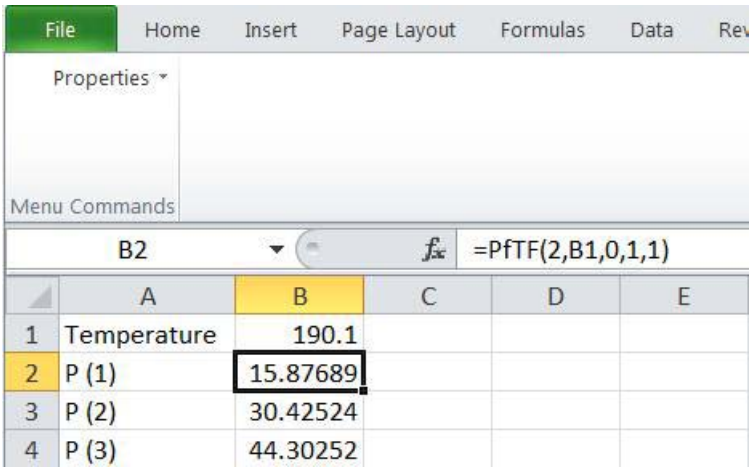
double p = PfTF(integer stream, double t, double pf, int state, int n)

which requires the stream, the equilibrium temperature, the phase fraction (range 0-1), the state (gas, liquid, solid) and the position (1-5) of the equilibrium point
In cell B1 we define the temperature as 190.208 K , then in cells B40 , B41, B42 we enter the macros
=PfTF(2,B1,0,1,1) in cell B2
=PfTF(2,B1,0,1,2) in cell B3
=PfTF(2,B1,0,1,3) in cell B4
where the first value (2) is the stream , the second (cell B1) represents the equilibrium temperature, the third (1) is the phase fraction (with 1 we specify 100% gas or a point on dew line, the same would be by setting the state as liquid and phase fraction as 0.0) the fourth (0) is the state (in Properties 0 = gas, 1 = liquid, 2 = solid) and the last is the required position (we require the points 1-3 along the dew line)



	A	B	C	D	E
1	Temperature	190.208			
2	P (1)	16.23916			
3	P (2)	30.04876			
4	P (3)	44.49123			

the procedure calculates the three equilibrium points (in this case units for pressure is Bar.a), if we change the temperature to 190.1 K we get different equilibrium pressures:



	A	B	C	D	E
1	Temperature	190.1			
2	P (1)	15.87689			
3	P (2)	30.42524			
4	P (3)	44.30252			

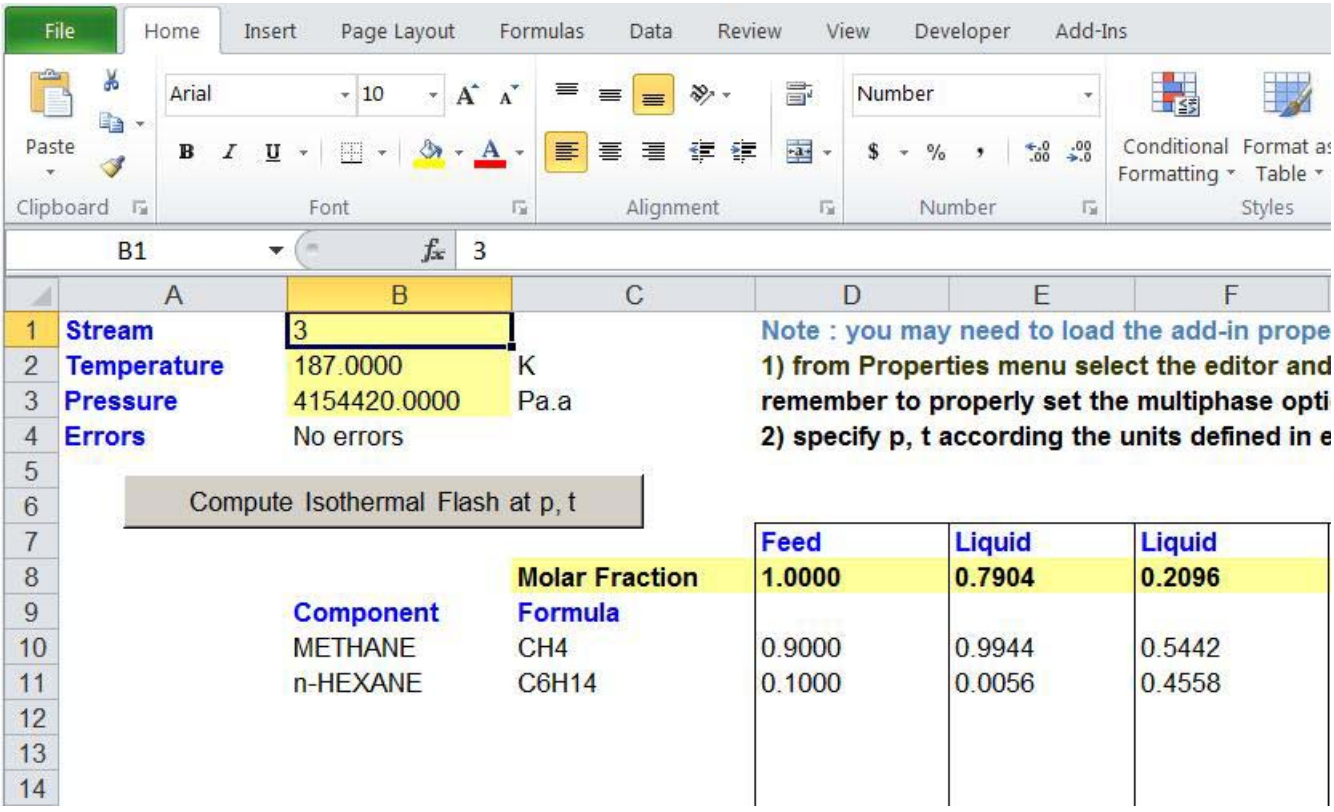
Getting Started from Microsoft Excel, part 3 working with predefined pages

IMPORTANT : do not enter Prode macros in Excel cells of predefined pages, instead, open a new blank workbook as discussed in “Getting Started from Microsoft Excel, part 2”

Check the units, the input values in predefined pages are in K (temperature) and Pa (pressure) but you may change the units, open Prode Properties editor and set K for temperature and Pa for pressure or convert the input values to your preferred units

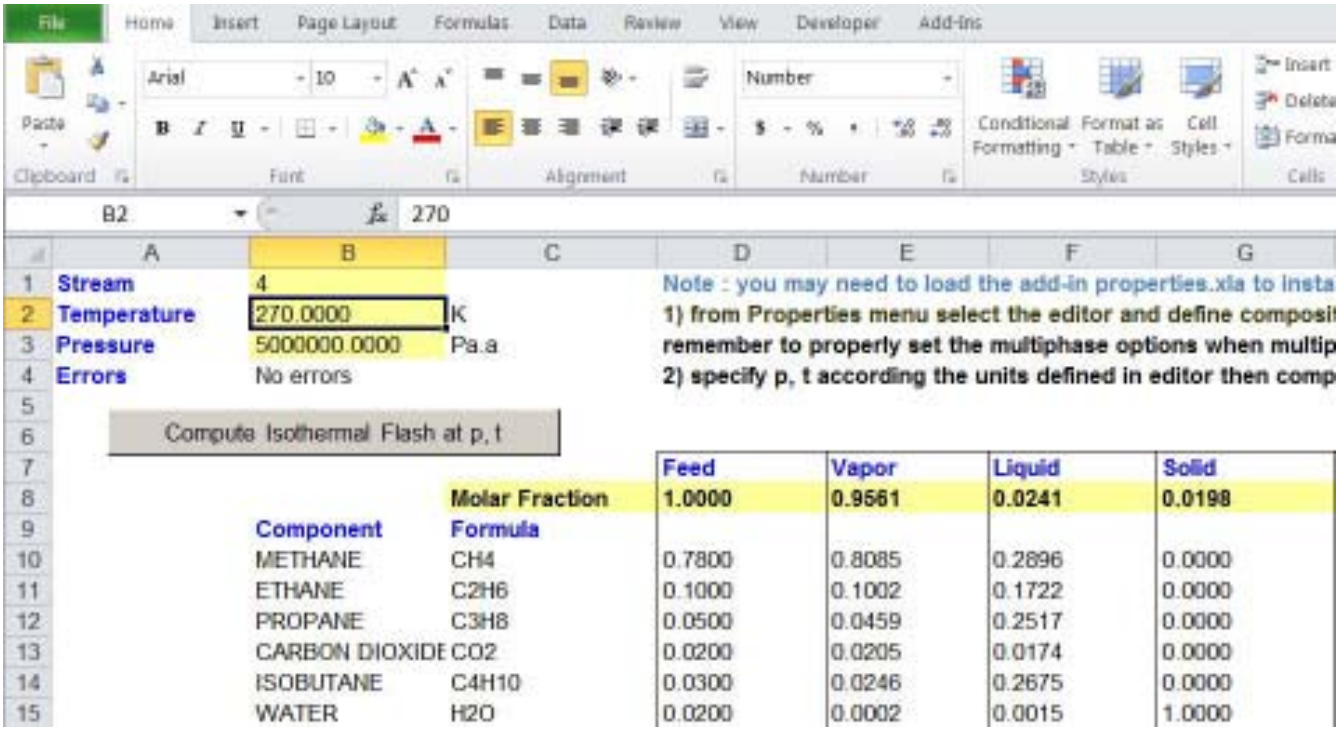
Open the page multiphase.xls, (available in folder /prode/excel) this page allows to solve a multiphase equilibria problem and see the results in Excel.

Define stream 3 as feed, temperature 187 K and pressure 4154420 Pa.a (40 atm.g)



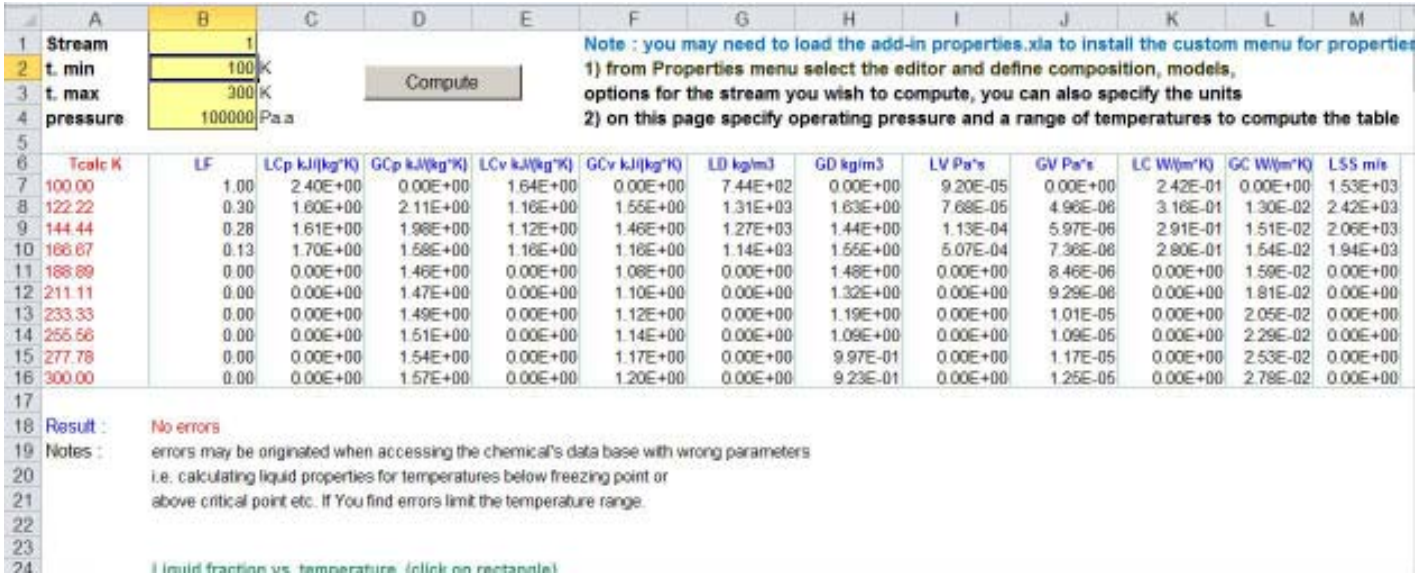
the procedure solves a isothermal flash showing the formation of two liquid phases.

In next example define stream 4 as feed, temperature 270 K and pressure 5000000 Pa.a (50 Bar.a) if required (different units selected) convert values.



The results show the presence of a vapor phase, a liquid phase (mainly hydrocarbons) and a solid phase (ice)

The page props.xls allows to calculate and graph tables of values in a range of temperatures for many different properties (liquid fraction, cp, cv, density, viscosity, thermal conductivity, speed of sound) for both gas and liquid phases (if present).
From Excel menu File->open , in Excel folder (in Prode Properties installation) select the file props.xls
If you wish you can modify the stream composition or the units of measurement (see previous examples).
In this page enter (in the proper units) the desired range of temperatures (cells B2-B3) and the operating pressure (cell B4) and click on compute button to calculate the data, Prode Properties will print the values with the desired units of measurement.



	A	B	C
1	Stream	1	
2	t. min	100	K
3	t. max	300	K
4	pressure	100000	Pa.a

Compute

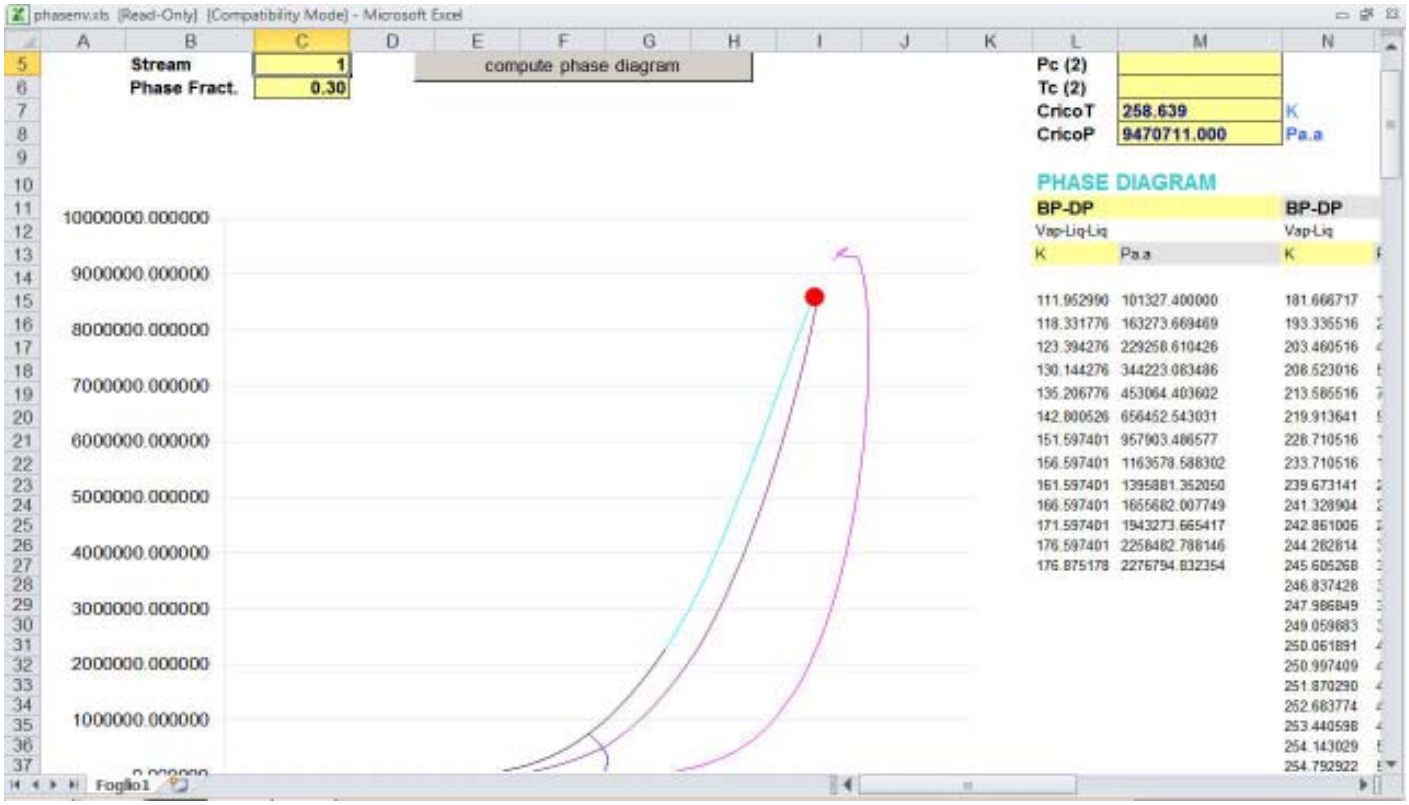
Note : you may need to load the add-in properties.xla to install the custom menu for properties
1) from Properties menu select the editor and define composition, models,
options for the stream you wish to compute, you can also specify the units
2) on this page specify operating pressure and a range of temperatures to compute the table

Tcalc K	LF	LCp kJ/(kg*K)	GCp kJ/(kg*K)	LCv kJ/(kg*K)	GCv kJ/(kg*K)	LD kg/m3	GD kg/m3	LV Pa*s	GV Pa*s	LC W/(m*K)	GC W/(m*K)	LSS m/s
100.00	1.00	2.40E+00	0.00E+00	1.64E+00	0.00E+00	7.44E+02	0.00E+00	9.20E-05	0.00E+00	2.42E-01	0.00E+00	1.53E+03
122.22	0.30	1.60E+00	2.11E+00	1.16E+00	1.55E+00	1.31E+03	1.63E+00	7.68E-05	4.96E-06	3.16E-01	1.30E-02	2.42E+03
144.44	0.28	1.61E+00	1.98E+00	1.12E+00	1.46E+00	1.27E+03	1.44E+00	1.13E-04	5.97E-06	2.91E-01	1.51E-02	2.06E+03
166.67	0.13	1.70E+00	1.58E+00	1.16E+00	1.16E+00	1.14E+03	1.55E+00	5.07E-04	7.36E-06	2.80E-01	1.54E-02	1.94E+03
188.89	0.00	0.00E+00	1.46E+00	0.00E+00	1.08E+00	0.00E+00	1.48E+00	0.00E+00	8.46E-06	0.00E+00	1.59E-02	0.00E+00
211.11	0.00	0.00E+00	1.47E+00	0.00E+00	1.10E+00	0.00E+00	1.32E+00	0.00E+00	9.29E-06	0.00E+00	1.81E-02	0.00E+00
233.33	0.00	0.00E+00	1.49E+00	0.00E+00	1.12E+00	0.00E+00	1.19E+00	0.00E+00	1.01E-05	0.00E+00	2.05E-02	0.00E+00
255.56	0.00	0.00E+00	1.51E+00	0.00E+00	1.14E+00	0.00E+00	1.09E+00	0.00E+00	1.09E-05	0.00E+00	2.29E-02	0.00E+00
277.78	0.00	0.00E+00	1.54E+00	0.00E+00	1.17E+00	0.00E+00	9.97E-01	0.00E+00	1.17E-05	0.00E+00	2.53E-02	0.00E+00
300.00	0.00	0.00E+00	1.57E+00	0.00E+00	1.20E+00	0.00E+00	9.23E-01	0.00E+00	1.25E-05	0.00E+00	2.78E-02	0.00E+00

Result : No errors
Notes : errors may be originated when accessing the chemical's data base with wrong parameters
i.e. calculating liquid properties for temperatures below freezing point or
above critical point etc. If You find errors limit the temperature range.

Liquid fraction vs. temperature (click on rectangle)

The page phasenv.xls allows to calculate and graph a phase diagram (phase envelope)
from Excel menu File->open , in Excel folder (in Prode Properties installation) select the file phasenv.xls
As for previous examples you can modify the stream composition, the units of measurement etc. from Prode Properties editor.



this phase diagram shows a three phase area for the mixture CH4 CO2 H2S (0.7 0.15 0.15)

To obtain different diagrams change the settings in models dialog

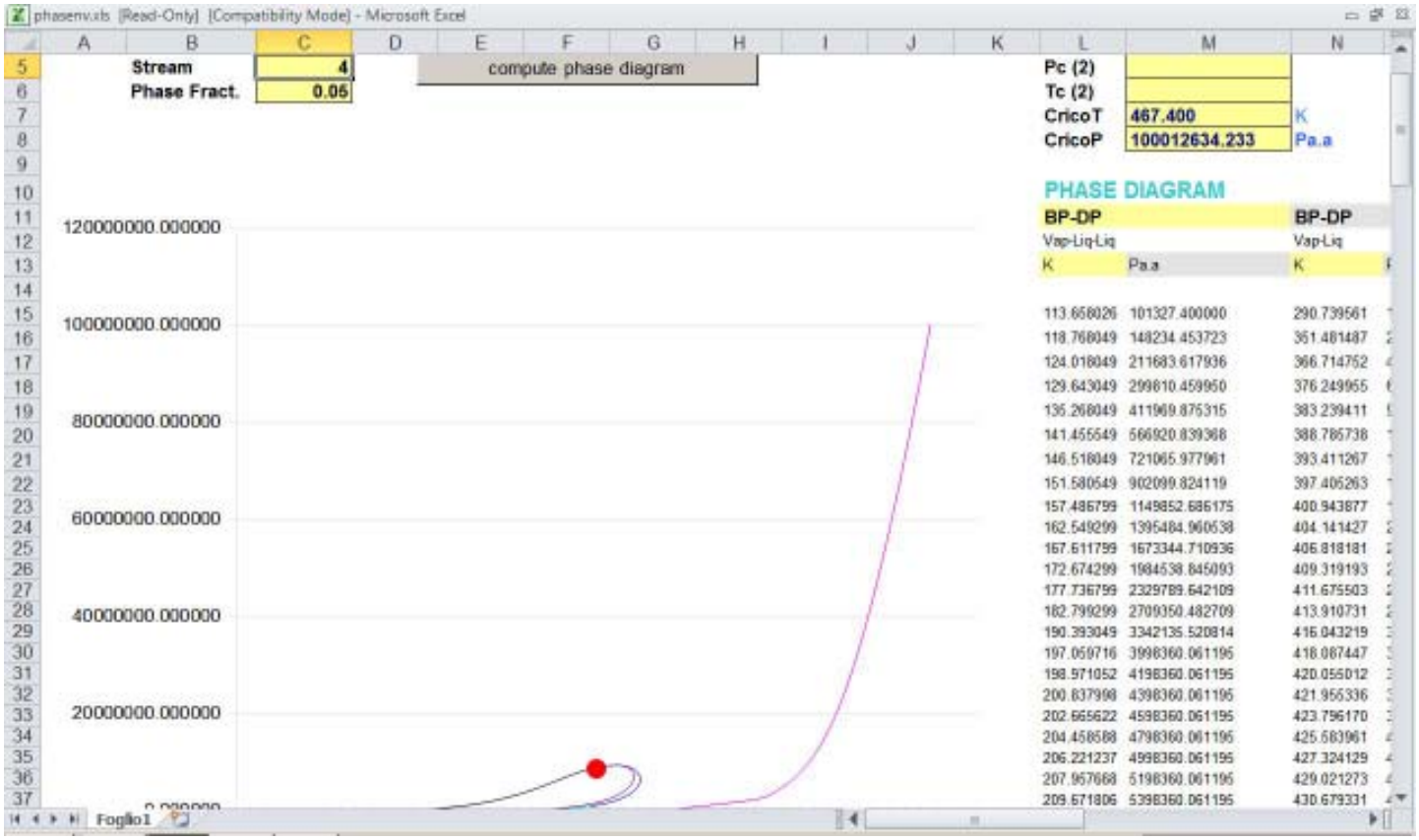
Multiphase equilibria	Multiphase vapor-liquid
Multiphase initialization	No multiphase, only two-phases Multiphase vapor-liquid Multiphase vapor-liquid-solid Multiphase vapor-liquid-solid-hydrate
Detect Phase State	
Phase diagram, check stability against feed	
Phase diagram, specified phase fraction lines	End when crossing phase boundary lines
Hydrate structures inclusion	Include normal structures generated by formers

Multiphase equilibria option allows to calculate
1) vapor-liquid phase diagrams
2) vapor-liquid-liquid phase diagrams
3) vapor-liquid-solid phase diagrams

Phase diagram, check stability against feed, permits to test stability of calculated points against feed, unstable points are not printed, to show all calculated points change the settings.

Phase diagram, specified phase fraction lines, allows to end (or continue) lines after crossing a phase boundary, set to end (when crossing phase boundary lines) to avoid generating lines containing inconsistent data.

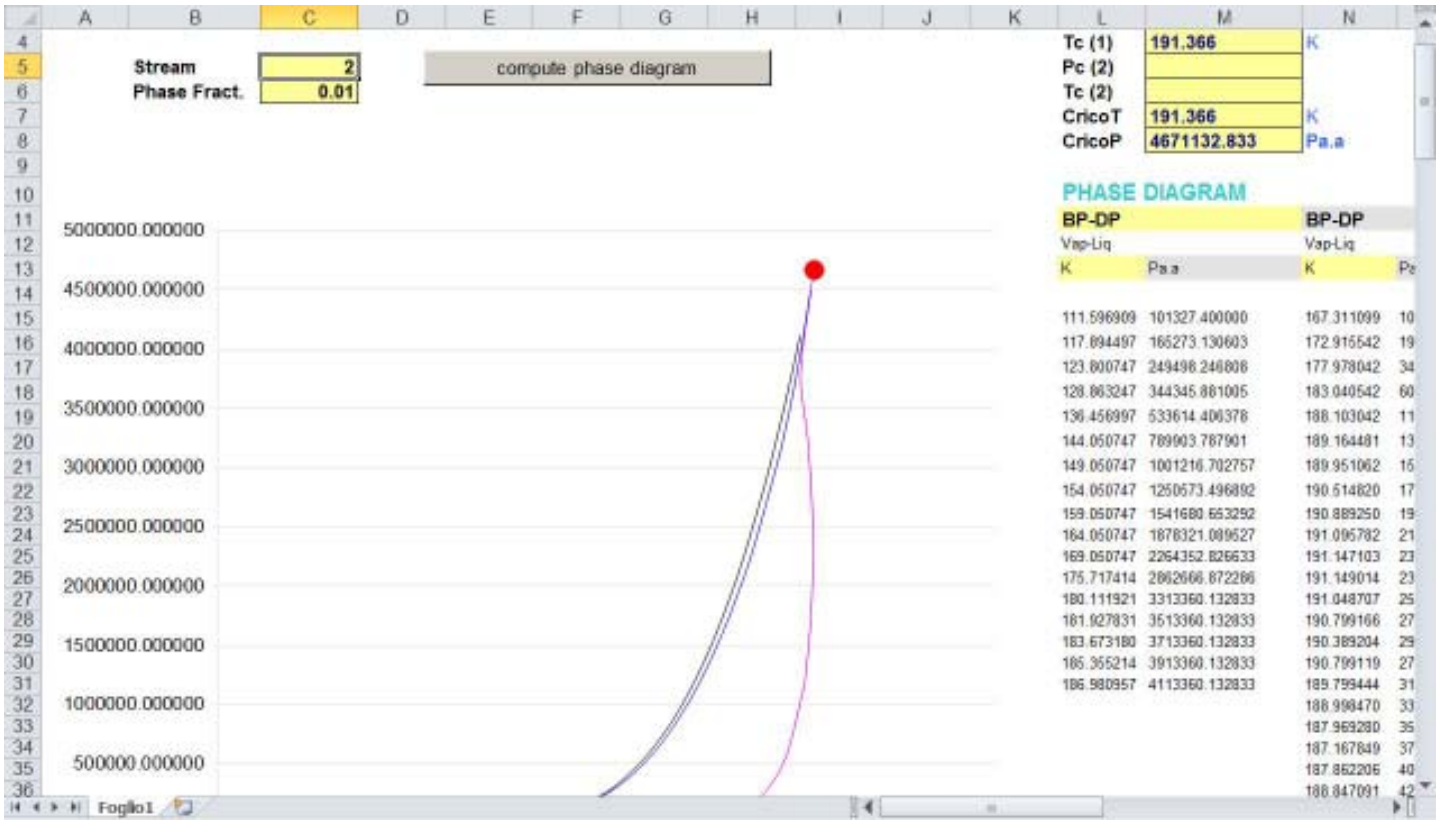
Next example will show a vapor-liquid-liquid phase diagram
a) enter stream 4, a predefined test case with a natural gas mixture including water
b) enter 0.0.5 as Phase Fraction and click on compute button to calculate phase diagram



Notice the water dew point line, the red line on the right

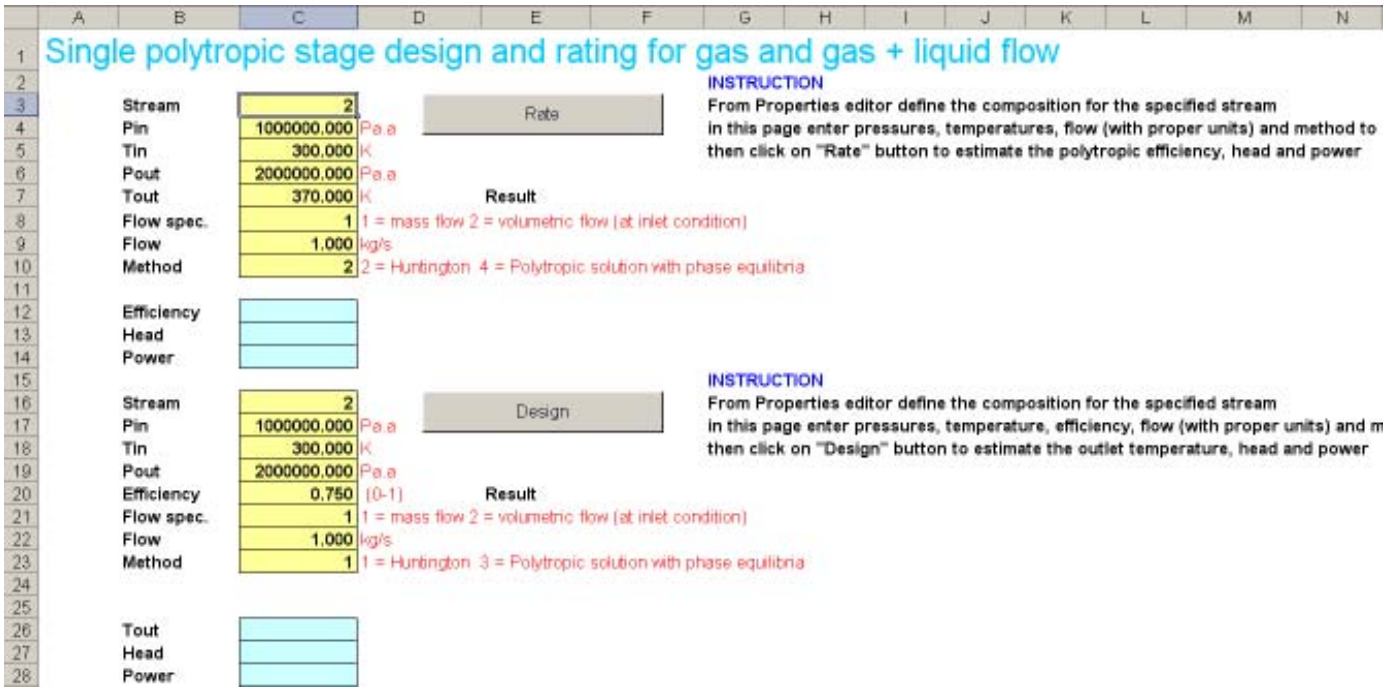
Next example will show a phase diagram with up to three dew points at the same temperature,

enter 2 as stream, 0.01 as liquid fraction then click on button “Compute phase diagram” to generate the graph

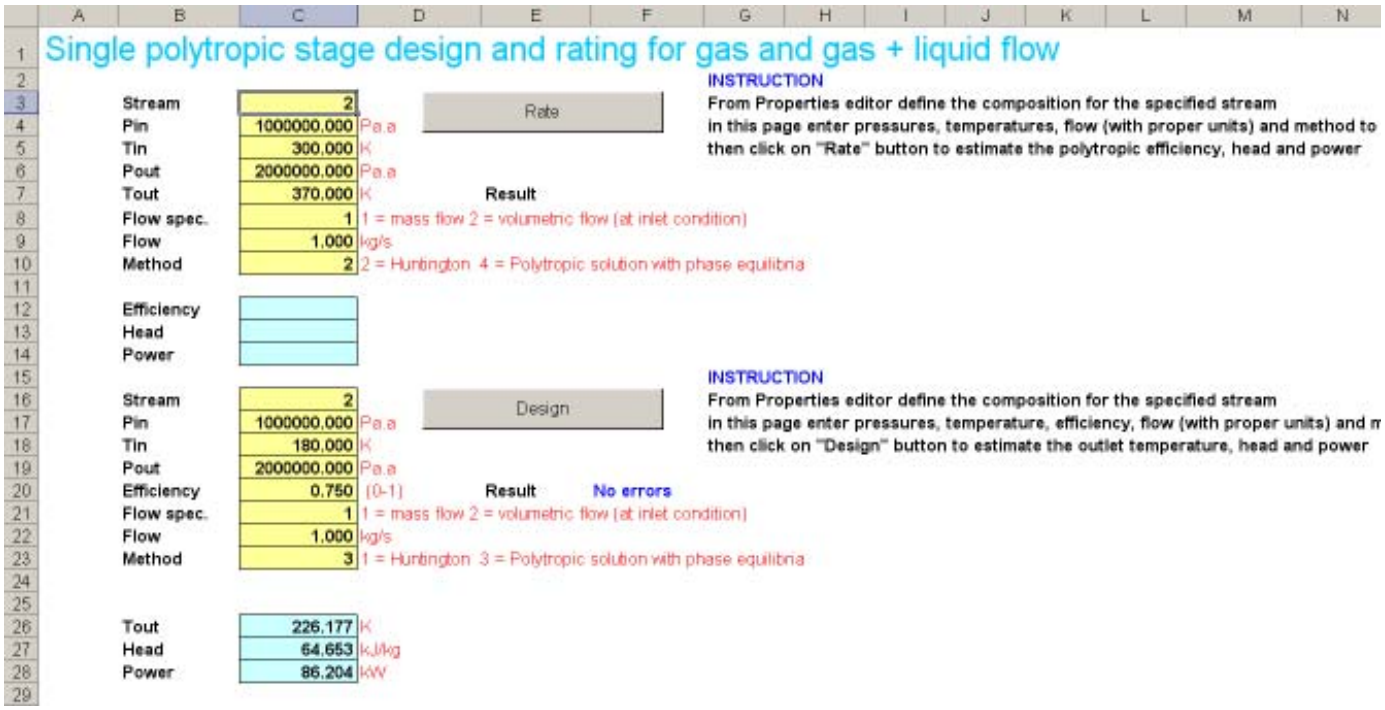


Observe that for this mixture the dew line (red line) shows up to three different equilibrium points at the same temperature (the area around 190 K), if you add the saturation point on the bubble line (black line) we have a total of four saturation point pressures at a given temperature, we will show how to calculate these points in Excel

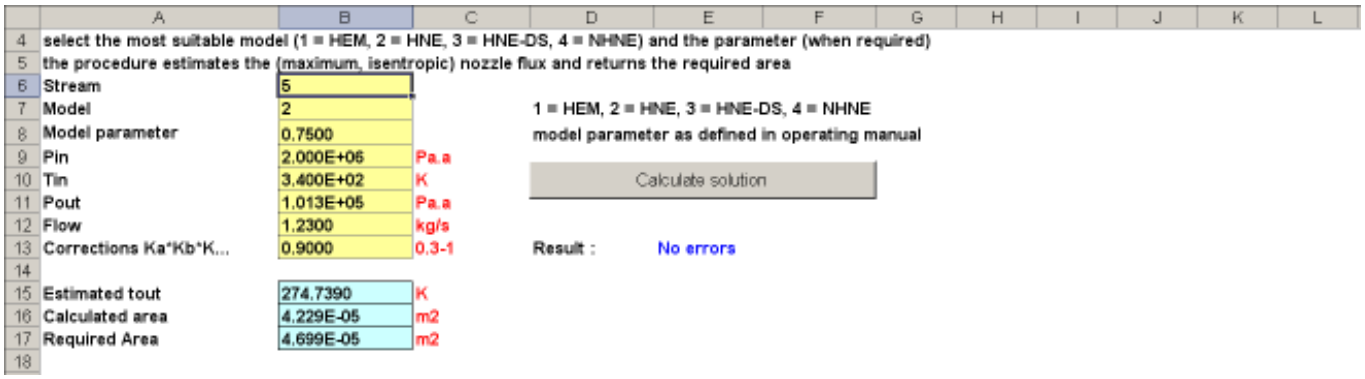
The page compressor.xls shows how to simulate a compression stage (as polytropic process) where the inlet stream can be vapor or vapor + liquid (mixed), comparing the results of different methods, see the paragraph "Methods for solving a Polytropic operation".for additional information.
From Excel menu File->open , in Excel folder (in Prode Properties installation) select the file compressor.xls



the page contains two sections, the first permits to calculate the polytropic efficiency of a single compression stage given the inlet temperature and pressure.
The second section allows to estimate the discharging temperature given inlet temperature and pressure, outlet pressure and polytropic efficiency.
Notice that Prode Properties includes a specific methof for solving a polytropic stage with phase equilibria, this method permits to simulate both single phase (vapor) and mixed (vapor + liquid) processes.
The mixture Methane 0.999, n-Butane 0.001 (predefined stream 2) at 10 Bar.a shows a dew point of 187.5 K , by setting a inlet temperature of 180 K we specify vapor + liquid as inlet condition, the standard method can simulate only gas streams, however the Polytropic solution with phase equilibria method allows to solve this case.



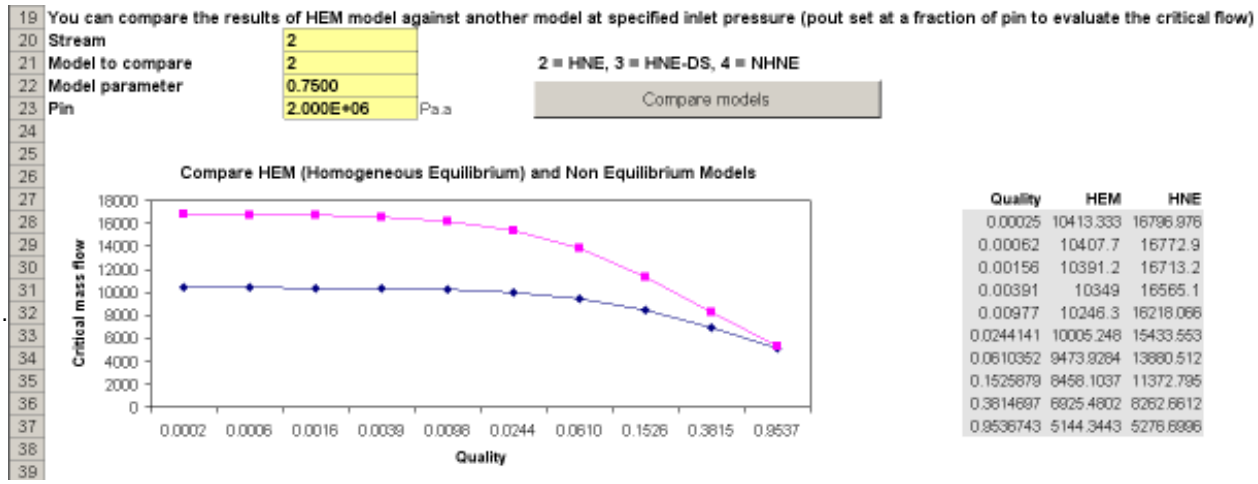
The page nozzle.xls allows to size a relief valve comparing the results of different methods for critical and two-phase flow, see the paragraph “Methods for solving a Isentropic operation” for additional information.
From Excel menu File->open , in Excel folder (in Prode Properties installation) select the file nozzle.xls



The steps to size a relief valve are easy to follow:
1) from Properties editor define the composition, models, BIPs (for mixtures)
2) enter the discharging temperature, pressure, flow, model, outlet pressure
3) click on button “Calculate Solution”

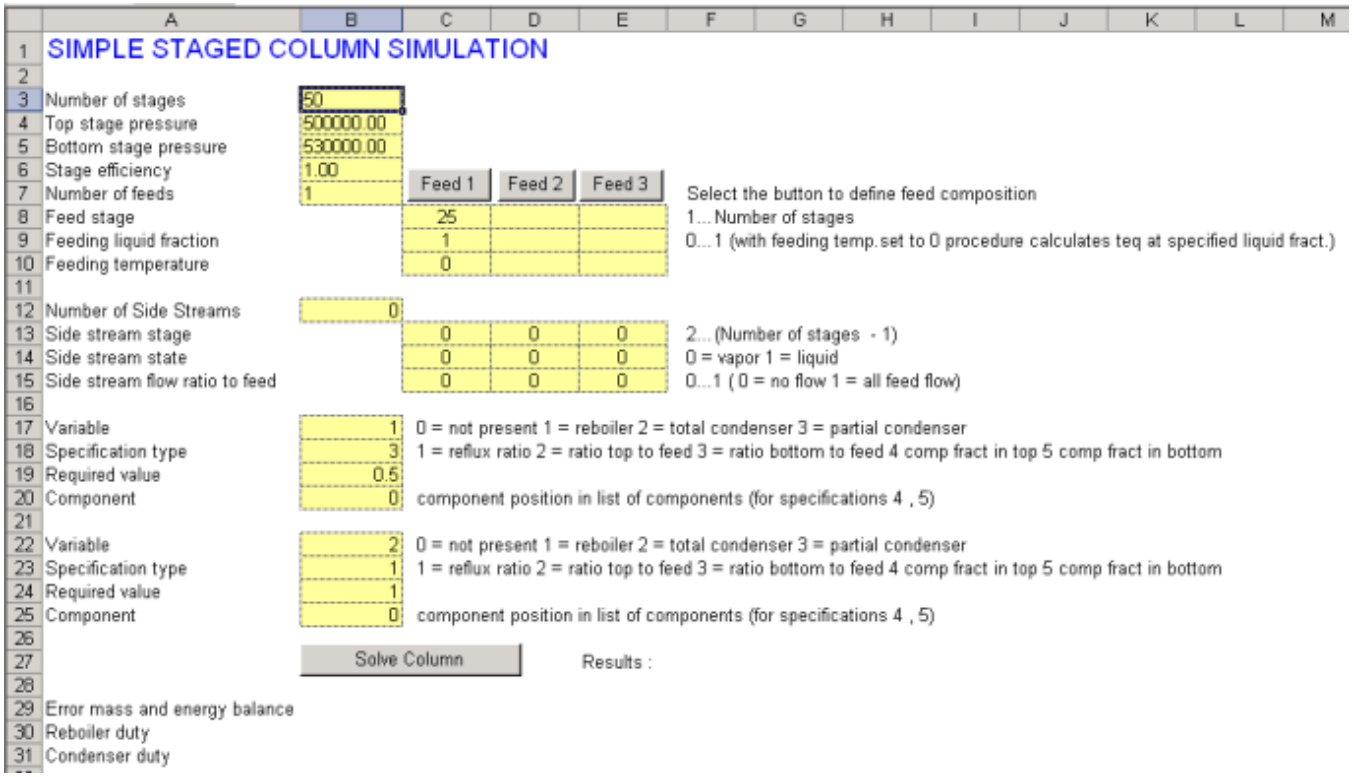
the procedure calculates the required area and the outlet temperature for critical and two-phase flow,
you may utilize the procedure to verify the results from a different software in applications as fluids in critical area, two-phases flow etc.

The same page includes a procedure to compare the results from HEM (Homogeneous Equilibrium) and different Non Equilibrium models for a specified pressure in a range of inlet vapor qualities
Please follow fhese steps to compare:two models,
1) from Properties editor define the composition, models, BIPs (for mixtures)
2) enter the pressure, model and parameter
3) click on button “Compare Models”



The Non Equlibrium models are mainly of interest for short nozzles where the final equilibrium condition (predicted by HEM models) is not reached cause the residence time of the fluid is too short.
The HNE models require specific parameters, for Prode HNE model a value of 0.75 is suggested for short nozzles but different values may be defined to fit specific data sets.

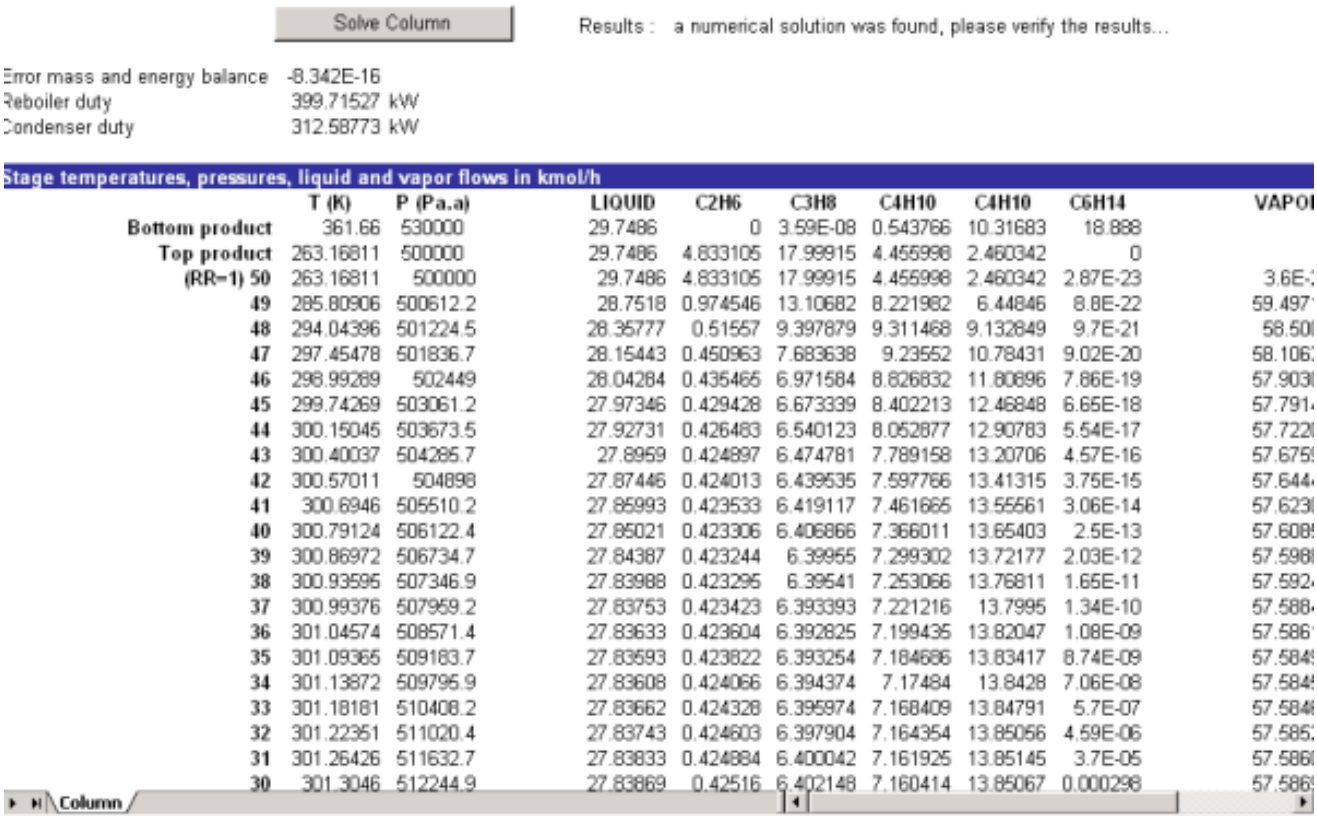
The page column.xls permits to solve a distillation column, refer to paragraph “Methods for solving staged columns” for additional information.
From Excel menu File->open , in Excel folder (in Prode Properties installation) select the file column.xls
In this page you can define different kind of columns with reboiler, condenser , one or more feeds and one or more side streams.



The steps to define a column are easy to follow:
1) define the number of stages
2) define pressure distribution (bottom and top stage)
3) define stage efficiency
4) define the number of feeds, each feed flow rate and compositions (click on the proper Feed button to access the stream editor), each feed stage (remember that reboiler (if present) is stage 1 and condenser (if present) is stage N, and the liquid fraction (or the temperature) of each feed.
5) Define the number of side streams (if any) , the stage, the type (vapor or liquid flow) and the flow specification
6) Define variables as condenser and reboiler and the related specifications, the procedure allows different specifications including molar fractions (and recovery) of a component in top or bottom stage

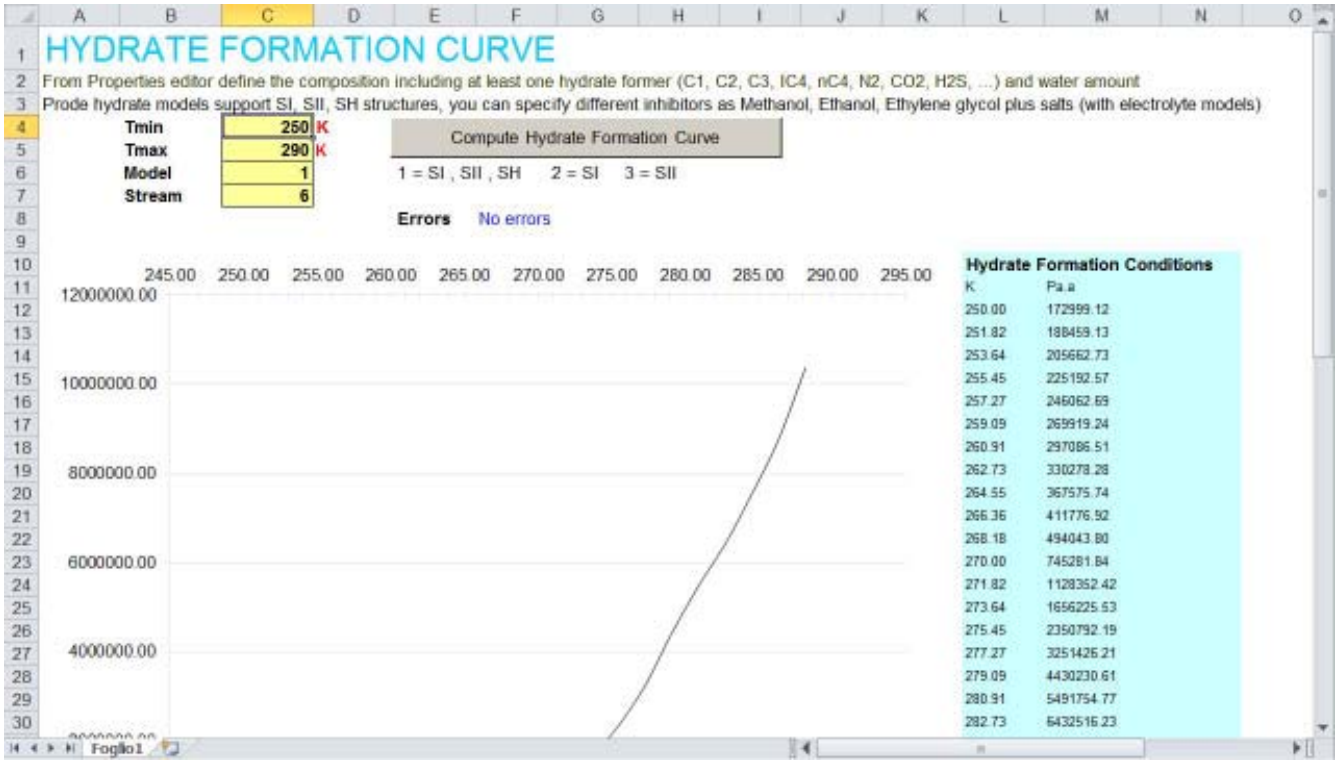
Notes :
In Stream Editor (Config->Units) you can define all the units for this project
in Stream Editor (Config->Setti gs) you can define mass units or molar units for flows in Stream Editor

Once the column has been defined it is suggested to verify the input data for inconsistent specifications, if you are sure that all is Ok run the solver (button Solve Column)



the report includes
1) the verified errors in mass and energy balance
2) reboiler and condenser duties
3) temperature and pressure in each stage
4) total and component vapor flows in each stage
5) total and component liquid flows in each stage

The page hydrate.xls allows to calculate the hydrate formation curve directly in Excel.
From Excel menu File->open , in Excel folder (in Prode Properties installation) select the file hydrate.xls



note that base version of Prode Properties allows to include only a few inhibitors (methanol, ethanol, MEG) with extended versions you can include salts and additional inhibitors

Getting started from MATLAB

IMPORTANT Microsoft MATLAB support files are located in the directory \Prode\MATLAB

MATLAB provides two ways to access external libraries as Prode Properties
-direct access
-access through scripts and mex files

Direct Access

Direct access is through the command-line interface, this interface lets you load an external library into MATLAB memory and access functions in the library, to load Prode Properties in MATLAB enter

```
>if not(libisloaded('ppp'))
  hfile = ['C:\Program Files\Prode\MATLAB\ppp.h'];
  loadlibrary('ppp.dll', hfile);
end
libfunctions ppp
```

this command will load Prode Properties in memory and print the list of methods avaliable, you may wish to modify 'C:\Program Files\Prode\MATLAB\ppp.h' to reflect your installation's settings

```
Functions in library ppp:

AFOpen      CompLD      Divi        EStrLSS     PSep        StrGIC      StrLV       StrST       getAji      getPatm     putMod
AFSave      CompLV      EStrFMH     EStrLV      PfPF        StrGICp     StrLVE      StrSv       getCC       getSUMS     putN
AOpen       CompMp      EStrFML     EStrLVE     PfTF        StrGJT      StrLf       StrSvd      getCNr      getT        putZ
ASave       CompMw      EStrGC      EStrLf      SPF         StrGMw      StrMDt      StrTc       getCi       getUMC      setAc
BFsave      CompN       EStrGCp     EStrPf      STF         StrGS       StrMw       StrTcm      getCj       getUMN      setErrFlag
BPF         CompNb      EStrGCv     EStrS       StrAc       StrGSS      StrN        StrVv       getErrFlag  getUMS      setKM
BPLine      CompPc      EStrGD      EStrSCp     StrCBp      StrGV       StrPc       StrVvd      getFCNr     getW        setMFg
BRegr       CompRg      EStrGIC     EStrSD      StrCBt      StrGVE      StrPcm      StrZv       getFPNr     getWm       setMH
CompAc      CompSC      EStrGJT     EStrST      StrCPnr     StrH        StrPf       StrlnFv     getGij      getX        setMS
CompCID     CompSD      EStrGMw     EStrZv      StrCTp      StrHC       StrPt       StrlnFvd    getGji      getY        setMV
CompDm      CompSG      EStrGSS     ErrMsg      StrCTt      StrHv       StrPts      StrlnFvdv   getKji      getZ        setMw
CompF       CompSL      EStrGV      GSep        StrCopy     StrHvd      StrS        UMAU        getMBPNr    initS       setOM
CompGC      CompSS      EStrGVE     HPF         StrFMH      StrLC       StrSCp      UMCR        getMCNr     isSDef      setOp
CompGV      CompST      EStrH       HPFORM      StrFML      StrLCp      StrSD       UMCS        getMFg      loadSB      setPc
CompGf      CompSf      EStrHC      HTF         StrFv       StrLCv      StrSGH      UMRAU       getMH       putAji      setS
CompHG      CompSol     EStrLC      HTFORM      StrFvd      StrLD       StrSGS      VLLSep      getMS       putBIP      setSOp
CompHL      CompTc      EStrLCp     LFLine      StrFvdv     StrLH       StrSH       defErrMsg   getMSNr     putCC       setTc
CompHS      CompVP      EStrLCv     LSep        StrGC       StrLIC      StrSLH      edCF        getMV       putCi       setUMC
CompHV      CompVc      EStrLD      LfPF        StrGCp      StrLJT      StrSLS      edCS        getMod      putCj       setVc
CompHf      DCOL        EStrLIC     LfTF        StrGCv      StrLMw      StrSS       edS         getOM       putGij      setWm
CompID      DPF         EStrLJT     MixF        StrGD       StrLS       StrSSH      edSS        getP        putGji
CompLC      DPLine      EStrLMw     PIPE        StrGH       StrLSS      StrSSS      edST        getPNr      putKji
```

to access a method in a shared library MATLAB provides the command calllib to call functions in the library, the syntax for calllib is:

calllib('ppp', 'FunctionName', arg1, ..., argN)

the FunctionName and arguments are detailed in Prode Properties manual, for example we can call the method edSS() to edit streams with the command

>***calllib('ppp', 'edSS')***

in the same way you can access other methods in Prode Properties, for example to calculate cp / cv and speed of sound for vapor fraction of stream 1 at 300 K and 5 Bar

```
>> calllib('ppp','EstrGCp'1,300,500000)/'calllib('ppp','EstrGCv'1,300,500000)
>> ans = 1.3211
>> calllib('ppp','EstrGSS'1,300,500000)
>> ans = 374.1625
```

you can call even complex functions as those to plot a phase envelope or calculate a column, for these remember before to pass an array from Matlab to Prode Properties that you must allocate the memory to avoid system errors. Finally you can use the unloadlibrary function to unload Prode Properties library from Matlab and free up memory.

```
>unloadlibrary ppp
```

Access from Matlab through scripts

In addition to direct access, you can utilize Prode Properties from Matlab with scripts or mex files (compiled scripts)
In many cases this way is more immediate since you use the original names of the functions in Prode Properties without need to write additional code.
Prode Properties includes a large number of Matlab scripts installed in directory \Prode\MATLAB\m
Before to utilize the scripts you must
-move the files into a Matlab directory (i.e. a directory where Matlab can access the scripts) , read Matlab documentation for additional information.
-edit the file pppdir.txt, this file contains a string with path and name of the header file required to instruct Matlab about the methods avalialable in Prode Properties library, once you have edited move the file on the same location of script files.

How the scripts work
Scripts act as interface between Matlab and Prode Properties, scripts have names identical to Prode Properties methods, then when you invoke the script StrGD (which is the method in Prode Properties to calculate density of vapor phase) MATLAB simply executes the commands found in the file, calls the method StrGD in Prode Properties and returns the result, by the way the script StrGD.m contains these MATLAB commands

```
function [] = StrGD(stream)
if not(libisloaded('ppp'))
fid = fopen('pppdir.txt'); hfile = fgetl(fid); fclose(fid);
loadlibrary('ppp.dll', hfile);
h = uimenu('Label','Properties');
h1 = uimenu(h,'Label','Edit Properties','Callback','edSS');
h2 = uimenu(h,'Label','Open Archive','Callback','AOpen');
h3 = uimenu(h,'Label','Save a Archive','Callback','ASave');
end
d = calllib('ppp', 'StrGD', stream)
end
```

By typing in Matlab the command

```
>>StrGD(1)
```

Matlab executes the code within the script, it loads ppp.dll (if not in memory) , creates a menu bar (with the standard Prode Properties commands) and then executes the method StrGD, to calculate the density.

Notice that the script creates a menu bar which permits to access directly Prode Properties from Matlab GUI,
there are three commands

-edit Streams
-open a archive
-save a archive

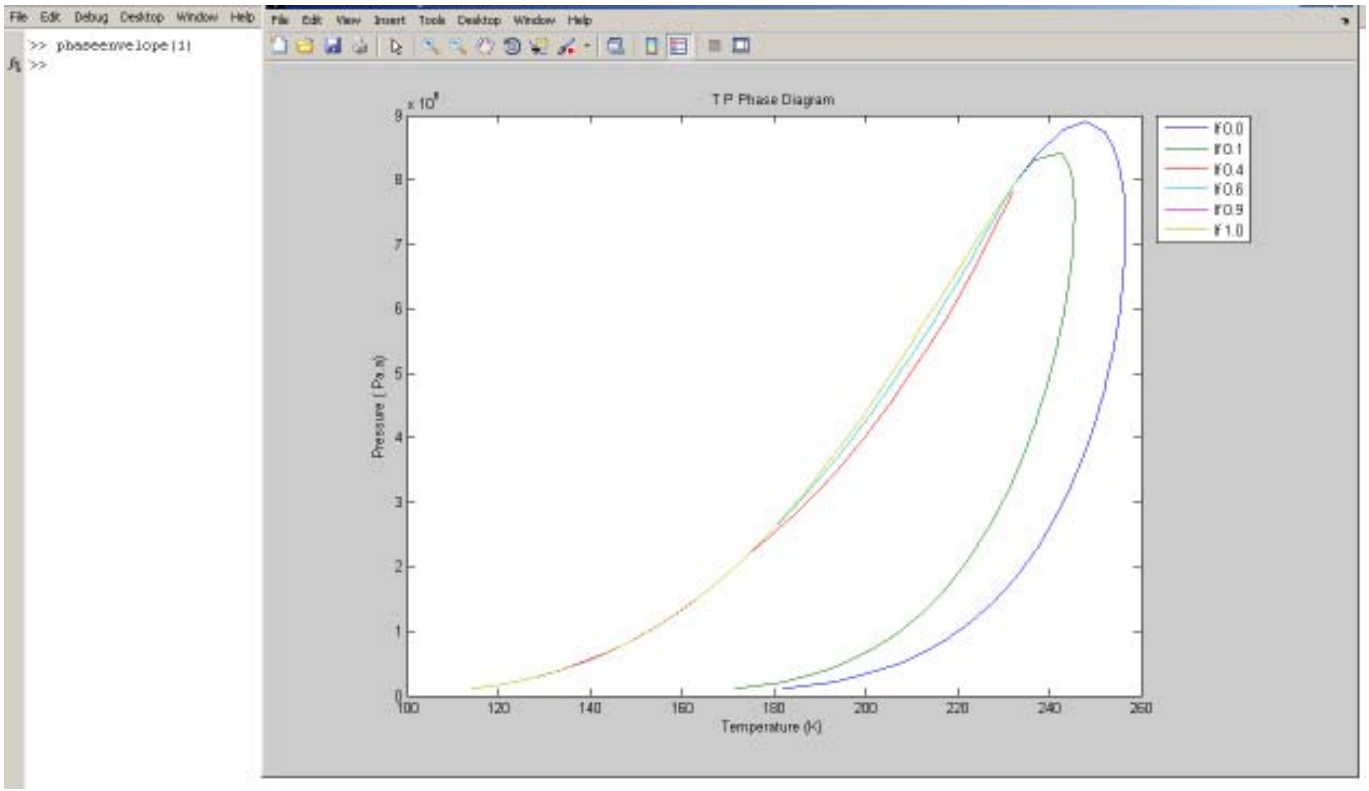
Important features of menu bar
-the characteristics may depend from Matlab version
-if you delete the associated figure the menu bar is deleted, to recreate the menu you must reenter the commands

```
h = uimenu('Label','Properties');
h1 = uimenu(h,'Label','Edit Properties','Callback','edSS');
h2 = uimenu(h,'Label','Open Archive','Callback','AOpen');
h3 = uimenu(h,'Label','Save a Archive','Callback','ASave');
```

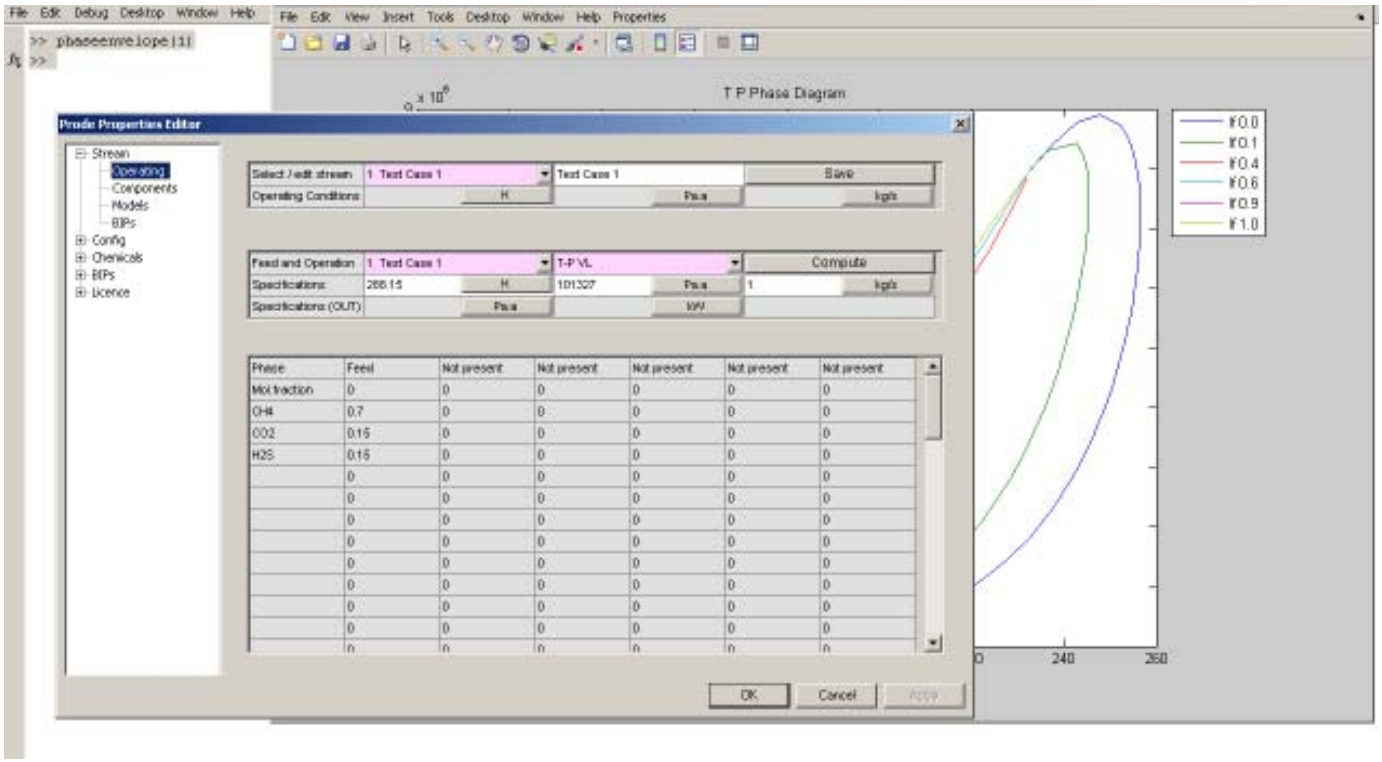
You can write scripts to solve more complex problems, an example is the script phaseenvelope.m which prints a phase envelope, to test the script type in Matlab the command

>>phaseenvelope(1)

Matlab will invoke Prode Properties to calculate the phase envelope for the stream 1 , then it plots the result



Notice that from Properties menu bar you can access Properties editor and modify the list of components or models of each stream



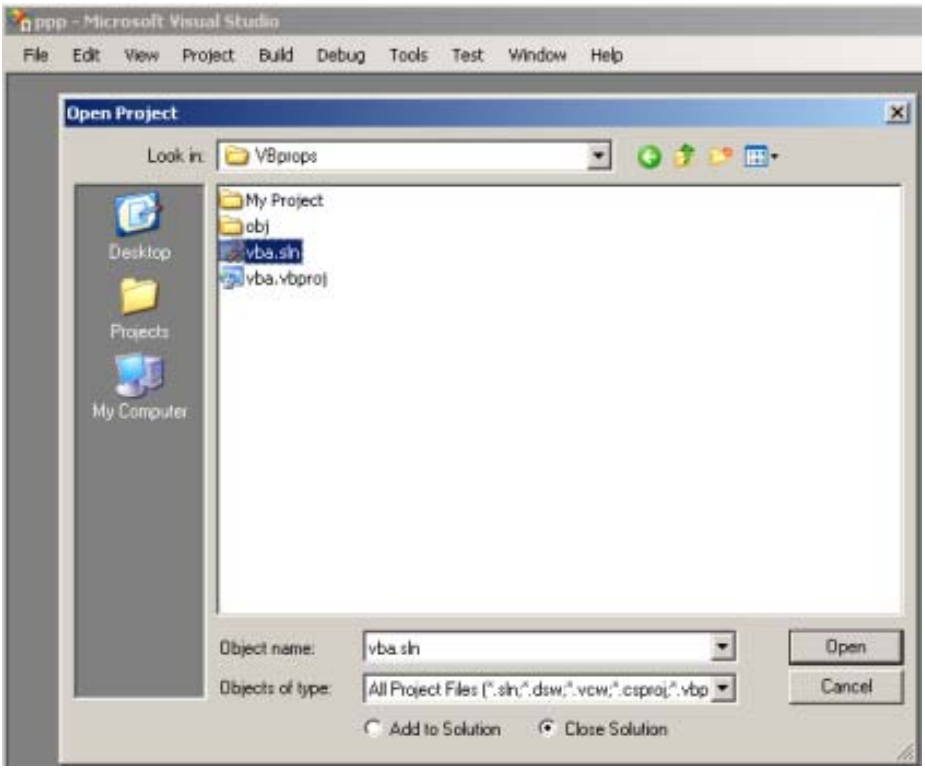
Access from MATHCAD

The files and the instructions required to link MathCad with Prode Properties are provided by Dr. Harvey Hensley , you can find application examples and download the files from web at HarveyHensley dot com

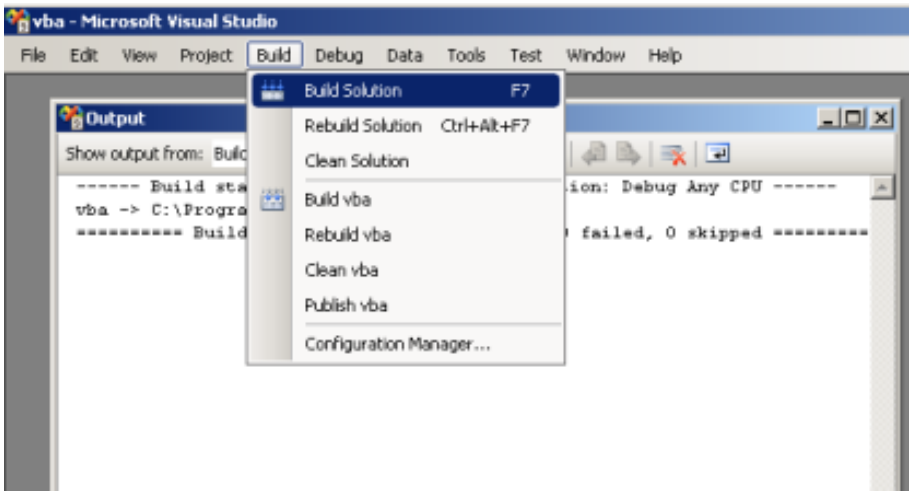
Getting started with Microsoft NET (VB , C) applications

IMPORTANT Microsoft NET support files are located in the directory \Prode\NET

Prode Properties can be easily included as unmanaged code in every Microsoft NET application, for compiling the sample code provided with Prode Properties a recent version of Microsoft Visual Studio is required.
From Microsoft Visual Studio compiler menu File->Open->Project/Solution , in NET folder (in Prode Properties installation) select the file vba.sln

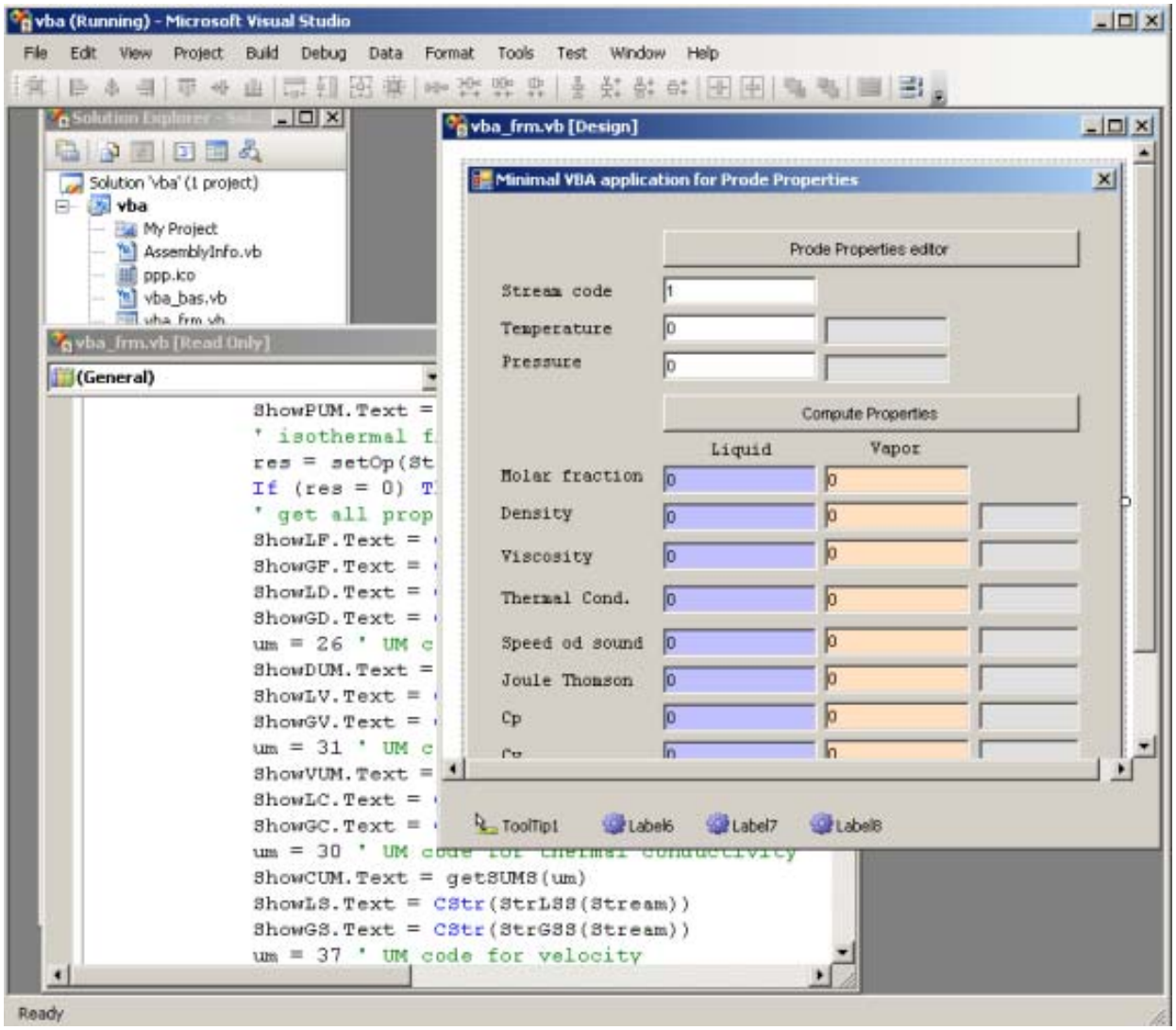


then from menu Build- select Build Solution.
Note: if desired you can edit the settings from Project->vba Properties



As next step you can test the application, from Visual Studio menu Debug->Start Debugging, then once the application is running :

1) click on the button Prode Properties editor to access the editor, define the streams and units of measurement
2) define a suitable temperature and pressure (with proper units)
3) click on button Compute Properties to print the properties



you can then modify the code according your requirements.

Solving problems (introduction)

There are several different classes of problems which Prode Properties can help to solve but the most common are probably :
- physical properties of pure fluids and mixtures
- equipment design
- system simulation

Prode Properties provides many methods for the prediction of physical properties, in general a single instruction is required for calculating a property.
The design and rating of unit operations as distillation columns, towers, pumps, compressors, valves, heat exchangers etc. is another area where Prode Properties can result useful, the use of programming languages is generally suggested when dealing with complex problems while some formula in a worksheet can solve the usual work.
The system simulation may be used in the design stage to evaluate parameters, to help achieve an improved design or applied to existing systems for optimizing operating conditions. Generally the required solution is the list of operating conditions at the input and output of the operating blocks in the simulation block diagram. When there are no recycle streams or controls the method for solving the system is very simple : the output information from the first operating block is utilized as input for the second operating block and so on. However when there are output conditions which may interfere with input conditions some sort of iteration is required since some or all the equations governing the system may be non linear. There are two well known methods for solving such a system of non linear equations, the method of successive substitutions and Newton-Raphson, refer to good books of numerical analysis for additional information.

Streams
Most thermodynamic calcs in Prode Properties library take as reference a stream entity. For example when simulating a plant it makes sense to define different streams to represent flows in different sections, a stream usually defines compositions and operating conditions, Prode Properties supports a variable number of streams and most methods in Prode Properties require a reference to a stream, the reference is a numeric code (a progressive integer starting from 1 for first stream) .

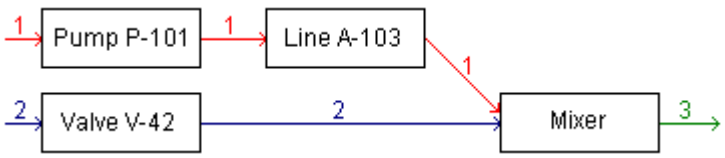
Streams attributes
As in process simulators each stream may include following information
- a list of components and relative weights
- a value for the operating pressure
- a value for the operating temperature
- a value for the operating flow
- thermodynamic models for different properties
- a list of BIPs

Working with streams
Prode Properties permits to define complex topologies as there is no limit to the number of operating blocks required for simulating a plant, with Prode Properties for simulating a plant you convert the different sections into pieces of code, to do so you can use the basic blocks available in all process simulators, for esample

- **isothermal flash**, for calculating multiphase equilibria at the specified temperature ad pressure
- **flash unit** (enthalpy, entropy or volume basis), calculates output temperature or pressure, with this unit you can simulate pipelines, valves, heat exchangers, pumps, compressors and many others operations.
- **fixed vapor fraction flash**, for constructing phase envelopes, calculating bubble and dew points etc.
- **mixer** to add the contents of two streams
- **divider** to subtract a part of flow from a stream

by putting together these blocks it is possible to simulate also complex plants.

Simulating a plant
transform the flow sheet in a simulation block diagram, fluid and energy flow diagrams are standard engineering tools, you assign a number to the different streams and identify the basic blocks which will be solved by Prode Properties.



Notice the number which identifies each stream, in this case different numbers mean (possible) different compositions (we do not consider chemical reactions here), the output of each block can be easily calculated providing the input has been defined

Working with archives, save and load data, default settings

Load and save archives
Archives are files which contain the data required by Prode Properties to work with stream's and units of measurement, when you open an archive the stream's data and units are loaded, when you choose to save an archive these data are stored in a file, in this way you can work with many different projetcs.
Prode Properties includes several methods to save and load data as archives.

The default settings
When Prode Properties starts it loads data from the archive named "def.ppp" so if you wish to use your own list of streams,units etc. just save your preferred settings under the name "def.ppp".

Properties editor

Prode Properties includes Properties editor, from the editor you have access to

- **Streams** edit operating conditions, flow, compositions, models, BIPs for all streams
- **Config** edit all units of measurement and settings
- **Chemicals** edit all chemical's data, regress data , add new chemicals
- **BIPs** edit BIPs, regress data, add new BIPs

How to activate the Properties editor

call one of the methods edS(stream), edSS() remember that in Prode Properties each stream is referenced with a code (integer value) in the range (1... max number of streams).

From Microsoft VB

```
Call edSS()
Call edS(8) ' start editing stream nr. 8
```

From Microsoft Excel

```
=edSS()
=edS(8)  ' start editing stream nr. 8
```

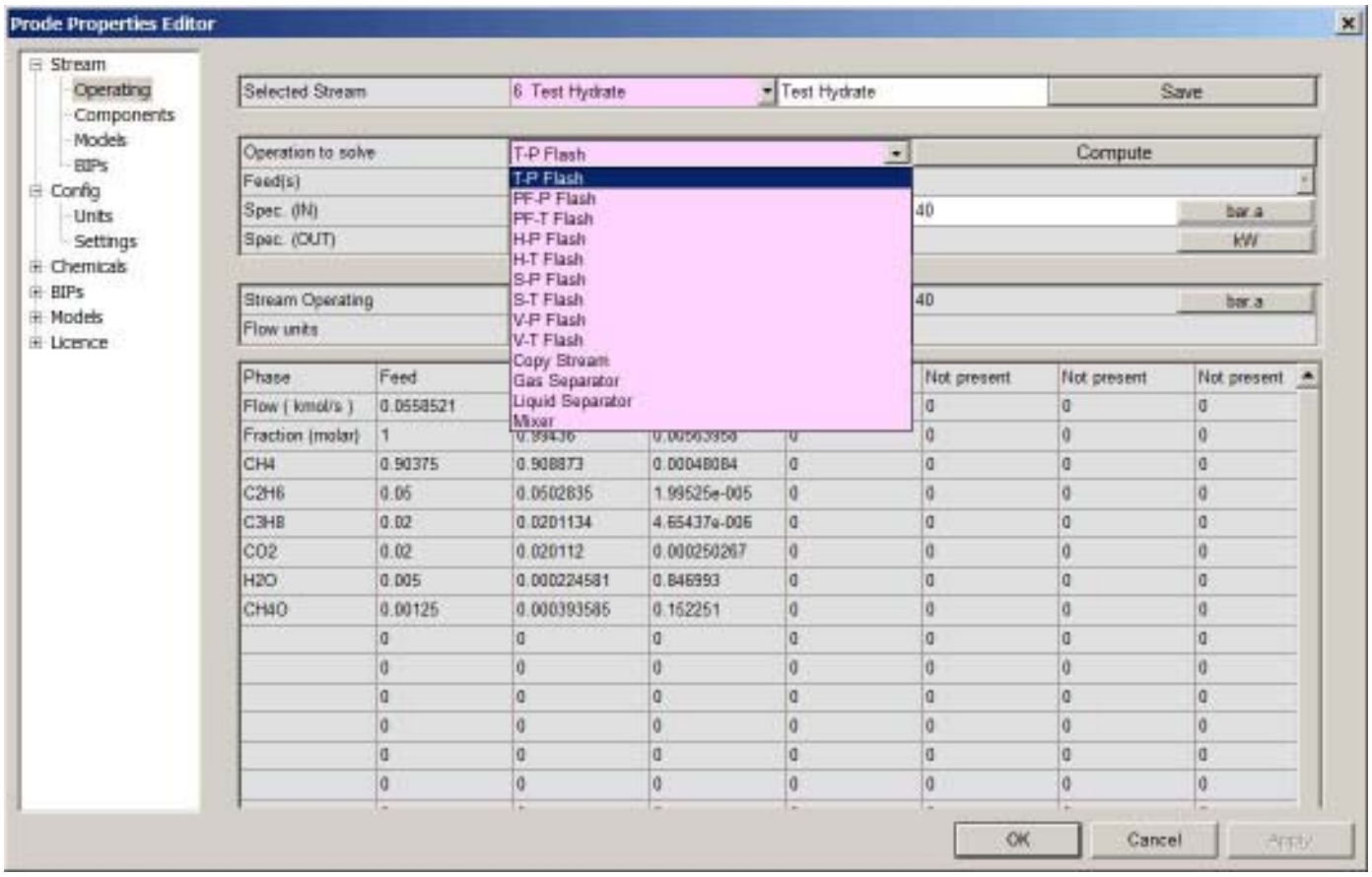
From Microsoft Visual C++

```
edSS();
edS(8);  // start editing stream nr. 8
```

Stream operating

From this page you can :

- select a stream (select / edit stream)
- define a name for the selected stream
- save edited stream data
- Compute different flash operatins



Define a new stream / edit existing streams

- select the stream (you wish to edit / define) from the “Select / edit stream list”
- go to page “Components”, define the list of components and relative weights
- go to page “Models”, define the models for the different properties, define settings, load BIPs
- back to page “Operating”, put a name for the stream and click on “Save”
- check that correct stream composition appears on the second grid

Compute a flash operation

- make sure a feeding stream has been properly defined (composition, models, settings) then select the feeding stream
- select the flash operation
- define the required specifications and clik on “Compute”

Predefined Operations

Prode Properties (Base) allows to solve directly from editor a few operations (see paragraph “Methods for thermodynamic calc’ s” for the list of methods available in library), these methods can solve multiphase equilibria (vapor-liquid-solid phases)

T-P Flash Temperature, Pressure, Flash Operation

LF-P Phase Fraction, Pressure, Flash Operation
LF-T Phase Fraction, Temperature, Flash Operation

H-P Flash Enthalpy, Pressure, Flash Operation
H-T Flash Enthalpy, Temperature, Flash Operation

S-P Flash Enttropy, Pressure, Flash Operation
S-T Flash Entropy, Temperature, Flash Operation

V-P Flash Volume, Pressure, Flash Operation
V-T Flash Volume, Temperature, Flash Operation

Copy Stream copy a stream into another stream

Gas Sep. simulates a gas separation at specified temperature and pressure

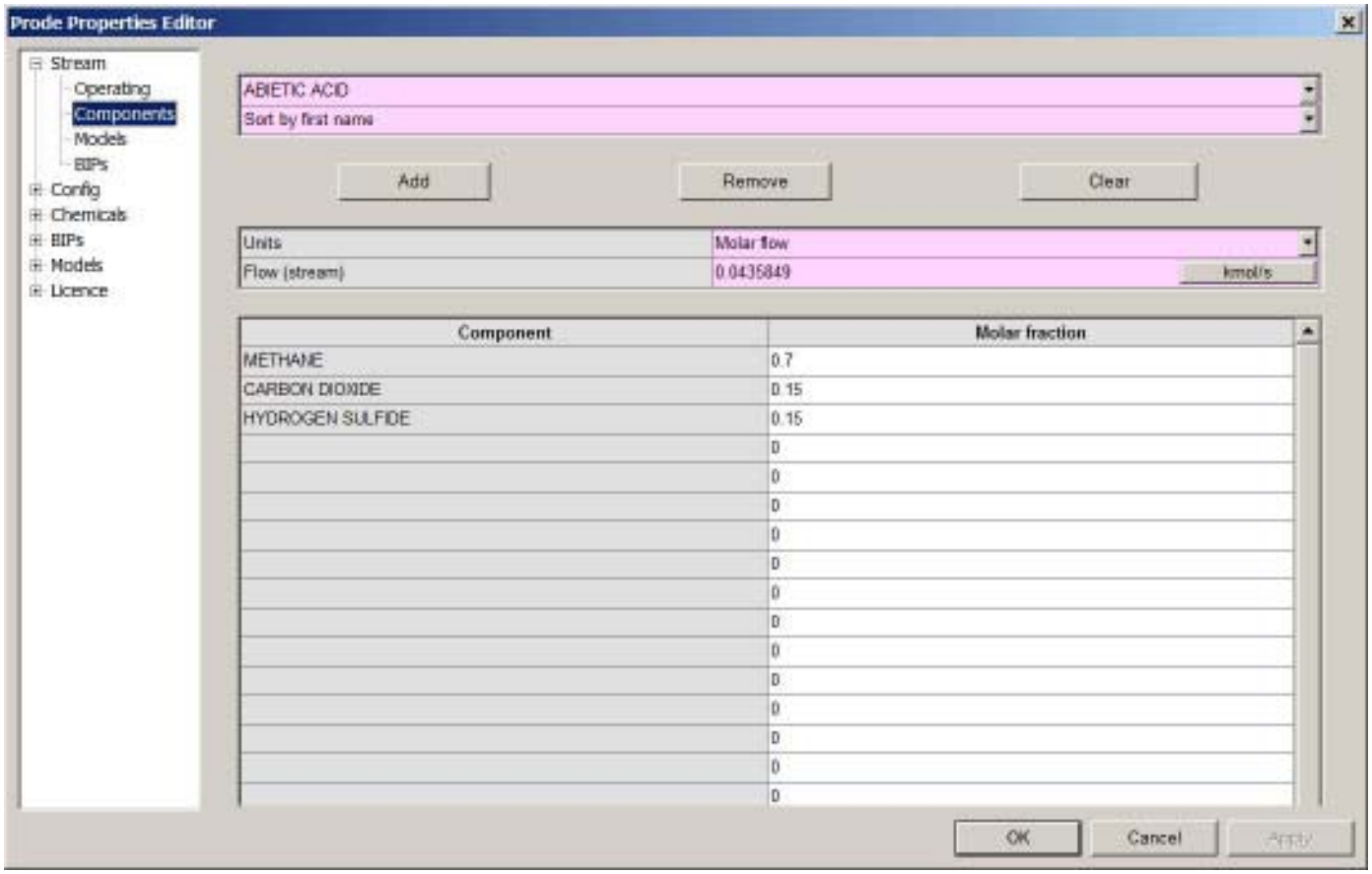
Liquid Sep. simulates a liquid separation at specified temperature and pressure

Mixer mix two streams

Stream Composition

From this page you can :

- define a list of components by selecting components from the library
- specifyi the amount of each component.



Define the sorting criteria

- select the preferred criteria

Add a component to the list

- select a component from the list of components
- click on **Add** button

Remove a component from the list

- click on **Remove** button to remove the last component in the list

Clear the list

- click on **Clear** button to rclear all components in the list

Units

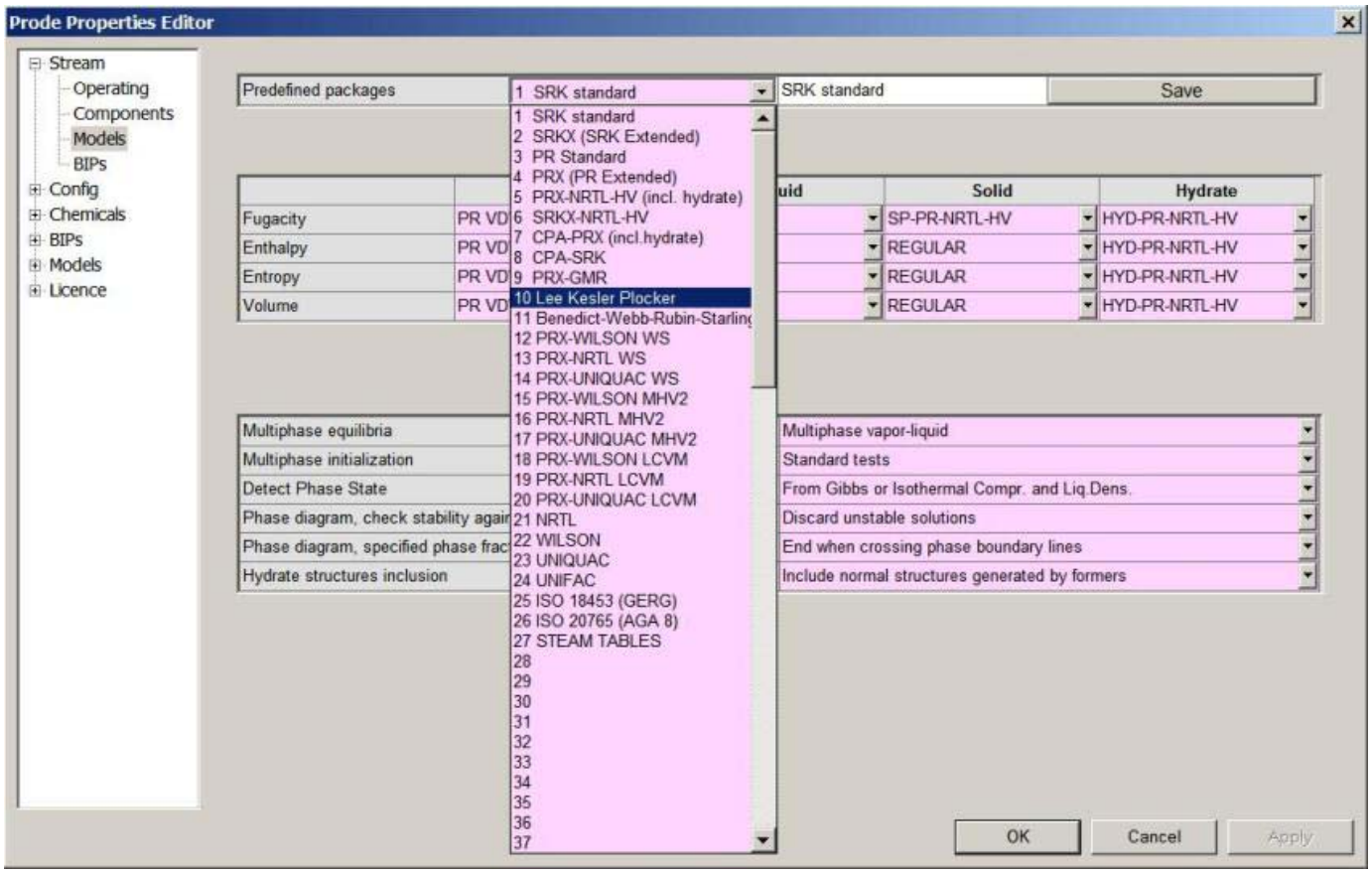
- select the desired Mole or Weight fractions (and flow)

Define flow (stream)

- enter flow (stream)

Stream Models

From this page you can :
- define many different packages with user defined models and options
- set models per each propertty and state (gas,liquid,solid)
- set different options



Define a new package / Edit existing package
- select a package in the list
- in Models and Options Window select the models and options for this package
- if required edit / change the name for the package,
- click on “Save” button to save this package

Select a package
- select a package in the list

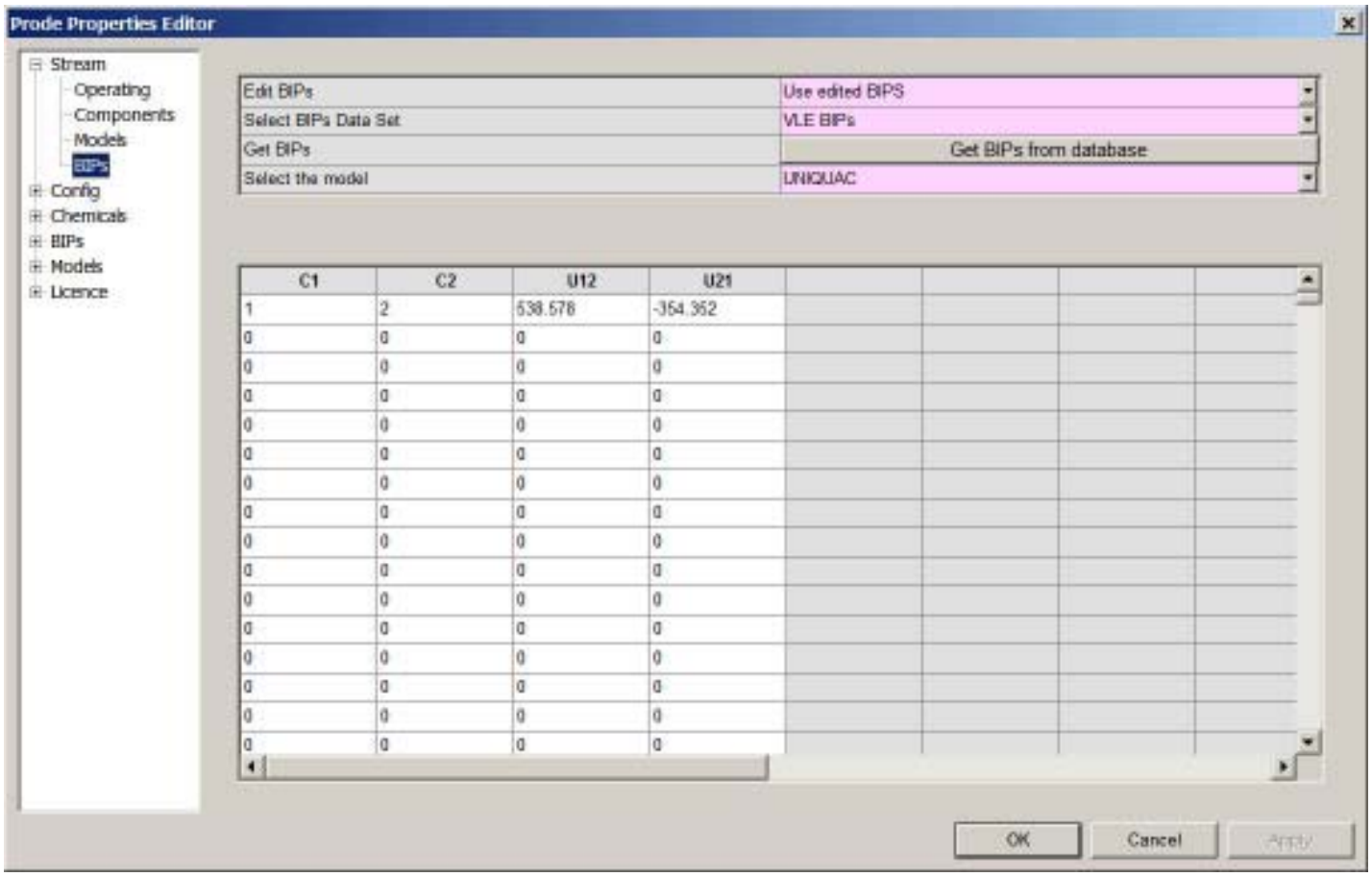
Define models
- specify the models per each property and state

Set a option
- Multiphase equilibria, allows to define different solutions as vapor-liquid, vapor-liquid-liquid and vapor-liquid-solid
- Multiphase initialization, allows to reduce the number of trial phases thus reducing time required
- Detect Phase State, allows to use different methods to detect the state of each phase
- Phase diagram, check stability against feed, allows to include stability analysis on each calculated point
- Phase diagram, specified phase fraction lines, allows to terminate lines when crossing a phase boundary
- Hydrates structures inclusion, allows to test all possible hydrate structures which may be generated by former(s)

BIPs

From this page you can :
- input / edit / load BIPs for the different models

Define a list of BIPs
- click on **Get BIPs** for loading all BIPs available in library
- if required add your own specific BIPs



IMPORTANT

For BIPs the first two columns Ci and Cj define the component's position in the list (i.e. 1 for the first component , 2 for second and so on), while othe colums allow to enter the values for the different BIPs required by selected model, note that some values (for example in Wilson, NRTL etc.models) have units of cal / gr.mol.

Example, binary of water methanol UNIQUAC U12 : 538, U21 :-354 in the first two columns (c1 , c2) enter the components relative position in the list, assuming that water is the first component and methanol the second

c1	**c2**	**U1-2**	**U2-1**
1	2	538	-354

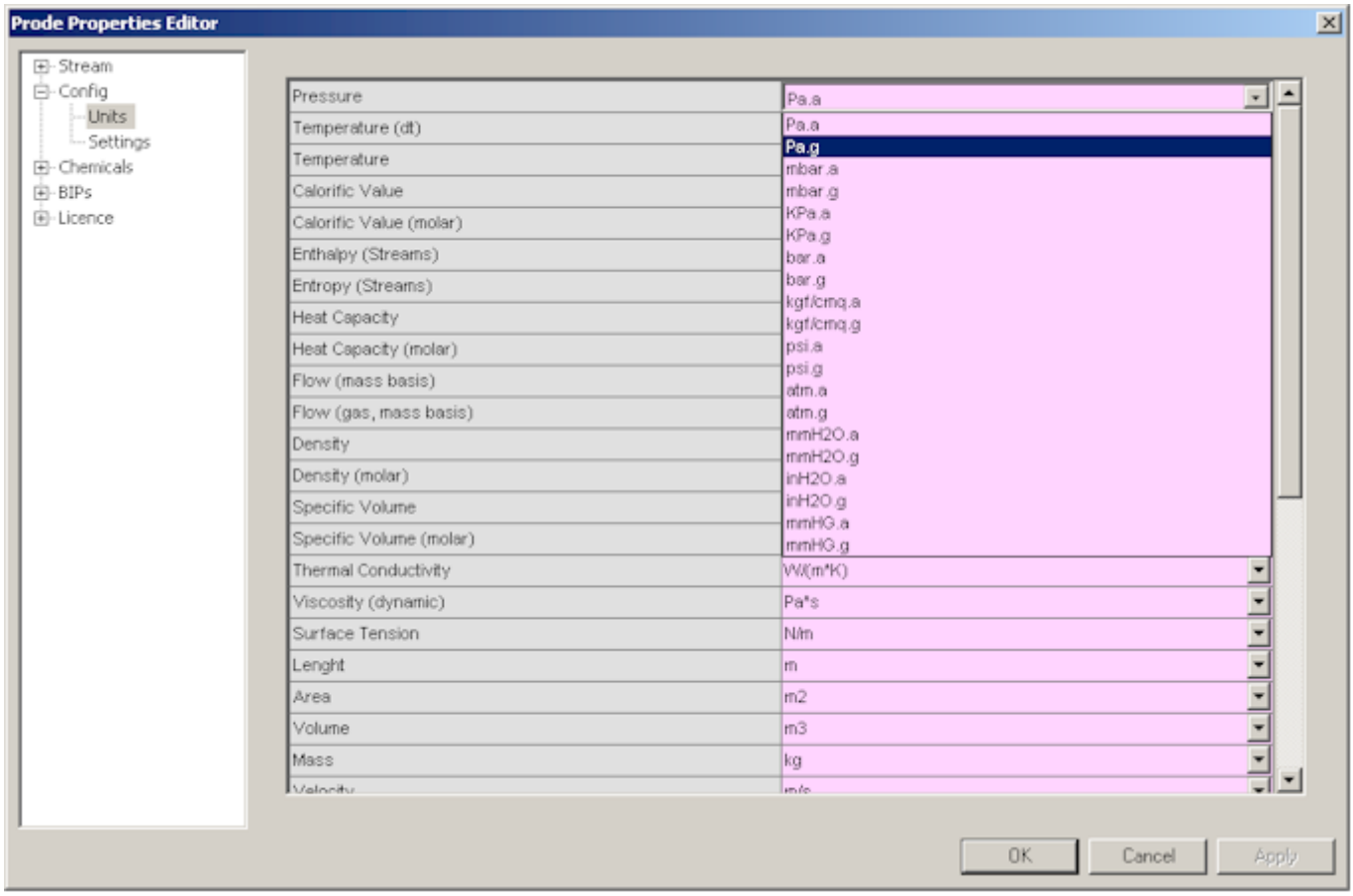
MPORTANT

Prode Properties allows to define Temperature Dependent BIPs for many models, Temperature Dependent BIPs can provide additional accuracy.

Config Units

From this page you can :

- define the unit of measurement for the different properties



IMPORTANT

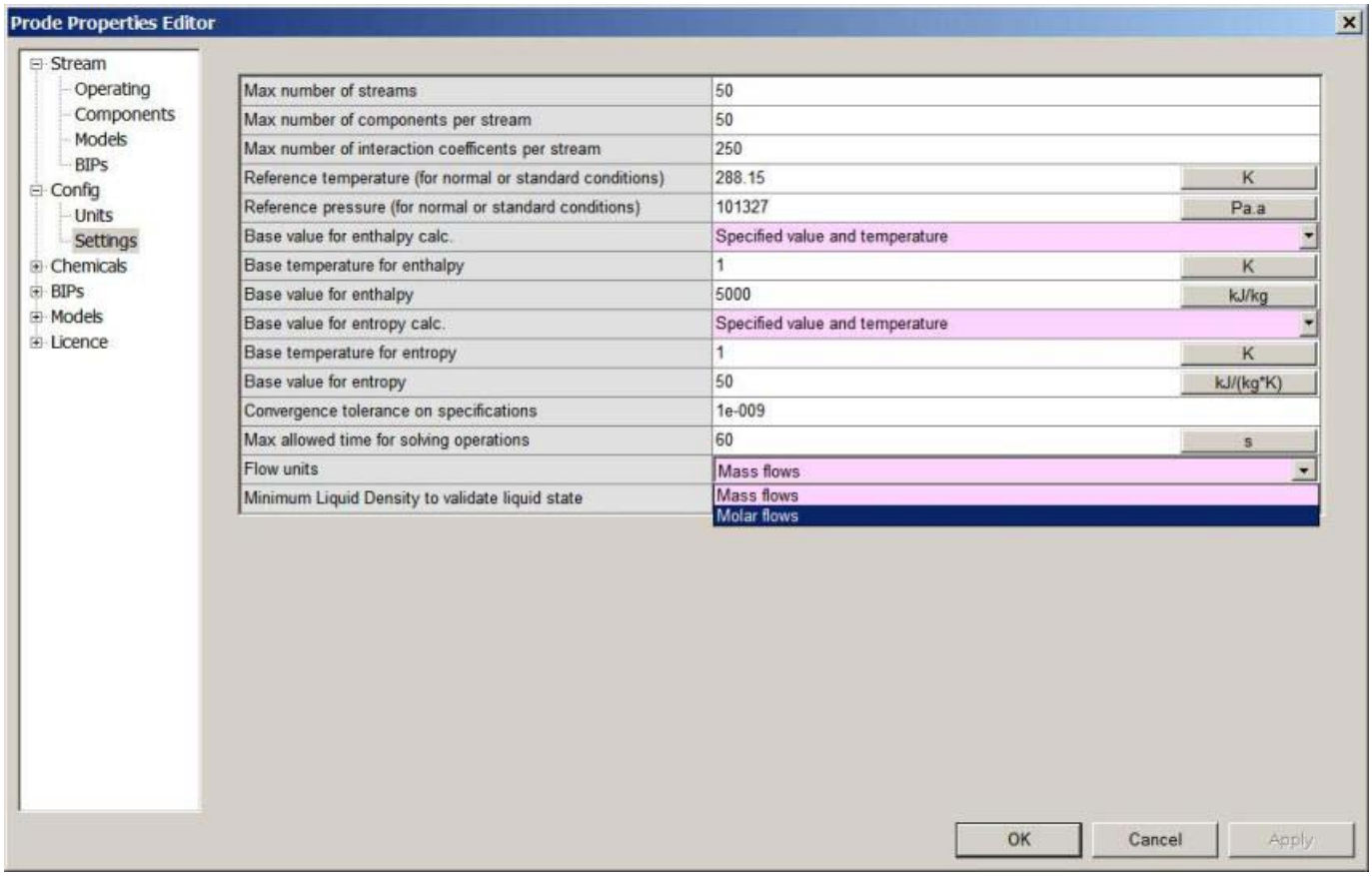
With Prode Properties you have complete control over the engineering units, this means that the program permits to select from a list of different units for each property, and Prode Properties automatically converts the input values and the results according to the selected units.

1) if you set Bar.a as unit for pressure all inputs and outputs will be in that unit
2) the new units will become effective after closing the Editor (select Ok button to confirm the Units)
3) if you wish to define your own set of units remember, before to leave the application, to save data into a archive otherways your changes will be lost

Config Settings

From this page you can :

- define the different settings as max number of streams, the temperature and pressure at reference conditions, the base values for entalpy and entropy, convergence tolerance etc.



configurable parameters :

- max number of streams
- max number of components per stream
- max number of interaction coefficients pairs per stream
- reference temperature and pressure
- base values for enthalpy and entropy calc's
- convergence tolerance
- max allowed time for solving a operation
- Flow units
- minimum liquid density to validate liquid phase

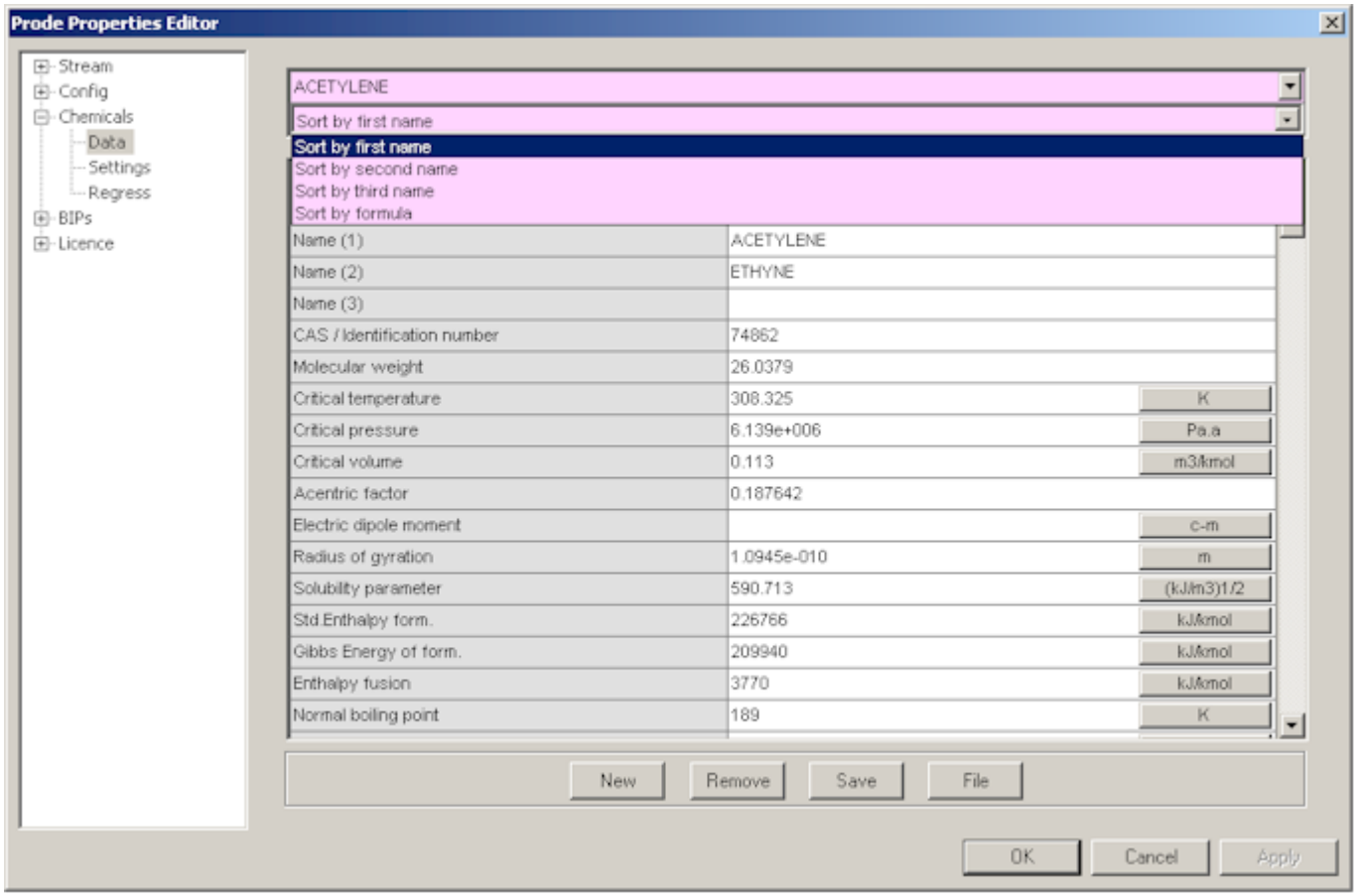
IMPORTANT

before to leave the application remember to save data into the archive differently your changes will be lost

Chemicals data

From this page you can :

- edit and change the physical properties data included in the databank
- save all data in a file



Edit / modify data :

- select a component from the component's list
- edit / modify the related fields (see also the section with description of all fields)
- select the "Save" button to save the modified data (differently new data will be discharged)

Adding a new component :

- select the "New" button
- edit the related fields (see also the following page with description of all data fields)
- select the "Save" button to save the data (differently new data will be discharged)

Remove a component :

- select a component from the component's list
- select the "Remove" button

IMPORTANT

Updating the file which contains the databank :

this option permits to store all data into a file, differently all changes will be lost when leaving the application

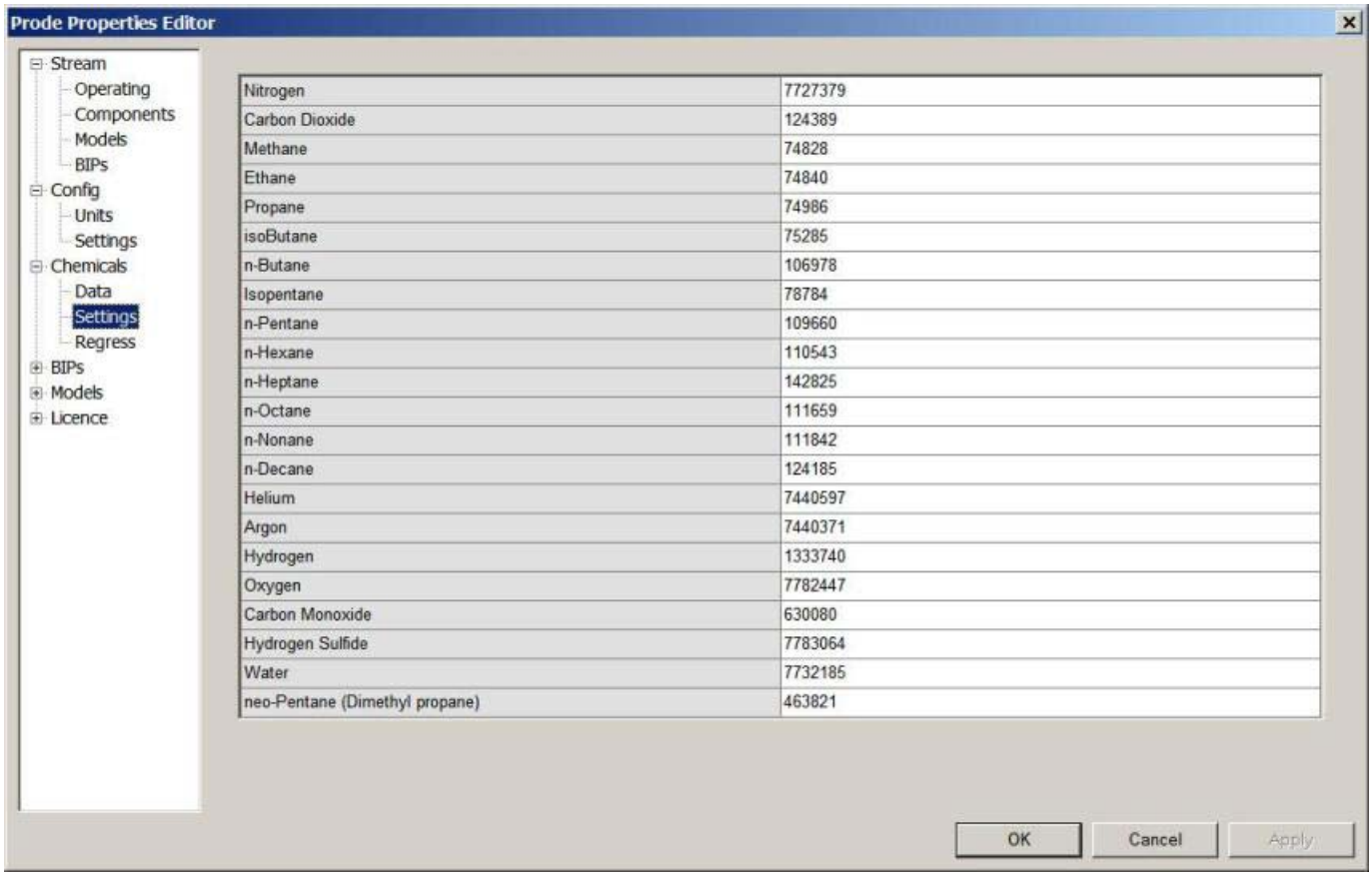
- select the "File" button

CAUTION : you may wish to create a backup of the file chem.dat before to overwrite the file

Chemicals Settings

From this page you can :

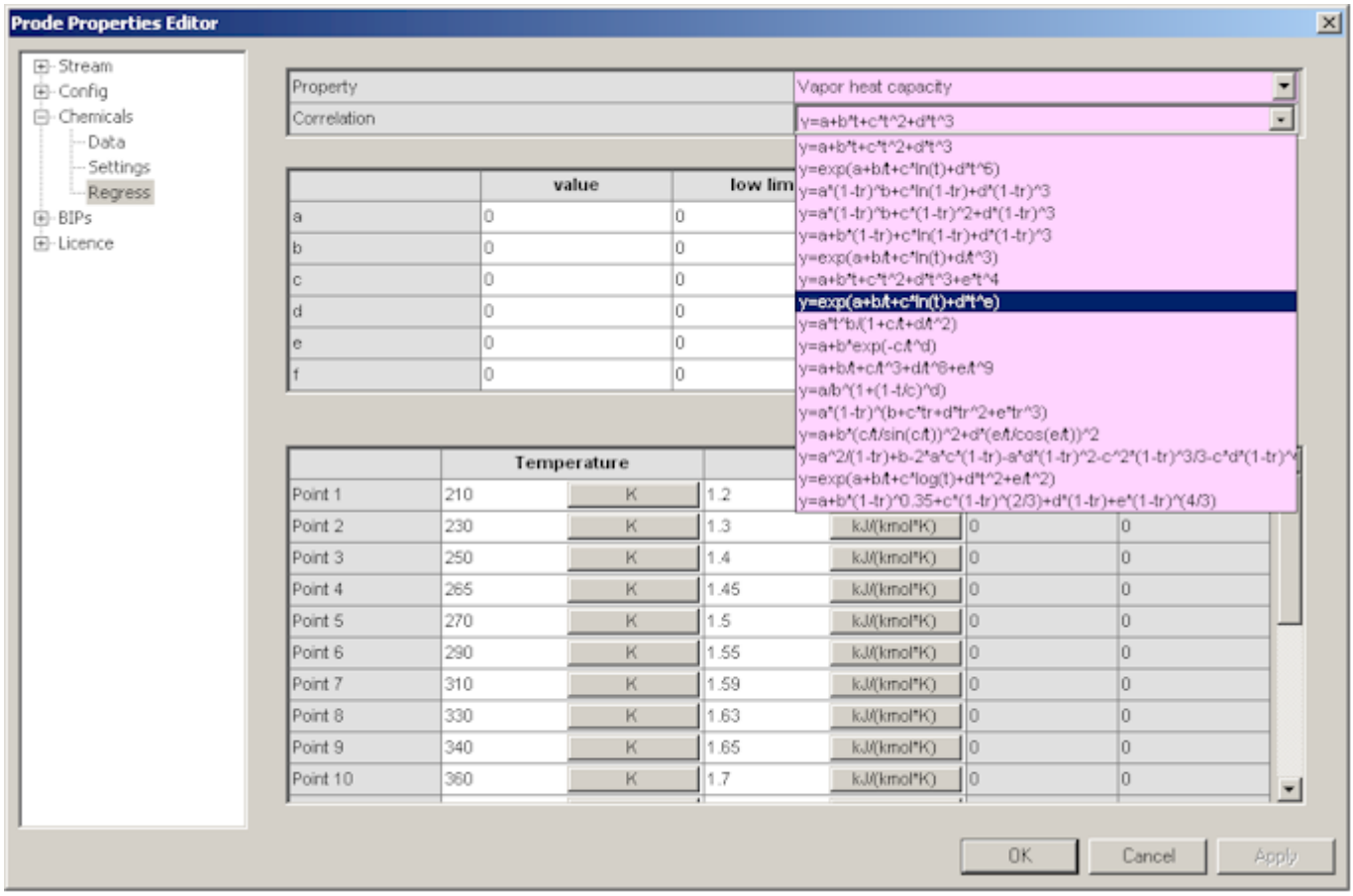
- edit informations (CAS number) required to identify some components .



Regress raw data

From this page you can :

- regress raw data into values compatible with chemical's database



Regress raw data

- in Chemicals Data page select a chemical
- in Chemical Regress page selet a property and the correlation for fitting raw data
- enter the available data (all temperature and value pairs) with the proper units of measurement
- if required you may enter initializing values and limits for calculated parameters
- click on Calculate button, evaluate calculated values and errors, you may try different correlations for best data fitting
- click on Transfer button to copy calculated values into Chemicals Data page
- in Chemicals Data page select the "Save" button to save the data (differently new data will be discharged)

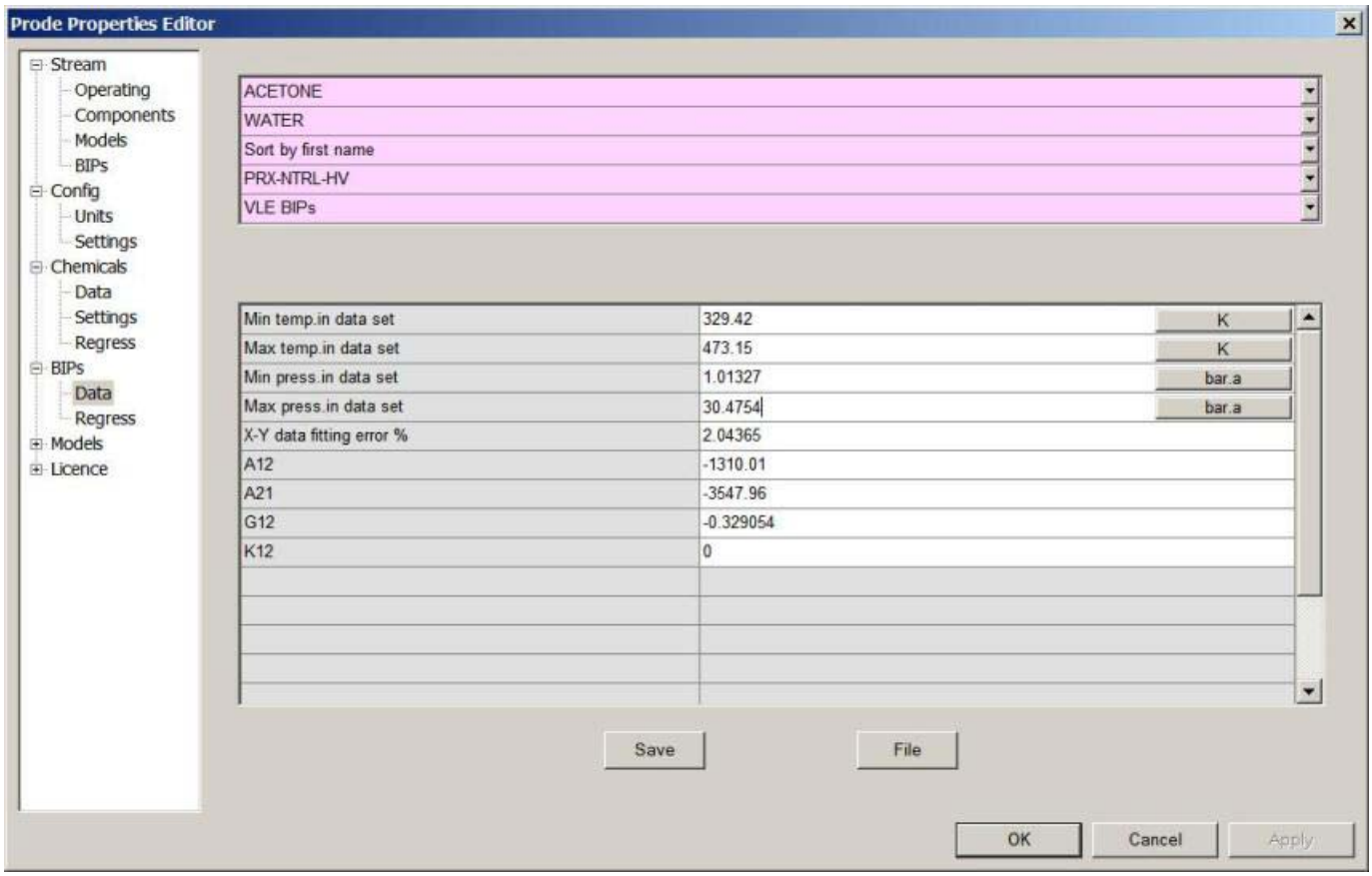
IMPORTANT

Prode Properties flexible datbase format supports more than 15 different correlations (30 in extended version), so for each property you can select the correlation which best fits experimental data.

Binary Interaction Parameters (BIP)

From these pages you can :

- edit Binary Interaction Parameters
- add / remove Binary Interaction Parameters
- regress VLE (vapor-liquid) , LLE (liquid-liquid) , SLE (solid-liquid) data points
- save all data in a file



Edit / modify data :

- select two components from the component's lists
- select the database (VLE/LLE/SLE/Hydrate)
- select the model
- edit / modify BIPs
- select the "Save" button to save the modified data (differently new data will be discharged)

IMPORTANT

Updating the file which contains the BIPs data :

this option permits to store all data into a file, differently all changes will be lost when leaving the application

- select the "File" button

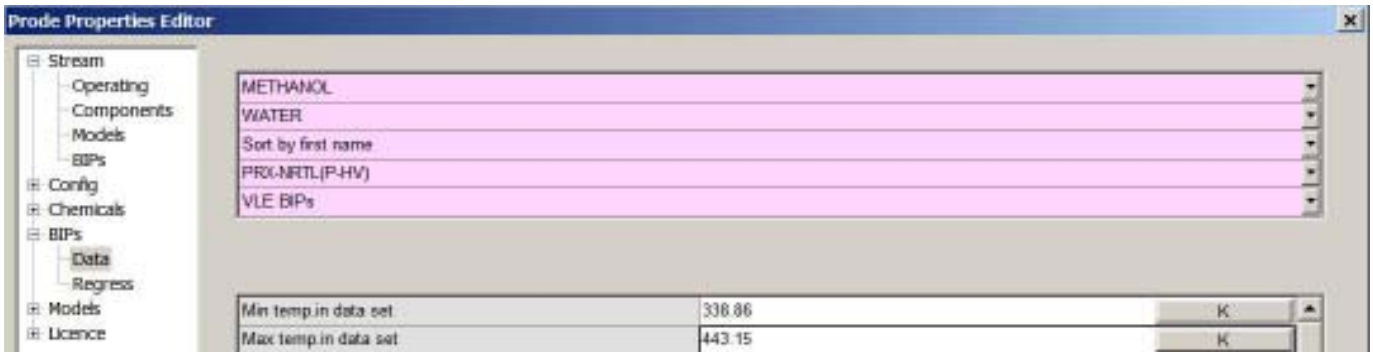
CAUTION you may wish to create a backup of the file bips.dat before to overwrite the file

Regress VLE-LLE-SLE data

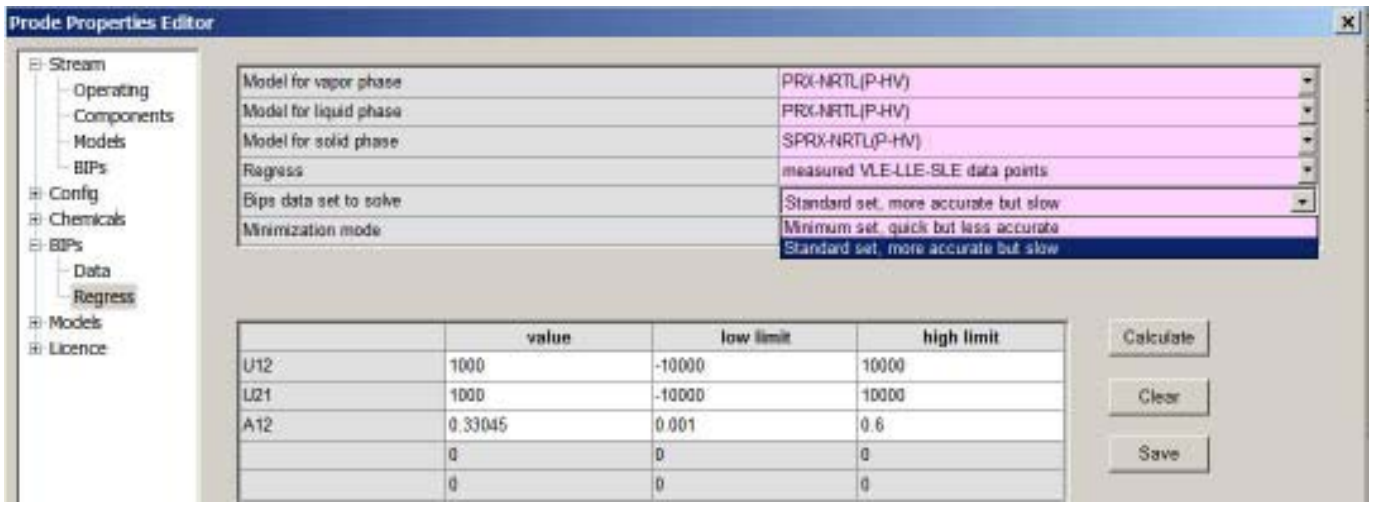
From this page you can :

- regress VLE-LLE-SLE data for calculating the best fitting paraneters for different models

as first step in BIPs->Data page you must select two components (in this example methanol, first component, and water)



The procedure permits to enter experimental (measured) VLE-LLE-SLE data points or fit automatically the model to VLE points calculated with UNIFAC.



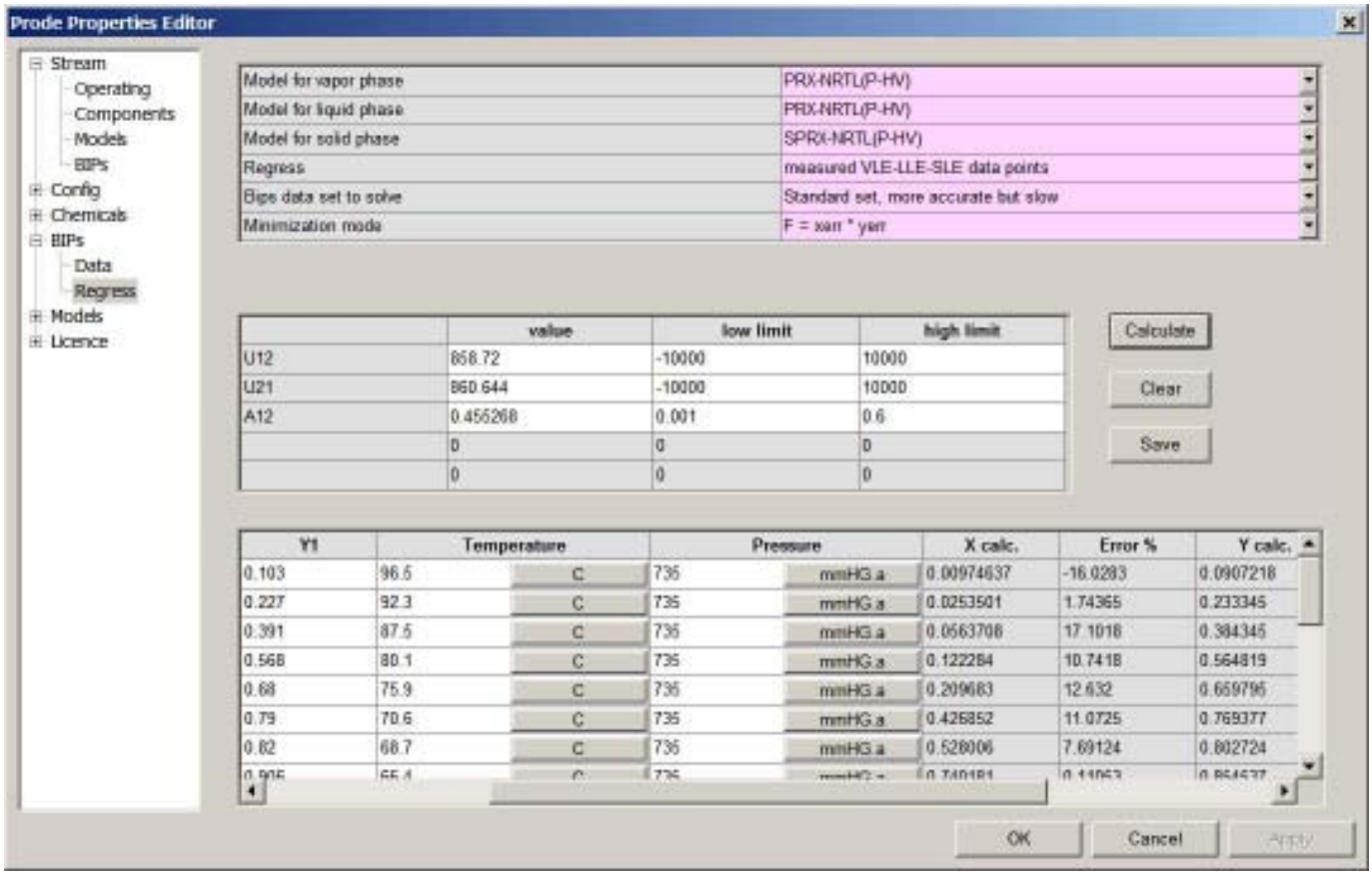
- select the models for the different states (in this case PRX-NRTL-HV)
- select measured VLE-LLE-SLE data points as data to regress
- select Standard as solution mode, this option allows to calculate all the BIPs for specified model
- select F = xerr*yerr as minimization mode, this is the default

enter the measured VLE (vapor-liquid data points)
on each row include in X1 column the measured liquid molar fraction of component 1, in Y1 column the measured vapor molar fraction of component 1, for LLE (liquid-liquid equilibria) include in X1 column the measured liquid molar fraction of component 1 in first liquid phase, and in Y1 column the measured liquid molar fraction of component 1 in second liquid phase, for SLE (solid-liquid equilibria) include in X1 column the measured liquid molar fraction of component 1 in liquid phase, and in Y1 column the measured solid molar fraction of component 1, finally enter the temperature and the pressure for that point.
Example of measured VLE data for Methanol-Water

VLE data for Methanol-Water at 735 mmHg

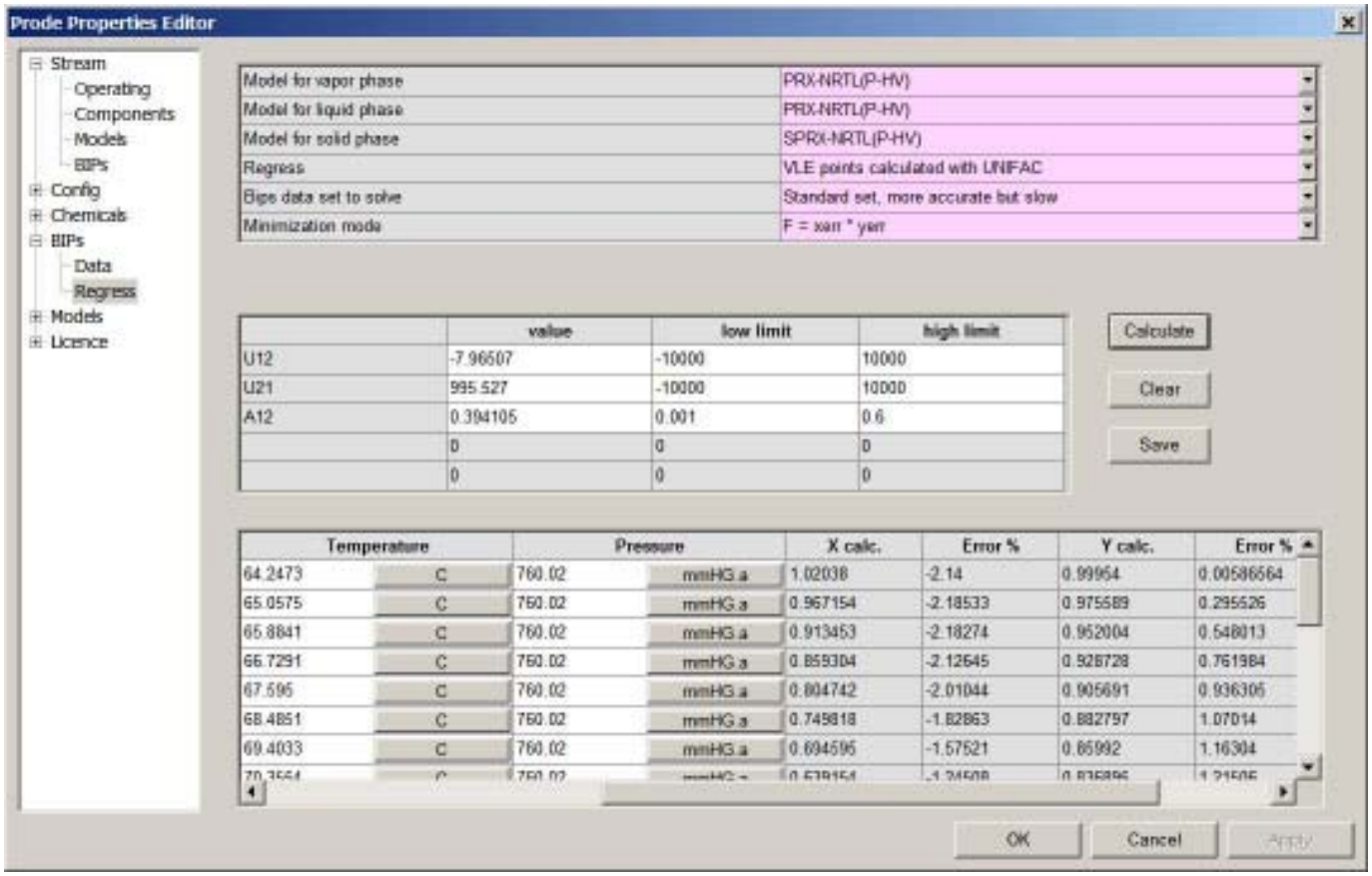
Point	X(1)	Y(1)	Temperature (C)	Pressure (mmHg)
VLE	0.008400	0.103000	96.5	735
VLE	0.025800	0.227000	92.3	735
VLE	0.068000	0.391000	87.5	735
VLE	0.137000	0.568000	80.1	735
VLE	0.240000	0.680000	75.9	735
VLE	0.480000	0.790000	70.6	735
VLE	0.572000	0.820000	68.7	735
VLE	0.741000	0.906000	66.4	735

- in Data dialog select the two components, in this example first is methanol, second is water
- in Regress dialog :
- select the models for the vapor and liquid phases (in this case PRX-NRTL-HV)
- select measured VLE-LLE-SLE data points as data to regress
- select Satndard as solution mode, this option allows to calculate all the BIPs for specified model
  if you do not select the Standard solution mode the procedure shows (for PRX-NRTL-HV model) only one BIP which is Kij in Peng-Robinson EOS , in this way you can calculate both sets
- select F = xerr*yerr as minimization mode, this is the default
- enter the measured data (see previous page) selecting the proper units

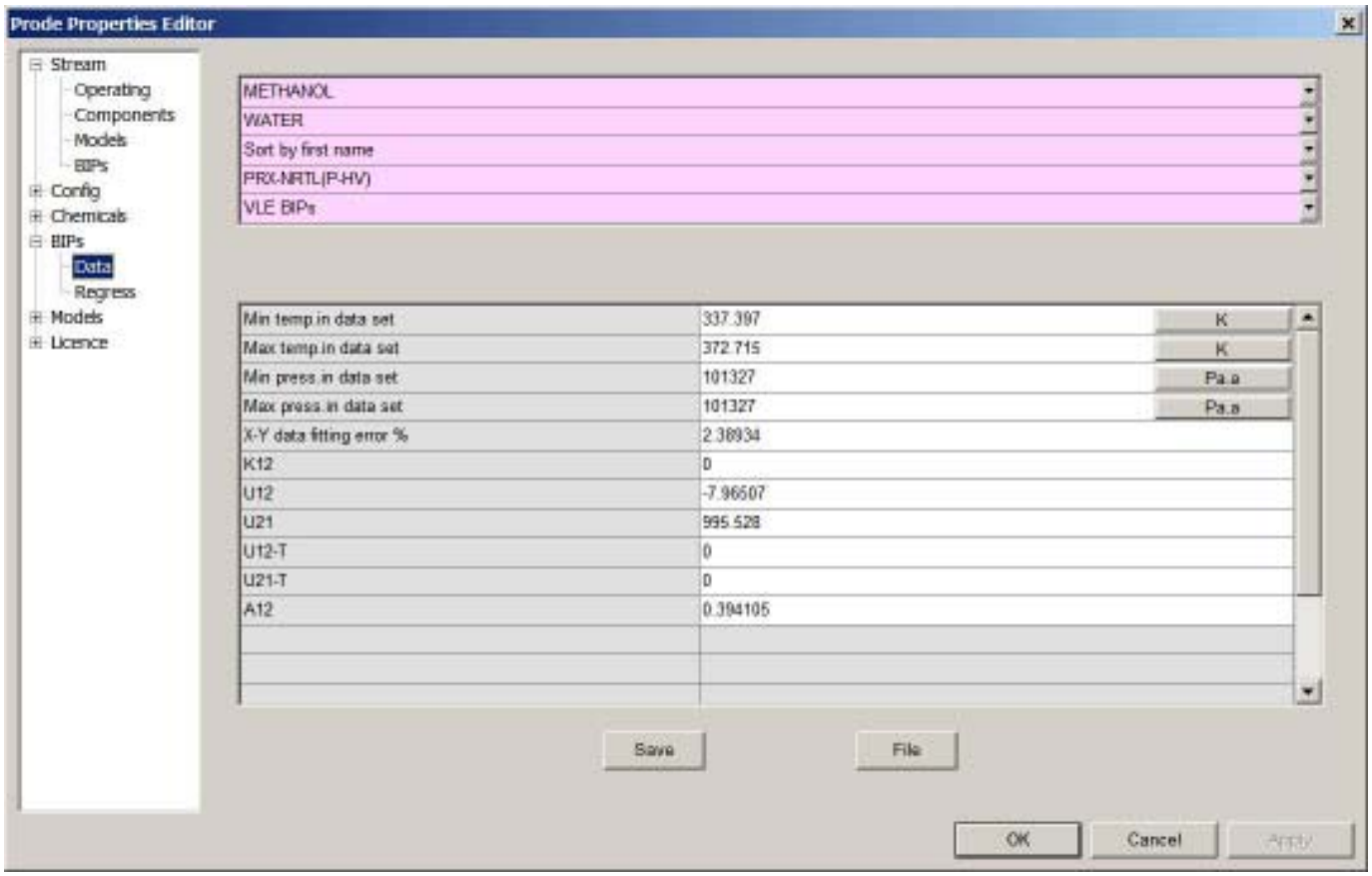


Selecting the button Calculate the procedure calculates the best fitting parameters, shows the calculatd results and the relative errors (%)
you can easily compare the different models

as alternative the procedure offers the option to calculate the best fitting parameters to VLE data points calculated with UNIFAC, this entirely predictive method doesn't require experimental data and in some cases can result useful.Select "VLE points calculated with UNIFAC" in "Regress" selector to use this method



Once the paramters have been calculated you must select the "Save" button to transfer the results in memory, from BIPs->Data page you can then inspect the values and store permanently in file (to store in the file select File button).

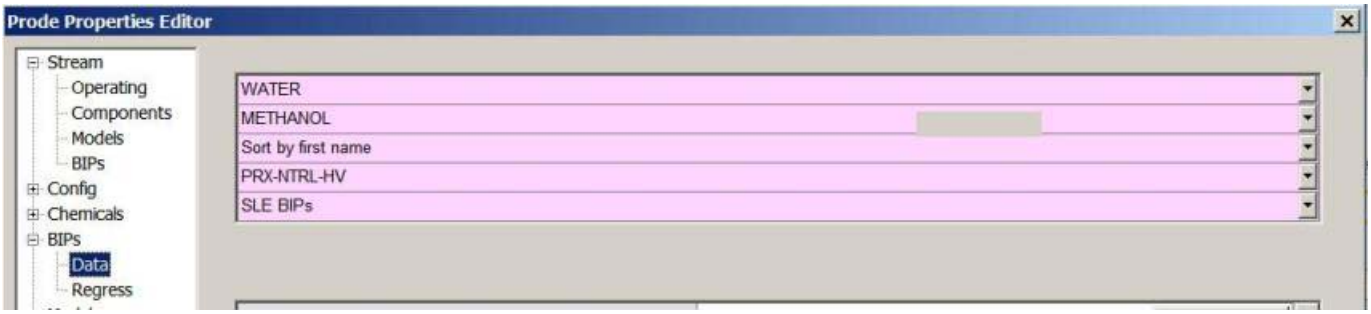


Regress SLE (Water-Methanol) and test the calculated freezing point depression
This example will show how to estimate BIPs for different models from available SLE equilibrium points

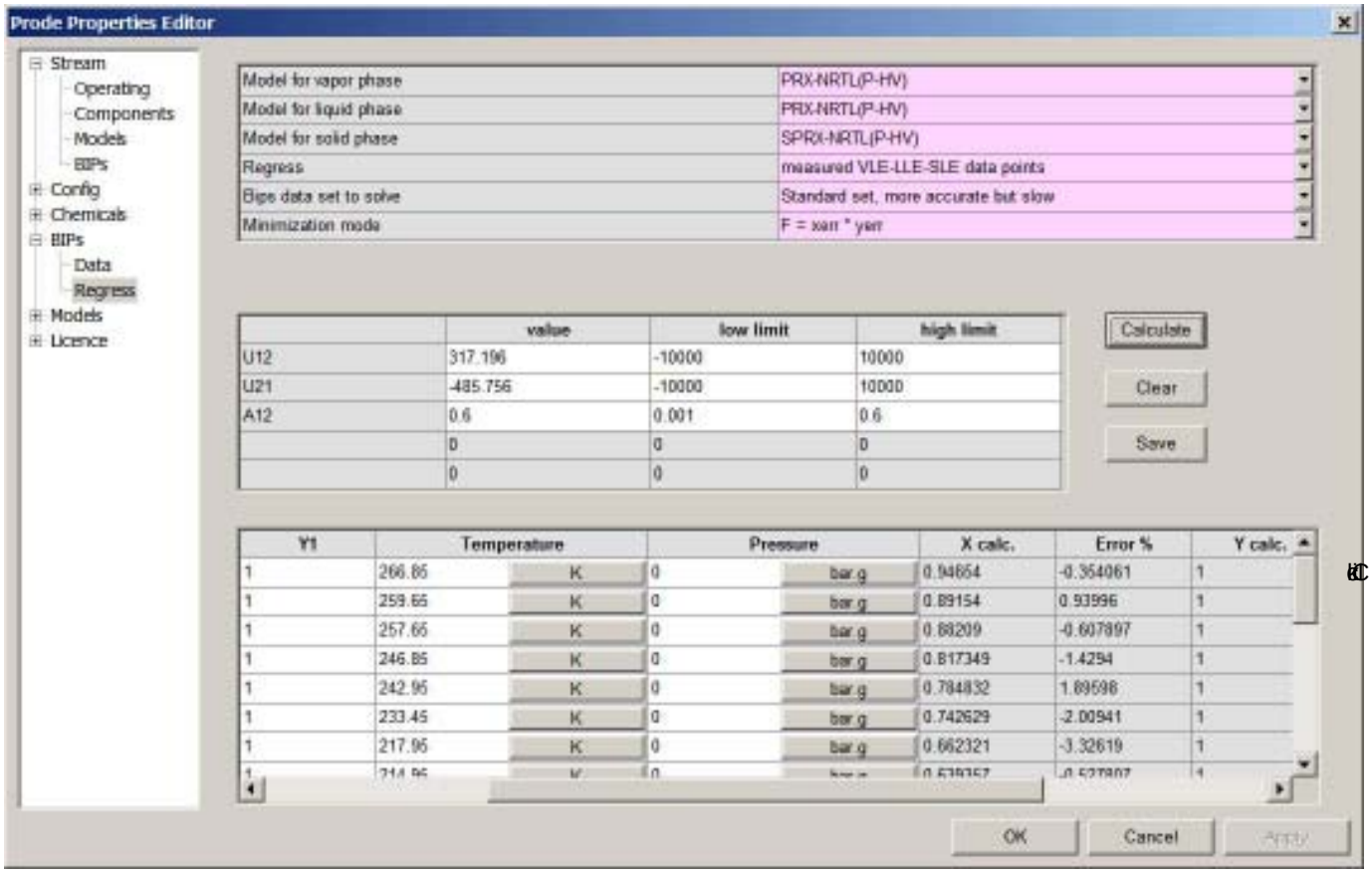
SLE data for Water-Methanol, atmospheric pressure
in this example first component is Water, second component Methanol
X1 is the water fraction in liquid phase while Y1 is solid fraction (we can set 1 for a solid pure model)

Point	X1	Y1	Temperature (K)	Pressure (Bar.g)
SLE	0.9432	1	266.85	0
SLE	0.9	1	259.65	0
SLE	0.87676	1	257.65	0
SLE	0.80583	1	246.85	0
SLE	0.8	1	242.95	0
SLE	0.728	1	233.45	0
SLE	0.641	1	217.95	0
SLE	0.636	1	214.95	0
SLE	0.6	1	208.15	0

as first step in BIPs->Data page you must select the components , first component is water, second component is methanol, then select the database (SLE BIPs) where to store calculated BIPs and the model PRX-NRTL-HV (Peng-Robinson + NRTL with Huron Vidal mixing rules)

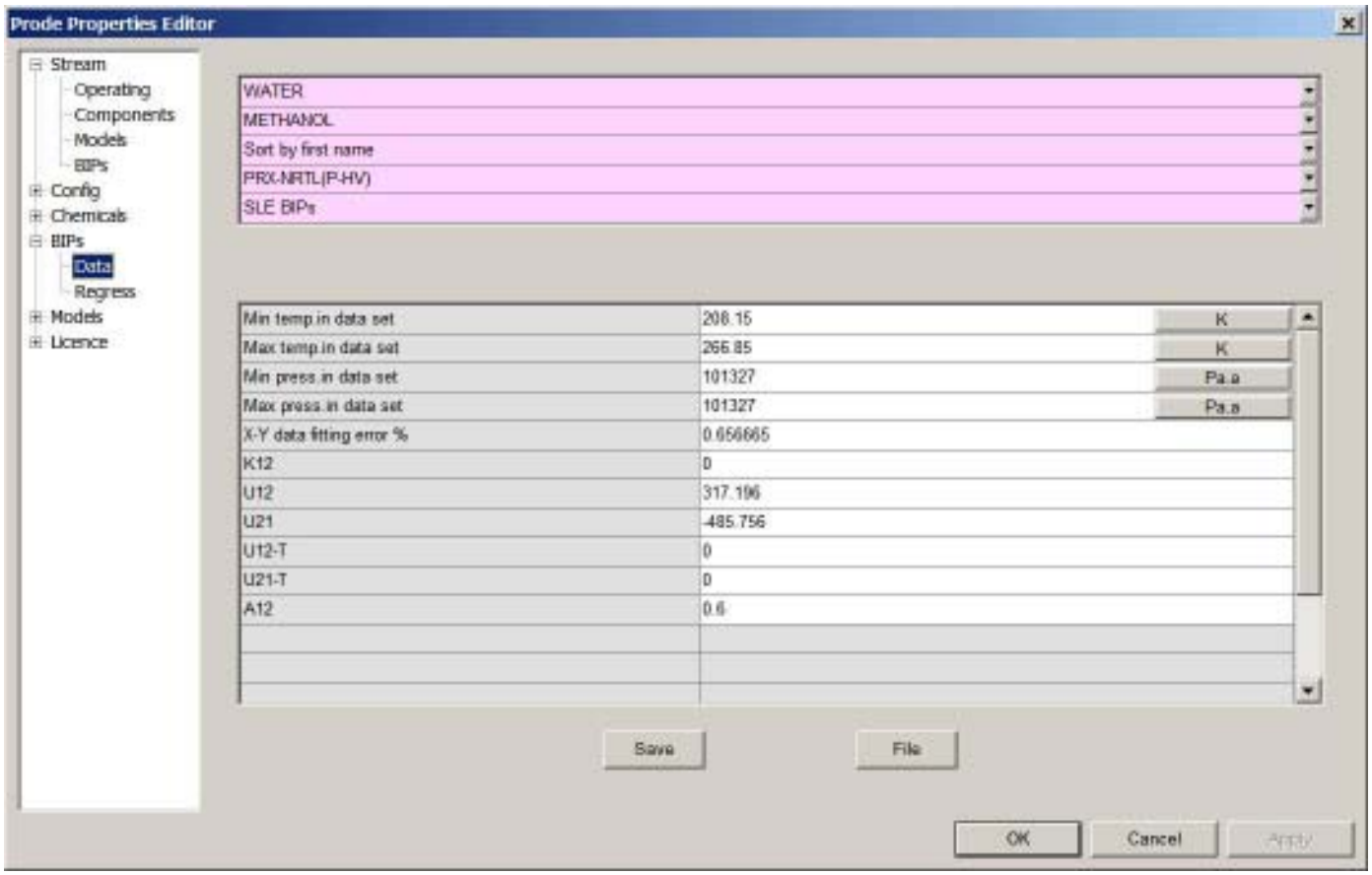


in BIPs->Regress page select models (PRX-NRTL-HV for vapor, PRX-NRTL-HV for liquid, SP-PRX-NRTL-HV for solid)
as for previous example select Standard as solution mode, which allows to calculate all the BIPs for the specified model, and enter all data points,
click on Calculate button to obtain the results



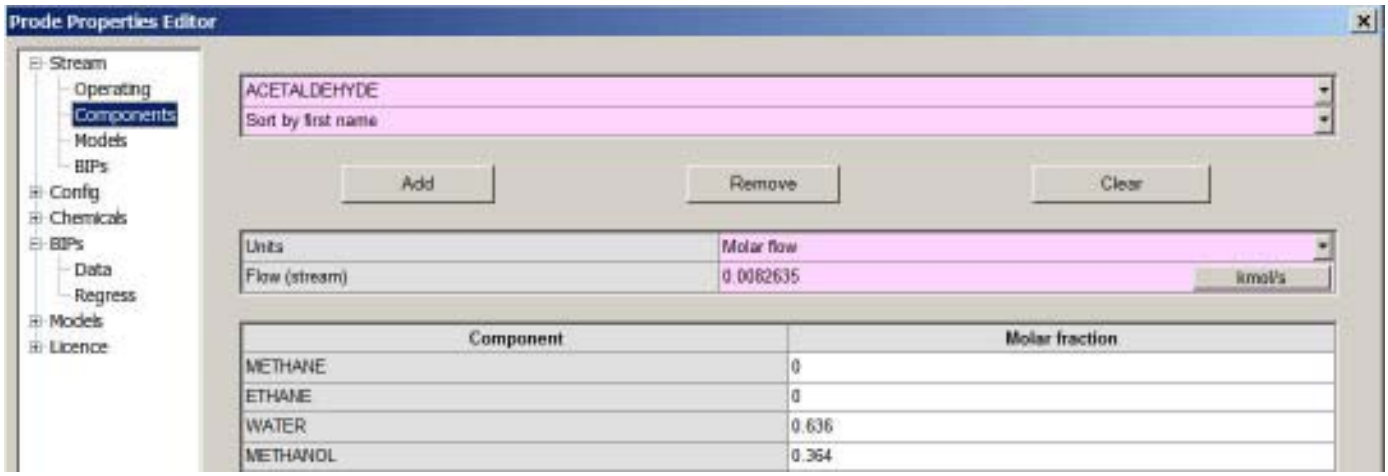
click on Save button to move calculated values in previous page

In BIPs->Data page verify values (see below) , select SLE BIPs Database and click on Save button to store data (you can store in file with the File button)

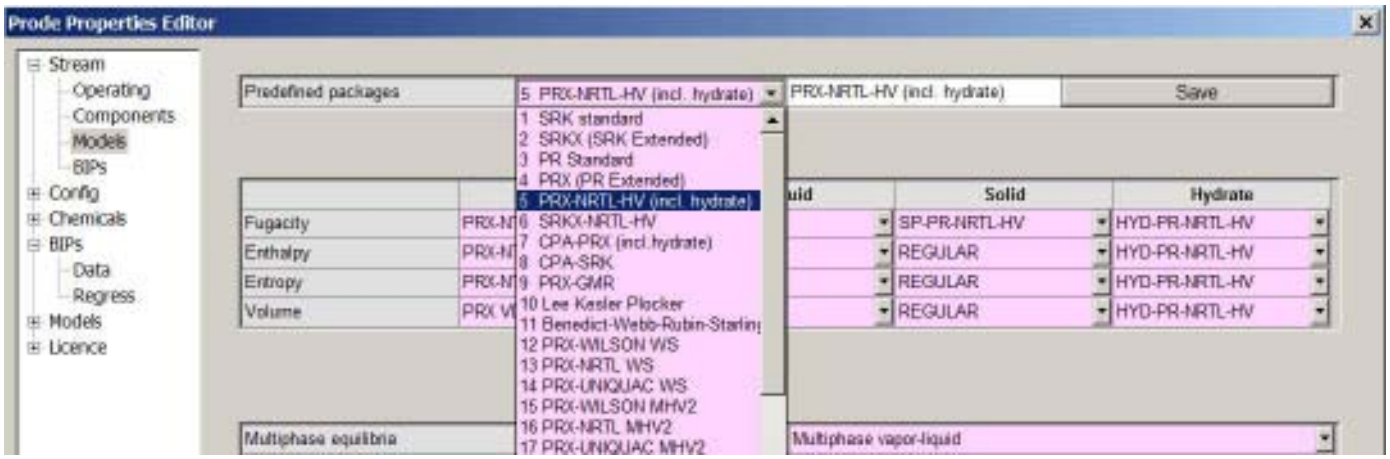


Now we wish to test the accuracy of calculated BIPs in estimating freezing point depression.
In Prode Properties Editor select stream 9 and define as composition C1 0 C2 0 CH4O 0.364 H2O 0.636 to test this point in the series

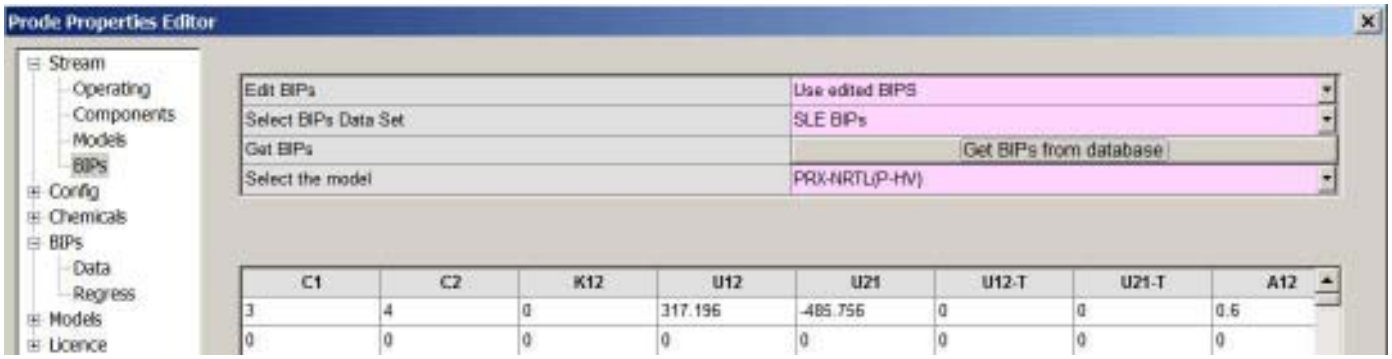
	X1	Y1	T(K)	P(Bar.g)
SLE	0.636	1	214.95	0



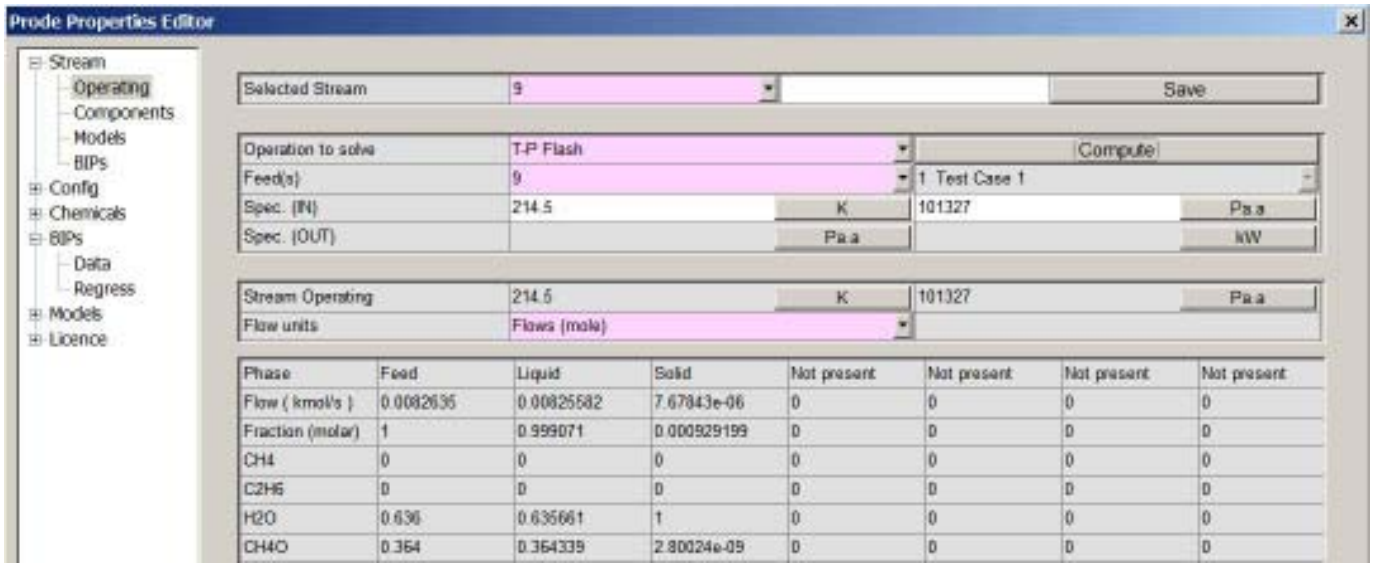
in Models tab select the same models adopted in data regression, (PRX-NRTL-HV for vapor, PRX-NRTL-HV for liquid, SP-PRX-NRTL-HV for solid) you may select the PRX-NRTL-HV package



In BIPs tab select SLE BIPs Data set (the database where calculated BIPs have been stored) and click on Load BIPs button to get BIPs, make sure that values shown on window are those previously calculated

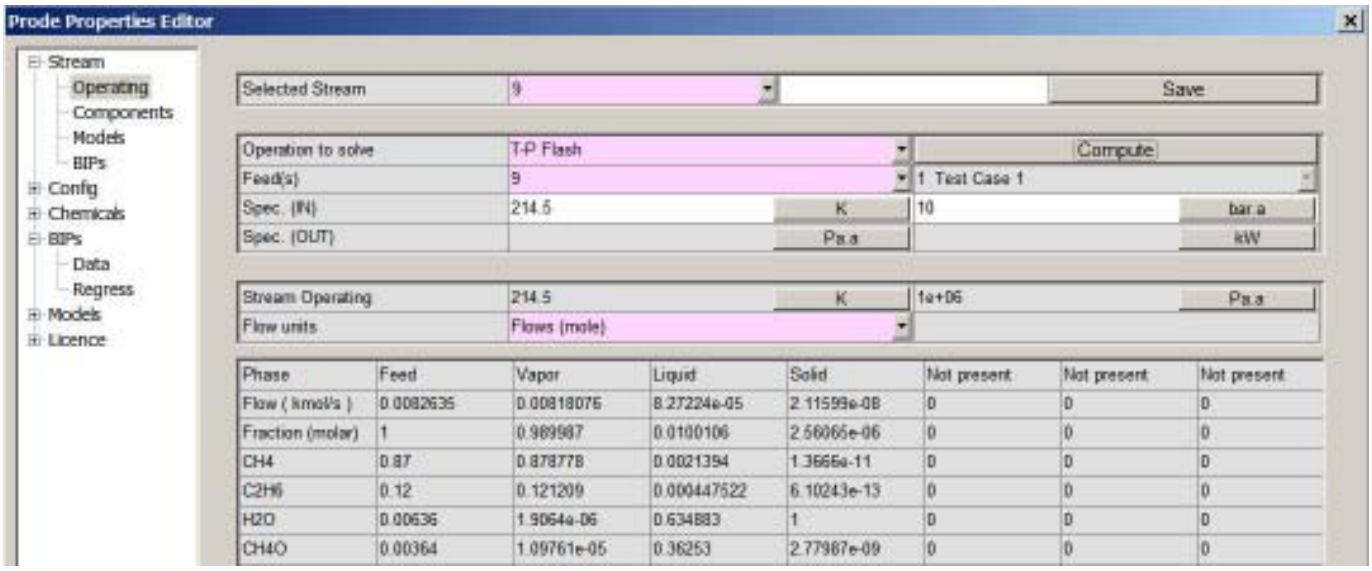


In Operating tab click Save Button to define the stream and make sure the list of components has been updated.
Now we can solve a TP-VLS flash operation to evaluate the point of incipient solidification



the calculated point is about 214.5 K ,
compare this value with the experimental data (214.95 K) utilized in data regression,
error is lower than 1 K

With the same method it is possibile to estimate solid formation for mixtures of different fluids, for example water and hydrocarbons with methanol as inhibitor, we repeat the example including C1 and C2 and keeping constant water / methanol fraction,
define the mixture C1 0.87 C2 0.12 H2O 0.00636 CH4O 0.00364 for this mixture the methanol fraction in water is equivalent to previous example, at 10 Bar the estimated point for solid (ice) formation is about 214.5 K (as previously calculated)

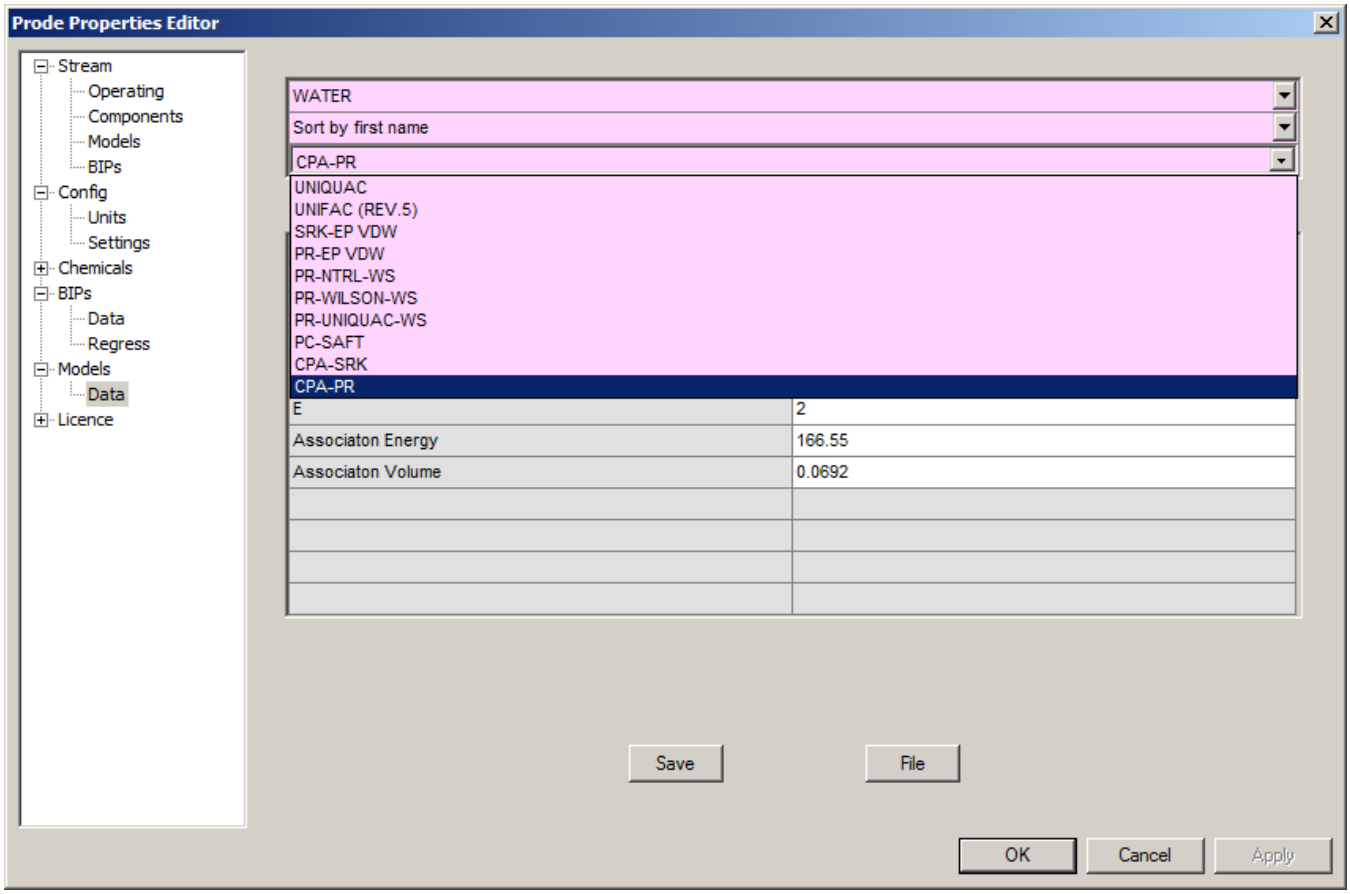


Note that we have not included BIPs for water-hydrocarbons and methanol-hydrocarbons, for accurate results you may wish to include BIPs for all pairs.
The PR-NRTL-HV model included in Prode Properties allows to enter (for each pair) a set of three BIPs (BIP1, BIP2, BIP3) or a single BIP (BIP4), when only single BIPs (BIP4), are included PR-NRTL-HV model gives the same results of PRX model.

Parameters of models

From this page you can :

- edit the parameters required by the different models available in library



Edit / modify data :

- select the components from the component's lists
- select the model
- edit / modify the parameters
- select the "Save" button to save the modified data (differently new data will be discharged)

IMPORTANT

Updating the file which contains the Model data :

this option permits to store all data into a file, differently all changes will be lost when leaving the application

- select the "File" button

CAUTION you may wish to create a backup of the file bips.dat before to overwrite the file

Accessing Prode Properties library

The technique for accessing the methods available in Prode Properties library will depend on which programming language You use. Languages such as FORTRAN , C, C++ or Microsoft NET (VB,C) exhibit differences in parameter passing in and out of functions. This may require you to adapt your code from the examples shown here. The calling convention determines how a program makes a call and where the parameters are passed. PROPERTIES does use of standard calls of Windows API, it pushes parameters on the stack, in reverse order. When accessing PROPERTIES You must also consider :

- Prode Properties real type is 8 bytes
- Prode Properties integer type is 4 bytes
- parameters are passed by value (with exception of strings which are arrays of characters)

IMPORTANT
C / C++ support files are located in the directory \Prode\C
FORTRAN support files are located in the directory \Prode\FORTRAN
Microsoft NET support files are located in the directory \Prode\NET
Microsoft EXCEL support files are located in the directory \Prode\Excel

Fortran

add ppp.lib file to the list of the files in Your project and include ppp.f90 to instruct the compiler about the methods available in Prode Properties then access the methods as they were included in your code

```
C   this procedure returns the critical temperature of a compound
INTERFACE TO REAL*8 FUNCTION TC ([C,ALIAS:'CompTc'] comp)
INTEGER*4 comp [VALUE]
END

REAL*8 tc
INTEGER*4 id
C   define the id value  here
tc = TC(id)
```

C / C++

- include the ppp.h header
- add ppp.lib file to the list of the files in Your project
- make sure you use the calling convention of ppp.h header file,
- access the Prode Properties methods

```
char *name;
name  = CompN(1);  // returns the name of the first component in the chemical's file
```

Microsoft NET (VB)

see the samples provided with Prode Properties for additional information

- include properties.vba to instruct the compiler about the methods available in Prode Properties and access the methods

```
CompName = MCompN(1)
```

Microsoft Excel

Microsoft Excel supports *macros* and VBA language for defining procedures; both of these can be used to access the functions in Prode Properties, see the samples provided with Prode Properties for additional information

- include properties.vba to instruct Microsoft Excel about the methods available in Prode Properties and access the methods

CompName = MCompN(1)

Translate resources to different languages

A large part of the resources are stored in the file res.lan , see the paragraph “**Data files folder**” for additional information about how to locate the file. The file res.lan is a text file, easily editable by the user.

Example

in English language
N2_NAME = “Nitrogen”;

in French language
N2_NAME = “Azote”;

in Italian language
N2_NAME = “Azoto”;

IMPORTANT

When editing a string take care to modify only the parts enclosed within the braces ”” and do not alter/modify the data structures composed by special characters as for example ::

Microsoft Applications and Strings

Prode Properties utilizes the standard API calling convention for Microsoft Windows applications. This assures that almost all Windows compatible applications which support DLLs will also support Prode Properties. There are, however, some exceptions in passing strings (arrays of characters) since Microsoft utilizes proprietary data formats. Prode Properties includes Microsoft specific methods in addition to the standard methods supporting the ANSI C standard, Microsoft specific methods are compatible with almost all Microsoft applications.

Define models, compatibility with old verions

Prode Properties includes many methods for defining (via software) the thermodynamic models and the related options, see the paragraph “Methods to set / access different options” for additional information

Units of measurement

Prode Properties allows to define via software the units of measurement, there are methods for defining the units and methods for retrieving codes and strings, see paragraph “Methods for accessing / defining the units of measurement” and the samples provided with the package for additional information, in Prode Properties to reference a unit must use a numeric code

QUANTITY	UNIT	CODE	DEFAULT UNIT
Pressure (abs)	CONV_P	15	“Pa.a”
Pressure (rel)	CONV_DP	16	“Pa”
Temperature (abs)	CONV_T	17	“K”
Temperature(rel)	CONV_DT	18	“K”
Calorific Value (weight)	CONV_HM	19	“Kj/Kg”
Calorific Value (molar)	CONV_HMM	20	“Kj/Kmol”
Power	CONV_HS	21	“KW”
Entropy (Streams)	CONV_SS	22	“KJ/(K*s)”
Heat Capacity (weight)	CONV_CP	23	“kJ/(kg*K)”
Heat Capacity (molar)	CONV_CPM	24	“kJ/(kmol*K)”
Flow (mass basis)	CONV_W	25	“Kg/s”
Flow (gas, mass basis)	CONV_WG	26	“Kg/s”
Density (weight)	CONV_D	27	“Kg/m3”
Density (molar)	CONV_DM	28	“Kmol/m3”
Specific Volume (weight)	CONV_SV	29	“m3/Kg”
Specific Volume (molar)	CONV_SVM	30	“m3/Kmol”
Thermal Conductivity	CONV_TC	31	“W/(m*K)”
Viscosity (dynamic)	CONV_V	32	“Pa*s”
Surface Tension	CONV_ST	33	“N/m”
Lenght	CONV_L	34	“m”
Area	CONV_A	35	“m2”
Volume	CONV_VOL	36	“m3”
Mass	CONV_M	37	“Kg”
Velocity	CONV_VL	38	“m/s”
Acceleration	CONV_ACC	39	“m/s2”
Force	CONV_FOR	40	“N”
Time	CONV_TM	41	“s”
Heat Flux	CONV_HF	42	“KW/m2”
Thermal Resistance	CONV_TR	43	“K*m2/KW”
Heat Transfer Coefficent	CONV_HTC	44	”KW/(m2*C)”
Flow (volume basis)	CONV_VW	45	“m3/s”
Viscosity (kinematic)	CONV_VK	46	“m2/s”
Energy	CONV_EN	47	“KJ”
Dipole moment	CONV_EDM	48	“c-m”
Solubility parameter	CONV_SP	49	“(J/m3)^1/2”
Flow Coefficient	CONV_CV	50	“Cv”
Compressibility coefficient	CONV_CC	51	“1/Pa”
Joule Thomson coefficient	CONV_JTC	52	“K/Pa”
Flow (molar basis)	CONV_WM	53	“Kmol/s”
Volume expansivity	CONV_VE	54	“1/K”

Introducing Prode Properties library methods

Prode Properties library includes a range of methods to deal with problems in chemical engineering and to achieve tight control over the calculations .

A non-inclusive list would include

- Thermodynamic calcs (flash operations, enthalpy, entropy, volume, energy, unit operations)
- Streams data access and calcs (set and retrieve operating conditions, critical and transport properties calcs)
- Chemicals library access (retrieve data from chemicals file)
- Error messages (management of errors messages)

Methods for thermodynamic calc' s

Prode Properties includes a complete set of methods for solving all the standard flash operations with specified final temperature or pressure and entropy or enthalpy or volume or energy basis, phase fraction with temperature or pressure basis plus mixers, dividers, gas,liquid phase separation operations etc.

integer result = setOp(integer stream, double t, double p)
Given a stream, operating pressure and temperature, performs an isothermal flash and sets operating conditions.

integer result = setSOp(integer stream)
Given a stream performs an isothermal flash at (user defined) standard conditions.

double t = PfPF(integer stream, double p, double pf, int state, int n)
Given a stream, the pressure , phase fraction (range 0-1), state (gas, liquid, solid) and position n calculates and returns the nth (n : 1-5) equilibrium temperature along the specified phase fraction line

double p = PfTF(integer stream, double t, double pf, int state, int n)
Given a stream, the temperature , phase fraction (range 0-1), state (gas, liquid, solid) and position n calculates and returns the nth (n : 1-5) equilibrium pressure along the specified phase fraction line

double t = LfPF(integer stream, double p, double lf)
Given a stream, the pressure and Liquid fraction (range 0-1) calculates and returns the first equilibrium temperature along the specified phase fraction line

double p = LfTF(integer stream, double t, double lf)
Given a stream, the temperature and Liquid fraction (range 0-1) calculates and returns the first equilibrium pressure along the specified phase fraction line

double t = HPF(integer stream, double p, double h, double et)
Given a stream, final pressure, the required (final) enthalpy (see the method StrH() for the definition) and a estimated value for final temperature (or 0 for automatic estimate), method solves the flash operation (enthalpy basis) and returns final temperature

double p = HTF(integer stream, double t, double h, double ep)
Given a stream, final temperature, the required (final) enthalpy (see the method StrH() for the definition) and a estimated value for final pressure (or 0 for automatic estimate), method solves the flash operation (enthalpy basis) and returns final pressure

double t = SPF(integer stream, double p, double s, double et)
Given a stream, final pressure, the required (final) entropy (see the method StrS() for the definition) and a estimated value for final temperature (or 0 for automatic estimate), method solves the flash operation (entropy basis) and returns final temperature.

double p = STF(integer stream, double t, double s, double ep)
Given a stream, final temperature, the required (final) entropy (see the method StrS() for the definition) and a estimated value for final pressure (or 0 for automatic estimate), method solves the flash operation (entropy basis) and returns final pressure.

double t = VPF(integer stream, double p, double v, double et)
Given a stream, final pressure, the required specific volume (see the method StrV() for the definition) and a estimated value for final temperature (or 0 for automatic estimate), method solves the flash operation (volume basis) and returns final temperature.

double p = VTF(integer stream, double t, double v, double ep)
Given a stream, final temperature, the required specific volume (see the method StrV() for the definition) and a estimated value for final pressure (or 0 for automatic estimate), method solves the flash operation (volume basis) and returns final pressure.

integer result = HVF(integer stream, double h, double v, double et, double ep)
Given a stream, the required (final) enthalpy (see the method StrH() for the definition) the required (final) specific volume (see the method StrV() for the definition) and estimated values for final temperature and pressure (or 0 for automatic estimate), method solves the flash operation

integer result = SVF(integer stream, double s, double v, double et, double ep)
Given a stream, the required (final) entropy (see the method StrS() for the definition) the required specific volume (see the method StrV() for the definition) and estimated values for final temperature and pressure (or 0 for automatic estimate), method solves the flash operation

integer result = HSF(integer stream, double h, double s, double et, double ep)
Given a stream, the required (final) enthalpy (see the method StrH() for the definition) the required (final) entropy (see the method StrS() for the definition) and estimated values for final temperature and pressure (or 0 for automatic estimate), method solves the flash operation

double t = EPF(integer stream, double p, double E, double aout, double et)
Given a stream, final pressure, outlet area, the term E (equal to Hin + 1/2Vin^2) and a estimated value for final temperaure (or 0 for automatic estimate) method solves the constant energy flash and returns final temperature.
Hin + 1/2Vin^2 = Ho + 1/2Vo^2
this method permits to model adiabatic, irreversible expansions when the contribute of kinetic energy cannot be neglected.

integer result = MixF(integer stream1, integer stream2, double et)
Given two streams, stream1 and stream2 and a estimated value for final temperature (or 0 for automatic estimate) method solves a mixer operation and returns the result on stream1, the feed streams are adiabatically flashed to the lowest inlet stream pressure

integer result = Divi (integer stream1, integer stream2, double wdiv)
Given two streams (stream1 and stream2) and a flowrate fraction (0-1) performs a divider operation so that stream 1 is shifted into two streams (stream1, stream2) of the same composition, temperature and pressure, flowrate fractions are subdivided as specified by wdiv (stream2 = wdiv, stream1 = 1- wdiv)

integer result = psep(integer stream1, integer stream2, integer phase)
Given a stream (stream1) performs an isothermal flash to simulate a phase type (vapor,liquid,solid) separator and returns the result as stream2.

Methods for stream' s data access

Prode Properties includes a set of functions for accessing stream parameters and calculating transport properties. Note that when calculating transport properties the program performs a VLE flash and returns 0 (zero value) when no associated liquid or gas phase is found.

integer res = isSDef(integer stream)
given a stream returns TRUE (integer = 1) if stream has been defined, otherwise returns FALSE (0)

double t = getT(integer stream)
given a stream returns stream's operating temperature

double p = getP(integer stream)
given a stream returns stream's operating pressure

integer nr = getPNr()
returns the maximum number of phases that procedure can detect

integer type = StrPt(integer stream, int phase)
given a stream and position in range 1- getPNr() returns the phase type (vapor,liquid,solid)

char *description = StrPts(integer stream, int phase)
given a stream and position in range 1- getPNr() returns a ANSI C string with the description (vapor, liquid, solid...)

int description MStrPts(integer stream, int phase, char *s, integer slm)
given a stream and position in range 1- getPNr() fills string s with the description (vapor, liquid, solid...) (eventually truncated to slm maximum lenght), this is the Microsoft Excel specific method

double lf = StrLf(integer stream)
given a stream returns the total liquid fraction (molar basis) in stream

double pf = StrPf(integer stream, integer phase)
given a stream and phase position in range 1- getPNr() returns the phase fraction

double w = getW(integer stream, integer phase, integer pos.)
given a stream, the phase position and component's position (in component's list) returns the component molar fraction in that phase

double Zi= getZ(integer stream, integer pos.)
given a stream and component's position (in component's list) returns the comp's Z (weight percentage, molar basis)

integer res = putZ(integer stream, integer pos., double Zi)
given a stream, comp's position and Z , sets the comp's pos. in Z vector (composition, molar basis) for that stream

integer nr = getCNr(integer stream)
given a stream returns the number of components defined in that stream

integer nr = getMCNr()
returns the maximum number of components in a stream

double zv = StrZv(integer stream)
given a stream returns the relevant compressibility factor (gas phase)

double mw = StrMw(integer stream)
given a stream returns the averaged molecular weight (all phases)

double v = StrV(integer stream)
given a stream returns the specific volume as sum of specific volumes of all phases

double mw = StrGMw(integer stream)
given a stream returns the averaged molecular weight (gas phase)

double mw = StrLMw(integer stream)
given a stream returns the averaged molecular weight (liquid phase)

double h = StrH(integer stream)
given a stream returns the total (stream) enthalpy (gas + liquid + solid phases)

double h = StrGH(integer stream)
given a stream returns the total (stream) enthalpy (gas phase)

double h = StrSGH(integer stream)
given a stream returns the specific (unit weight) enthalpy (gas phase)

double h = StrLH(integer stream)
given a stream returns the total (stream) enthalpy (liquid phase)

double h = StrSLH(integer stream)
given a stream returns the specific (unit weight) enthalpy (liquid phase)

double h = StrSH(integer stream)
given a stream returns the total (stream) enthalpy (solid phase)

double h = StrSSH(integer stream)
given a stream returns the specific (unit weight) enthalpy (solid phase)

double cp = StrGICp(integer stream)
given a stream returns the ideal gas heat capacity

double cp = StrGCp(integer stream)
given a stream returns the specific heat capacity (constant pressure, gas phase)

double cv = StrGCv(integer stream)
given a stream returns the specific heat capacity (constant volume, gas phase)

double cp = StrLCp(integer stream)
given a stream returns the specific heat capacity (constant pressure, liquid phase)

double cv = StrLCv(integer stream)
given a stream returns the specific heat capacity (constant volume, liquid phase)

double cp = StrSCp(integer stream)
given a stream returns the specific heat capacity (constant pressure, solid phase)

double ss = StrMSS(integer stream)
given a stream returns the speed of sound (gas, liquid) as calculated with HEM model for mixed phases

double ss = StrGSS(integer stream)
given a stream returns the speed of sound in gas phase

double ss = StrLSS(integer stream)
given a stream returns the speed of sound in liquid phase

double jt = StrGJT(integer stream)
given a stream returns the Joule Thomson coefficient in gas phase

double jt = StrLJT(integer stream)
given a stream returns the Joule Thomson coefficient in liquid phase

double ic = StrGIC(integer stream)
given a stream returns the isothermal compressibility coefficient - (1 / V) * dV / dP in gas phase

double ic = StrLIC(integer stream)
given a stream returns the isothermal compressibility coefficient - (1 / V) * dV / dP in liquid phase

double v = StrGVE(integer stream)
given a stream returns the volumetric expansivity coefficient - (1 / V) * dV / dT in gas phase

double ic = StrLVE(integer stream)
given a stream returns the volumetric expansivity coefficient - (1 / V) * dV / dT in liquid phase

double s = StrGS(integer stream)
given a stream returns the total (stream) entropy (gas phase)

double s = StrSGS(integer stream)
given a stream returns the specific (unit weight) entropy (gas phase)

double s = StrLS(integer stream)
given a stream returns the total (stream) entropy (liquid phase)

double s = StrSS(integer stream)
given a stream returns the total (stream) entropy (solid phase)

double s = StrSLS(integer stream)
given a stream returns the specific (unit weight) entropy (liquid phase)

double s = StrSSS(integer stream)
given a stream returns the specific (unit weight) entropy (solid phase)

double s = StrS(integer stream)
given a stream returns the total (stream) entropy (gas + liquid + solid phases)

integer res = setWm(integer stream, double W)
given a stream and flow (mass basis), sets the flow

double w = getWm(integer stream)
given a stream returns the flow specified for that stream.

double hc = StrHC(integer stream)
given a stream returns the calculated net heat of combustion (gas phase).

double fl = StrFML(integer stream)
given a stream returns the calculated flammability lean limit (gas phase).

double fl = StrFMH(integer stream)
given a stream returns the calculated flammability rich limit (gas phase).

double d = StrLD(integer stream)
given a stream returns the calculated liquid density (at operating conditions).

double d = StrGD(integer stream)
given a stream returns the calculated gas density (at operating conditions).

double tc = StrLC(integer stream)
given a stream returns the calculated liquid thermal conductivity (at operating conditions).

double tc = StrGC(integer stream)
given a stream returns the calculated gas thermal conductivity (at operating conditions).

double v = StrLV(integer stream)
given a stream returns the calculated liquid viscosity (at operating conditions).

double v = StrGV(stream)
given a stream returns thecalculated gas viscosity (at operating conditions).

double st = StrST(integer stream)
given a stream returns the calculated surface tension (at operating conditions).

Integer cpnr = StrCPnr(integer stream)
given a stream returns the number of critical points detected and calculated, to get a critical point use the methods StrPc() and
StrTc() setting value of pos in the range 1-cpnr

double p = StrPc(integer stream, Integer pos)
given a stream and the critical point position in the list (see method StrCPnr()) returns the critical pressure

double t = StrTc(integer stream, Integer pos)
given a stream and the critical point position in the list (see method StrCPnr()) returns the critical temperature.

double p= StrCBp(integer stream)
given a stream returns the cricodenBar pressure.

double t= StrCBt(integer stream)
given a stream returns the cricodenBar temperature.

double p= StrCTp(integer stream)
given a stream returns the cricodenTherm pressure.

double t= StrCTt(integer stream)
given a stream returns the cricodenTherm temperature.

double ac = StrAc(integer stream)
given a stream returns the acentric factor (mole fraction average).

double p= StrRVP(integer stream, integer mode)
given a stream returns the Reid vapor pressure
mode = 1 simulation of D6377 procedure (liquid not saturated with air)
mode = 2 simulation of D323 procedure (liquid saturated with air)

double fp = StrFLP(integer stream)
given a stream returns the Flash point (for pure fluids the method returns the value stored in databank while for mixtures the flash point is calculated by a iterative procedure where VLE is solved according the selected models for stream)

Methods for stream' s definition

Prode Properties includes a set of functions to define a stream by program (as alternative to utilize the Properties Editor)

- to create a NEW list of components, call initS() ant define the list of components with putCC()
- define the mole fraction of each component with putZ()
- call setS() to define the stream
- call setW() to define the flow
- utilize the methods described in paragraph "Methods to define thermodynamic models" to define the models
- call loadSB() to load the BIPs from database or define specific BIPs with methods PutCi(), PutCj(), PutMB(), PutBIP()

integer res = initS (integer stream)
given a stream initializes all data, call this method before to create a new list of components.

integer res = putCC (integer stream, integer pos, integer compcode)
given a stream, component's position (in component's list) and component code sets the code in component's list.

integer res = putZ(integer stream, integer pos., double Zi)
given a stream, comp's position and Z , sets the comp's pos. in Z vector (composition, molar basis) for that stream

integer res = setS(integer stream)
given a stream performs a sequence of validating operations on data. This method must be called after to have restored stream's data from archives (files etc.)Methods to define a initial condition for a stream

nteger res = loadSB(integer stream, integer btype)
given a stream loads all BIP available in database. This method must be called after the stream has been defined since it requires the list of components. Codes for btype are 0 for VLE, 1 for LLE, 2 for SLE, 3 for Hydrates

double Zi= getZ(integer stream, integer pos.)
given a stream and component's position (in component's list) returns the comp's Z (molar fraction)

integer cc = getCC(integer stream, integer pos)
given a stream and component's position (in component's list) returns the component code (a integer that identifies the component in chemical's file).

integer nr = getMBPNr()
returns the maximum number of (interaction coefficients) binary pairs in a stream

int ci = getCi(integer stream, integer pos)
given a stream and position (in interaction's coeff. list) returns the first component reference (a integer that identifies the component in component's list).

integer res = PutCi (integer stream, integer pos, integer ci)
given a stream, position (in interaction coefficients list) and first component reference sets the component's reference in interaction coefficient's list.

int cj = getCj(integer stream, integer pos)
given a stream and position (in interaction's coeff. list) returns the second component reference (an integer that identifies the component in component's list).

integer res = PutCj (integer stream, integer pos, integer cj)
given a stream, position (in interaction coefficients list) and second component reference sets the component's reference in interaction coefficient's list.

int model = getMB(integer stream, integer pos)
given a stream and position (in interaction's coeff. list) returns the related model (an integer that identifies the model).

integer res = PutMB(integer stream, integer pos, integer model)
given a stream, position (in interaction coefficients list) and a model identifier sets the model in interaction coefficient's list.

double BIP = getBIP(integer stream, integer pos, integer id)
given a stream, position (in binary coeff. list) and BIP identifier (0-max nr. of BIPs for that model) returns BIP.

integer res = PutBIP(integer stream, integer pos, integer id. double Kji)
given a stream, position (in binary coeff. list) BIP identifier (0-max nr. of BIPs for that model) and value stores BIP in that position of the list.

Methods to define stream's operating conditions

Prode Properties includes a set of functions to define the intial (operating) condition of a stream, these can be utilized as alternative to the standard initialization via setOp() method

- call rstValidSop()
- define compostion in phase 1 with putW()
- define phase fraction with putPF()
- define phase type with putPT()
- set phase 1 as valid , setValidPhase()
- continue with another phase (2...n)
- define temperature with putT()
- define pressure with putP()
- set conditions as valid with setValidSop()

integer result = rstValidSop(integer stream)
Given a stream clears the compostions of different phases at operating conditions

integer result = setValidSop(integer stream)
Given a stream sets the compostions of different phases at operating conditions.as valid.

integer result = setValidPhase(integer stream, integer phase)
Given a stream and phase sets the phase compostion.as valid.

integer result = putW(integer stream, integer phase, int compnr, double w)
Given a stream, phase, component number and component's molar fraction in that phase stores the value

integer result = putPF(integer stream, integer phase, double fraction)
Given a stream, phase and phase fraction stores the phase .fraction value

integer result = putPT(integer stream, integer phase, int type)
Given a stream, phase and phase type (vapor,liquid,solid) stores the phase type

nteger result = putT(integer stream, double t)
Given a stream and operating temperature stores the value

nteger result = putP(integer stream, double p)
Given a stream and operating pressure stores the value

Copy of streams

to make a copy of a stream utilize the method

integer res = StrCopy(integer stream1, integer stream2)
Given two streams (stream1 and stream2) copies the stream 2 into stream 1

Methods for solving staged columns

Note : this method utilizes the standard ANSI C convention for exchanging parameters (see the samples provided with the software)

Properties includes a procedure for solving staged columns, the column is modeled with stgnr equilibrium stages, column may include a condenser and a rebolier, stage numbering is bottom up, the bottom stage (reboiler, if specified) is number one and the top stage (condenser, if specified) is number stgnr
There may be one or more feeds, a feed is modeled by entering liquid on the specified stage and vapor portion to the stage above (with exception of top stage).
There may be one or more side streams
Heat added/removed on each stage can be specified
Efficiency parameter on each stage can be specified

integer res = DCOL(int csep, int stgnr, int init, double *stgt,double *stgp,double *stgef,double *stgdH, int prod_h, int btm_h,int fnr,int *fstr,int *fpos,int snr,int *sstr,int *spos,int *sft, double *sflow,int vnr,double *vrv,int *vtype,int *ptype,int *piv,double *prv, double*flows)

Parameters :

csep	(int)	column type : 1 VLE , 2 VLLE , 3 LLE (some features available in extended versions)
stgnr	(int)	number of stages
init	(int)	0 for automatic initialization, 1 temperatures and flows are defined by user
stgt	(double*)	vector (stgnr) with stage temperatures
stgp	(double*)	vector (stgnr) with specified stage pressures
stgef	(double*)	vector (stgnr) with specified stage efficiency, permitted range 0,1-1
stgdH	(double*)	vector (stgnr) with specified dH (heat added, removed)
prod_h	(int)	stream for top product/distillate
btm_h	(int)	stream for bottom product
fnr	(int)	number of feeds
fstr	(int*)	vector (fnr) with the feeding streams
fpos	(int*)	vector (fnr) with feeds positions 1-stgnr
snr	(int)	number of side streams
sstr	(int*)	vector (snr) with the list of side streams
spos	(int*)	vector (snr) with side streams positions (1-stgnr)
sft	(int*)	vector (snr) with specified flow type (GAS_PHASE, LIQ_PHASE, see Codes used in Prode library)
sflow	(double*)	vector (snr) with the specified (on each side stream) side product to feed flow ratio
vnr	(int)	number of variables to solve
vtype	(int*)	vector (vnr) with type of variable (seebelow)
vrv	(double*)	vector (vnr) with calculated values for variable
ptype	(int*)	vector (pnr) with type of specification (see below)
piv	(int*)	vector (pnr) with integer values as the position of components in the list
prv	(double*)	vector (pnr) with values of the specifications to solve
flows	(double*)	vector with calculated values for vapor/liquid flows in all stages, dimension nrphases*nrc*stgnr when a condenser is present the reflux is the liquid flow on top stage

Codes for variables

reboiler	1
total condenser	2
partial condenser	3

Codes for specifications

reflux ratio	1
product to feed ratio (molar fract.)	2
bottom to feed ratio (molar fract.)	3
component (molar fract.) in top product	4
component (molar fract.) in bottom product	5
component recovery in top product	6
component recovery in bottom product	7

Notes :
When passing / returning paramenters the first element in vectors is the element 0
Main variables (1-vnr) are (when specified) reboiler and condenser (partial or total), each variable (of type defined in vtype) requires a suitable specification (in ptype, piv, prv), usually for reboiler the specification is the product to feed ratio and for a condenser the reflux ratio, but specifications based on component's fractions on top and bottom products are permitted, in these cases specify in piv the position of selected component in the list and in prv the value of the fraction required

Secondary variables are side streams (1-snr), each side stream (defined in sstr, spos) requires (in sflow) a specification for the side product to (total) feed flow ratio.
The column is modeled with thermodynamics and options defined for the first feed in the list.

Initialization
in most cases the procedure doesn't require to initialize values, when required set the variable init to 1 and define the proper initial values in vectors stgt and flows, note that in a sequence of similar operations (for example when controlling the operating point of a column) it may result useful to reintroduce the calculated values as starting point for the new calculus

Examples

Column with 8 stages, 1 feed (stage 4), pressure reboiler 12.5 Bar, pressure top 12 Bar, efficiency 1, dH = 0

```
csep            = 1
stgnr           = 8
init            = 0
stgp[0]         = 12.5
......
stgp[7]         = 12
stgef[0]        = 1
......
stgef[7]        = 1
stgdH[0]        = 0
......
stgdH[7]        = 0
prod_h          = stream1
btm_h           = stream2
fnr             = 1
fstr            = stream3
fpos            = 4

variables       : reboiler and total condenser
specifications  : component 2 fraction in top product and bottom product to to feed ratio

vnr             = 2
vtype[0]        = 1
ptype[0]        = 4
piv[0]          = 2
prv[0]          = 0.96
vtype[1]        = 2
ptype[1]        = 3
piv[1]          = 0
prv[1]          = 0.4


variables       : reboiler , partial condenser and 2 side streams (liquid and gas phases)
specifications  : component 2 fraction in top product , bottom product to feed ratio, side streams flow to feed ratio

vnr             = 2
vtype[0]        = 1
ptype[0]        = 4
piv[0]          = 2
prv[0]          = 0.96
vtype[1]        = 3
ptype[1]        = 3
piv[1]          = 0
prv[1]          = 0.4

snr             = 2
sstr[0]         = stream4
spos[0]         = 4
sft[0]          = LIQ_PHASE
sflow[0]        = 0.12
sstr[1]         = stream5
spos[1]         = 7
sft[1]          = GAS_PHASE
sflow[1]        =  0.078
```

Methods for Reactors

Note : this method utilizes the standard ANSI C convention for exchanging parameters (see the samples provided with the software)

simulation of reactors

int res = REACT(int streamIn, streamOut, int model, int NrReactions, double **Conv, double Pout, double dHeat)

Parameters :

streamIn (int)		inlet stream
streamOut	(int)	outlet stream
model	(int)	model for reactor (see below)
NrReactions	(int)	number of reactions
Conv	(double**)	matrix (NrComponents, NrReactions) to specify reactions
Pout	(double)	output pressure
dHeat	(double)	heat added, removed

Codes for models

Gibbs	1
Equilibrium Reactor	2

additional models on request

Methods for fluid flow problems

simulation of single phase, two-phases, multiphase flow on circular pipes

int res = PIPE(int stream, int model, double diam, double rough, double length, double dHeight, double dHeat)

Parameters :

stream	(int)	inlet stream
model	(int)	model for fluid flow and phase equilibria (see below)
diam	(double)	pipe internal diameter
rough	(double)	parameter defining relative pipe roughness
length	(double)	lenght of this segment
dHeight	(double)	height difference (inlet, outlet)
dHeat	(double)	heat added, removed

Codes for models

Beggs & Brill / Hazen-Williams / AGA 1

additional models on request

Methods for Hydrates phase equilibria

methods for calculating hydrate formation pressure (or temperature)

double p = HPFORM(int stream, double t, int method)
double t = HTFORM(int stream, double p, int method)

Parameters :

stream (int)	inlet stream
t (double)	operating temperature (or operating pressure)
method (int)	1 = include SI , SII , SH
	2 = SI
	3 = SII

Methods for solving a Polytropic operation

Rigorous polytropic stage (to model compressors and expanders)

double val = PSPF(int stream, double pout, int model, double param)

Parameters :

stream (int)	inlet stream
pout (double)	outlet pressure
model (int)	model, see below codes 1-4
param (double)	for model 1 and 3 specified polytropic efficiency (range 0-1)
	for model 2 and 4 (measured) outlet temperature

the procedure can model compression and expansion units such as centrifugal compressors, expansion turbines etc.
the procedure returns

-calculated temperature	options 1,3
-calculated efficiency	options 2,4

models available (**)

1	given initial condition, pout and polytropic efficiency calculates outlet condition, integration of a 4 poles Z function, method applicable to gas phase only, derived from Huntington “Another New Look at Polytropic Calculation Methods for Turbomachinery performance”,
2	given initial condition, pout and tout calculates polytropic efficiency, integration of a 4 poles Z function, method applicable to gas phase only, derived from Huntington “Another New Look at Polytropic Calculation Methods for Turbomachinery performance”,
3	given initial condition, pout and polytropic efficiency calculates outlet condition R.Paron “Polytropic solution with phase equilibria” method applicable to gas and mixed (gas + liquid) phases
4	given initial condition, pout and tout calculates polytropic efficiency, R.Paron “Polytropic solution with phase equilibria” method applicable to gas and mixed (gas + liquid) phases

(**) additional models available from Prode

Methods for orifices, relief valves design / rating

This unit models a relief valve with different methods

double area = ISPF(int stream, double pout, int model, double *param)

Parameters :

stream (int)	inlet stream
pout (double)	outlet pressure
model (int)	model, see below codes 1-4
param(double)	correction parameter, see below the range of values

the procedure models a relief valve at specified operating conditions and returns the calculated area

models available (**)

1	HEM Homogeneous Equilibrium (Solution of Mass Flux integral)
2	HNE Homogeneous Non-equilibrium (HEM with Boling Delay and Gas-Liquid Slip Contributes)
3	HNE-DS, Homogeneous Non-equilibrium
4	NHNE Non-homogeneous Non-equilibrium

model	recommended range of values for correction parameter
HEM	not required
HNE	0.7-0.8 for safety valves
HNE-DS	see the paper
NHNE	0.7-0.8 for safety valves

(**) additional models available from Prode

Methods for calculating equilibrium lines in phase diagrams

Note : these methods utilize the standard ANSI C convention for exchanging parameters, the distribution includes samples to show how to utilize these methods in different languages

Prode Properties includes methods for calculating different types of phase diagrams
vapor-liquid
vapor-liquid-liquid
vapor-liquid-solid (**)

(**) feature available in extended versions

typical application

- define the stream, set the required phase equilibria (vapor-liquid, vapor-liquid-liquid, vapor-liquid-solid)
- call PELnr() to calculate the phase diagram and obtain the number of lines available
- on each line call PELP(), PELT(), PELine() to obtain the data for the different lines
- if required call PFLine() to calculate a line with specified phase fraction ad state

integer lnr = PELnr(integer stream)
Given a stream calculates the phase diagram and returns the number of equilibrium lines available

integer lnr = PELT(integer stream, integer line)
Given a stream and the line, returns the line type (see below)

1 = bubble line
2 = dew line
3 = three phase line

integer lnr = PELP(integer stream, integer line)
Given a stream and the line, returns the line property (see below)

1 = vapor-liquid
2 = vapor-liquid-liquid
3 = vapor-solid
4 = liquid-solid

integer nrpt =PELine(integer stream, integer line, double *P, double *T, int maxpt)
Given a stream, the line and two arrays (0 -maxpt elements) the procedure returns nrpt < maxpt equilibrium points in specified line

integer nrpt =PVLine(integer stream, integer line, double *P, double *T, double *H, double *S,double *V,int maxpt)
Given a stream, the line and five arrays (0 -maxpt elements) the procedure returns nrpt < maxpt equilibrium points in specified line,
in additions to t,p values this method returns enthalpy, entropy and volume values calculated at equilibrium points

this method allows to calculate a line with specified phase fraction in specified state (gas,liquid,solid)

integer nrpt =PFLine(integer stream,int line, double pf, double *P, double *T, int maxpt)
Given a stream, the line, a specified phase fraction and two arrays (0-maxpt elements) the procedure returns nrpt < maxpt equilibrium points in specified phase fraction line

Methods for direct access to properties (F,H,S,V) and derivatives (T,P,W)

Prode Properties includes methods for fast access to the procedures for calculating thermodynamic properties, to access these methods create one or more processes with method DPinit() passing a integer as process code (permitted range of values 1..5) to identify each process and a stream (caution: must define the stream, i.e. the list of components and molar fractions, before to call DPinit), then call in sequence the methods for calculating thermodynamic properties passing as first parameter the process code, the methods will return the properties calculated for the stream associated with that specific process.
Note: Base version allows to define up to 5 iindependent processes

example of application
DPinit(1,stream);
StrHv(1,0,t ,p,X,&HL);
StrHv(1,1,t ,p,Y,&HV);

integer res = DPinit(integer process,integer stream)
Given a process (code 1-5) and a stream the method loads all data

integer res = StrFv(integer process,integer state,double t ,double p, double *w,double *fg)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the vector of fugacities (Pa)

integer res = StrFvd(integer process,integer state,double t ,double p, double *w,double *fg, double *dfgt, double *dfgp, double **dfgw)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the vector of fugacities (Pa) and related derivatives vs. temperature (K), pressure (Pa), composition (note : derivatives vs. composition as matrix [n][m])

integer res = StrFvdv(integer process,integer state,double t ,double p, double *w,double *fg, double *dfgt, double *dfgp, double *dfgw)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the vector of fugacities (Pa) and related derivatives vs. temperature (K), pressure (Pa), composition (note : derivatives vs. composition as vector [n*m])

integer res = StrHv(integer process, integer state,double t ,double p, double *w,double *H)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the molar enthalpy (Kj/Kmol)

integer res = StrHvd(integer process,integer state,double t ,double p, double *w,double *H, double *dHt, double *dHp, double *dHw)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the molar enthalpy (Kj/Kmol) and related derivatives vs. temperature, pressure, composition

integer res = StrSv(integer process,integer state,double t ,double p, double *w,double *S)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the molar entropy (Kj/Kmol-K)

integer res = StrSvd(integer process,integer state,double t ,double p, double *w,double *S, double *dSt, double *dSp, double *dSw)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the molar entropy (Kj/Kmol-K) and related derivatives vs. temperature, pressure, composition

integer res = StrVv(integer process,integer state,double t ,double p, double *w,double *V)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the molar volume (M3/Kmol)

integer res = StrVvd(integer process,integer state,double t ,double p, double *w,double *V, double *dVt, double *dVp, double *dVw)
Given a predefined stream the required state (vapor, liquid, solid)) and operating conditions returns the molar volume (M3/Kmol) and related derivatives vs. temperature, pressure, compositionMethods for stream' s data access

Extended methods for accessing stream's properties

These functions (which are otherways equivalent to standard methods) permit in addition to set the operating conditions at which the required property must be evaluated. This may result useful in many cases, for example when utilizing Prode Properties methods as macros from Excel cells. Caution : the isothermal flash will reset any previous settings and you should use with care these methods in sequential calculus.

double mw = EStrGMw(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the molecular weight for gas phase

double mw = EStrLMw(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the molecular weight for liquid phase

double lf = EStrLf(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns liquid fraction (molar basis) in stream

double pf = EStrPf(integer stream, integer state, double t, double p)
given a stream , state (gas, liquid, solid) pressure and temperature performs an isothermal flash and returns the phase fraction (molar basis) in specified state

double zv = EStrZv(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the relevant compressibility factor (gas phase)

double h = EStrH(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the enthalpy (gas + liquid phase)

double v = EStrV(integer stream, double t, double p)
given a stream, pressure and temperature performs an isothermal flash and returns the specific volume as sum of specific volumes of all phases

double cp = EStrGCp(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant pressure, gas phase)

double cv = EStrGCv(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant volume, gas phase)

double cp = EStrLCp(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant pressure, liquid phase)

double cv = EStrLCv(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the specific heat capacity (constant volume, liquid phase)

double c = EStrGIC(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the isothermal compressibility in gas phase

double c = EStrLIC(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the the isothermal compressibility in liquid phase

double ss = StrMSS(integer stream, double t, double p)
given the stream pressure and temperature performs an isothermal flash and returns returns the speed of sound (gas, liquid) as calculated with HEM model for mixed phases

double ss = EStrGSS(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the speed of sound in gas phase

double ss = EStrLSS(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the speed of sound in liquid phase

double jt = EStrGJT(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the Joule Thomson coefficient for gas phase

double jt = EStrLJT(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the Joule Thomson coefficient for liquid phase

double ic = EStrGIC(integer stream double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the isothermal compressibility coefficient - (1 / V) * dV / dP in gas phase

double ic = EStrLIC(integer stream double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the isothermal compressibility coefficient - (1 / V) * dV / dP in liquid phase

double v = EStrGVE(integer stream double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the volumetric expansivity coefficient - (1 / V) * dV / dT in gas phase

double v = EStrLVE(integer stream double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the volumetric expansivity coefficient - (1 / V) * dV / dT in liquid phase

double hc = EStrHC(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the net heat of combustion (gas phase).

double fl = EStrFML(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the flammability lean limit (gas phase).

double fl = EStrFMH(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the flammability rich limit (gas phase).

double s = EStrS(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the relative entropy (gas + liquid phase)

double d = EStrLD(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the calculated liquid density (at operating conditions).

double d = EStrGD(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the calculated gas density (at operating conditions).

double tc = EStrLC(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the calculated liquid thermal conductivity (at operating conditions).

double tc = EStrGC(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the calculated gas thermal conductivity (at operating conditions).

double v = EStrLV(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the r calculated liquid viscosity (at operating conditions).

double v = EStrGV(stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the calculated gas viscosity (at operating conditions).

double st = EStrST(integer stream, double t, double p)
given the stream, pressure and temperature performs an isothermal flash and returns the calculated surface tension (at operating conditions).

Methods for chemical's file access

Prode Properties includes a set of functions for chemical data file access. Components are referenced via a component code which is an integer with value in the range 1 to getFCNR()

Integer nr = getFCNr()
returns the number of components in Chemical's File

int str = MCompF(integer code, char *s, integer slm)
given the component code fills string s with the relevant component formula (eventually truncated to slm maximum lenght) , this is the Microsoft Excel specific method

char *str = CompF(integer code)
given the component code returns the relevant component formula (eventually truncated to string maximum length) , this is the ANSI C compatible method

int str = MCompN(integer code, char *s, integer slm)
given the component code fills string s with the relevant component name (eventually truncated to slm maximum length) , this is the Microsoft specific method

char *str = CompN(integer code)
given the component code returns the relevant component name (eventually truncated to string maximum length) , this is the ANSI C compatible method

int id = CompID(integer code)
given the component code returns component's ID (it's the CAS number)

int cc = CompCID(integer id)
given the component ID returns the component's code

double mw = CompMw(integer code)
given the component code returns the relevant molecular weight

double tc = CompTc(integer code)
given the component code returns the relevant critical temperature

double ac = CompAc(integer code)
given the component code returns the relevant acentric factor

double vc = CompVc(integer code)
given the component code returns the relevant critical volume

double pc = CompPc(integer code)
given the component code returns the relevant critical pressure

double dm = CompDm(integer code)
given the component code returns the dipole moment

double rg = CompRg(integer code)
given the component code returns the radius of gyration

double sol = CompSol(integer code)
given the component code returns the solubility parameter

double hf = CompHf(integer code)
given the component code returns the std. enthalpy of formation

double gf = CompGf(integer code)
given the component code returns the Gibbs energy of formation

double sf = CompSf(integer code)
given the component code returns the enthalpy of fusion

double nb = CompNb(integer code)
given the component code returns the normal boiling point

double mp = CompMp(integer code)
given the component code returns the melting point

double p = CompVP(integer code, double t)
given the component code and a temperature, returns the calculated saturation pressure (calculated via Chemical's file temperature dependent correlation)

double h = CompHG(integer code, double t0, double t1)
given the component code , initial and final temperatures for integration, returns the calculated ideal gas enthalpy (calculated via Chemical's file temperature dependent correlation)

double s = CompSG(integer code, double t0, double t1)
given the component code , initial and final temperatures for integration, returns the calculated ideal gas entropy (calculated via Chemical's file temperature dependent correlation)

double h = CompHL(integer code, double t0, double t1)
given the component code , initial and final temperatures for integration, returns the calculated ideal liquid enthalpy (calculated via Chemical's file temperature dependent correlation)

double s = CompSL(integer code, double t0, double t1)
given the component code , initial and final temperatures for integration, returns the calculated ideal liquid entropy (calculated via Chemical's file temperature dependent correlation)

double h = CompHS(integer code, double t0, double t1)
given the component code , initial and final temperatures for integration, returns the calculated ideal solid enthalpy (calculated via Chemical's file temperature dependent correlation)

double s = CompSS(integer code, double t0, double t1)
given the component code , initial and final temperatures for integration, returns the calculated ideal solid entropy (calculated via Chemical's file temperature dependent correlation)

double h = CompHV(integer code, double t)
given the component code and a temperature, returns the calculated latent heat (calculated via Chemical's file temperature dependent correlation)

double v = CompLV(integer code, double t)
given the component code and a temperature, returns the calculated liquid viscosity (calculated via Chemical's file temperature dependent correlation)

double v = CompGV(integer code, double t)
given the component code and a temperature, returns the calculated gas viscosity (calculated via Chemical's file temperature dependent correlation)

double d = CompLD(integer code, double t)
given the component code and a temperature, returns the calculated liquid density (calculated via Chemical's file temperature dependent correlation)

double tc = CompLC(integer code, double t)
given the component code and a temperature, returns the calculated liquid (thermal) conductivity (calculated via Chemical's file temperature dependent correlation)

double tc = CompGC(integer code, double t)
given the component code and a temperature, returns the calculated gas (thermal) conductivity (calculated via Chemical's file temperature dependent correlation)

double st = CompST (integer code, double t)
given the component code and a temperature, returns the calculated surface tension (calculated via Chemical's file temperature dependent correlation)

double d = CompSD(integer code, double t)
given the component code and a temperature, returns the calculated solid density (calculated via Chemical's file temperature dependent correlation)

double tc = CompSC(integer code, double t)
given the component code and a temperature, returns the calculated solid (thermal) conductivity (calculated via Chemical's file temperature dependent correlation)

Methods to set / access different options

To set / access the different options available in Prode Properties the library includes two methods, getOM() and setOM(), these methods accept / return a 32 bit integer, each bit in the integer represents a different option, see below a short list of the most important options available.

int om = getOM(integer stream)
given a stream returns a code (integer) which defines the settings

integer res = setOM (integer stream, integer option)
given a stream and model code sets the options.

Codes used in Prode library

Table of codes to specify the different options

reference : methods getOM(), setOM() ...

Caution! The codes may change in different versions.

Bit	Decimal value	Option
1	1	set multiphase vapor + liquid
2	2	set multiphase vapor + liquid + solid
3	4	set multiphase vapor + liquid + solid + hydrate
4	8	reduce the number of trial phases (in multiphase)
5	16	use iso compressibility coeff. to detect single phase state
6	32	evaluate stability of each phase in equilibrium
7	64	end specified phase fraction lines when crossing phase boundary lines
8	128	include all hydrate structures (also those not normally generated by formers)

to set one or more options call setOM() passing as value a integer with the sum (decimal values) of all required options.

Table of codes to specify the different states

reference : methods setMP() , PfTF() , PfTF() , StrFv(), StrFvd() ...

Code	State
0	Vapor phase
1	Liquid phase
2	Solid phase
3	Hydrate phase

Table of codes to specify the different models

reference : methods setMP(), getMP() ...

Some models may not be available and/or the numerical codes may change in different versions, contact Prode for details

Code	Description	Model
1	Regular	Regular
10	Wilson	Wilson
11	NRTL	NRTL
12	UNIQUAC	UNIQUAC
30	Soave-Redlich_Kwong Std. (VDW)	SRK(VDW)
31	Soave-Redlich_Kwong Ext. (VDW)	SRKX(VDW)
40	Soave-Redlich_Kwong Ext. + NRTL (Modified Huron Vidal)	SRKX-NRTL(P-HV)
41	Soave-Redlich_Kwong Ext. + NRTL (Modified LCVM)	SRKX-NRTL(P-LCVM)
50	Peng Robinson Std. (VDW)	PR(VDW)
51	Peng Robinson Ext. (VDW)	PRX(VDW)
55	Peng Robinson Ext. + Wilson (Wong Sandler)	PRX-Wilson(WS)
56	Peng Robinson Ext. + UNIQUAC (Wong Sandler)	PRX-UNIQUAC(WS)
57	Peng Robinson Ext. + NRTL (Wong Sandler)	PRX-NRTL(WS)
60	Peng Robinson Ext. + NRTL (Modified Huron Vidal)	PRX-NRTL(P-HV)
61	Peng Robinson Ext. + Wilson (Modified Huron Vidal)	PRX-Wilson(P-HV)
62	Peng Robinson Ext. + UNIQUAC (Modified Huron Vidal)	PRX-UNIQUAC(P-HV)
65	Peng Robinson Ext. + Wilson (MHV2)	PRX-Wilson(MHV2)
66	Peng Robinson Ext. + UNIQUAC (MHV2)	PRX-UNIQUAC(MHV2)
67	Peng Robinson Ext. + NRTL (MHV2)	PRX-NRTL(MHV2)
70	Peng Robinson Ext. + NRTL (Modified LCVM)	PRX-NRTL(P-LCVM)
71	Peng Robinson Ext. + Wilson (Modified LCVM)	PRX-Wilson(P-LCVM)
72	Peng Robinson Ext. + UNIQUAC (Modified LCVM)	PRX-UNIQUAC(P-LCVM)
73	Peng Robinson Ext. + UNIFAC (Modified LCVM)	PRX-UNIFAC(WS)
80	Benedict-Webb-Rubin (modified)	BWR
81	Benedict-Webb-Rubin-Starling	BWRS
90	Lee Kesler	LK
91	Lee Kesler Ploecker	LKP
100	P-SAFT	PSAFT
110	Soave-Redlich_Kwong Ext. - CPA	SRKX-CPA(VDW)
111	Peng Robinson Ext. - CPA	PRX-CPA(VDW)
115	Peng Robinson Ext. - CPA + NRTL (Modified Huron Vidal)	PRXCPA-NRTL(P-HV)
116	Peng Robinson Ext. - CPA + NRTL (Modified LCVM)	PRXCPA-NRTL(P-LCVM)
117	Peng Robinson Ext. - CPA + NRTL (Modified MHV2)	PRXCPA-NRTL(P-MHV2)
118	Peng Robinson Ext. - CPA + NRTL (Modified Wong Sandler)	PRXCPA-NRTL(P-WS)
120	Soave-Redlich_Kwong Ext. - CPA + NRTL (Modified Huron Vidal)	SRKX-CPA-NRTL(P-HV)
130	UNIFAC	UNIFAC
150	Solid Pure (derived from) PRX-NRTL(P-HV)	SPRX-NRTL(P-HV)
151	Solid Pure (derived from) PRXCPA-NRTL(P-HV)	SPRXCPA-NRTL(P-HV)
153	Solid Solution (derived from) PRX-NRTL(P-HV)	SSPRX-NRTL(P-HV)
170	Hydrate (derived from) PRXCPA-NRTL(P-HV)	HPRXCPA-NRTL(P-HV)
171	Hydrate (derived from) PRX-NRTL(P-HV)	HPRX-NRTL(P-HV)
180	Wax	Wax
185	Asphaltene	Asphaltene
200	Pitzer (Electrolyte)	PITZER
205	Peng Robinson Ext. -CPA-(MSA) + NRTL (Modified Huron Vidal) Electr.	PRXCPA-E-NRTL(P-HV)
210	P-SAFT-(MSA) Electrolyte	PSAFT-E
300	Steam tables based on IAPWS 1995 formulation	IAPWS 95
311	GERG 2008	GERG 2008
312	ISO 18453 (GERG)	ISO 18453
315	ISO 20765 (AGA 8)	ISO 20765

Methods to define thermodynamic models

To define or retrieve the thermodynamic models associated with a stream the library includes several methods setKM() works with predefined packages while setMP(), getMP() allow to define specific models on each property (Fg, H, S,V..).

integer res = setKM (integer stream, integer Kcode)
given a stream and the code for the predefined package (contact Prode for the list of predefined packages available in different versions) sets the package.

integer res = setMP(integer stream, integer mp, integer model, integer state)
given a stream, property (Fg,H,S..) model and state (Vapor,Liquid,Solid,Hydrate) this method sets the specified model for that property and returns TRUE in case of success, otherwise returns FALSE

integer m = getMP(integer stream, integer mp, integer state)
given a stream, related property (Fg,H,S..) and state (Vapor,Liquid,Solid,Hydrate) this method returns the specified model for that property and state

Table of codes to specify the different properties in setMP() and getMP()

reference : methods setMP(), getMP() ...

Code	Property
1	Fugacity
2	Enthalpy
3	Entropy
4	Volume
5	Viscosity
...	(additional properties available in extended versions)

Methods to define base values for Enthalpy and Entropy

The library allows to define the base values (the temperature and initial value from which to start integration) for entropy and enthalpy from Properties Editor, in setting's page, these values are stored in archive and restored when program starts.
In addition it is possible to modify these value by code with the following methods,

integer res = setHB(integer mod, double t, double val)
given a code to identify the procedure (see the table with codes), the temperature and initial value sets base value for enthalpy .

integer res = setSB(integer mod, double t, double val)
given a code to identify the procedure (see the table with codes), the temperature and initial value sets base value for entropy .

Table of codes to specify the different base values in setHB() and setSB()

reference : methods setHB(), setSB() ...

Code	Procedure
1	initial values specified by user (values of t and val)
2	initial values are enthalpy of formation (or entropy of formation) and temperature 25 C

Methods to set / access stream's names

In Prode Properties streams have several properties including a label (name) which could match (for example) the name of a line in your project, you can easily set / access these labels through a series of methods.

integer str = MStrN(integer stream, char *s, integer slm)
given a integer (that identifies a stream) method fills string s with the name of stream (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = StrN(integer stream)
given a integer (that identifies a stream) method returns as ANSI C type the string identifying that stream.

integer res = putN(integer stream, char *str)
given a integer (that identifies a stream) and a ANSI C string identifying that stream this method sets the label.

Methods to access Model's data

Prode Properties includes models for calculating properties as fugacities, enthalpies, entropies, volumes, viscosities etc. these methods allow to access the models available

integer nr = getMDnr()
returns the number of models available in this version

char *str = getMDN(int model)
given the model position (in the range 1-number of models available) method returns as ANSI C type the string identifying that model.

integer res = getMDP(int model, int prop, int state)
given the model position (in the range 1-number of models available) the required property and state returns TRUE if model can calculate the specified property, otherwise returns FALSE

integer code = getMDC(int model)
given the model position (in the range 1-number of models available) returns the code of the model

Methods to control error's messages

PROPERTIES includes a set of functions to control the error messages. By default PROPERTIES produces an error message via a Microsoft Windows Dialog Box every time an error is discovered. This approach can slow down the process when a long sequence of errors occurs in an extended calculation sequence, such as an iterative convergence calculation. A better solution in that situation is to provide a status flag that can be interrogated and used by the users at convenient points in the sequence.

setErrFlag (integer state)
given a Boolean (state) sets the error flag to TRUE or FALSE. The flag should be cleared (state = FALSE) before each sequence of calculations and tested (method getErrFlag()) after the calcs. If this is done, then a flag state of TRUE indicates that an error has occurred somewhere in the calculation sequence).

integer res = getErrFlag ()
a value of TRUE means that an error has been found, please note that PROPERTIES doesn't clear the error flag state, You should clear the error flag (via setErrFlag()) before each sequence of calc's.

defErrMsg (integer state)
a value TRUE for variable state sets on the Microsoft Windows Dialog Box and a message will appear every time an error is discovered. A value FALSE sets off the dialog box (no messages of error).

integer str = MErrMsg(char *s, integer slm)
fills string s with the last error message generated (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = ErrMsg()
Returns the last error message generated, this is the ANSI C compatible method

Methods for accessing data-editing windows

Prode Properties includes two predefined methods for activating Properties editor

integer res = edS(nteger stream)
given a integer (that identifies a stream) method activates the Properties Editor on the specified stream

integer res = edSS()
this method activates the Properties Editor on first stream

Methods to load / save archives

Archives are files which contain a copy of the data used by Prode Properties to manage stream's and units of measurement, when you open an archive the stream's data and units are loaded, when you choose to save an archive these data are stored in a file. Archives are useful to create copies of your work otherways all data will be lost when leaving the application, Prode Properties includes methods for operations on archives.

integer res = AOpen()
open a file as archive (browse for file)

integer res = AFOpen(char *path)
open the file specified in *path as archive

integer res = ASave()
save a file as archive (browse for file)

integer res = AFSave(char *path)
save the file specified in *path as archive

Methods for accessing / defining the units of measurement

Prode Properties includes methods for accessing and defining the units of measurement, these methods utilize a numeric code for identifying the correspondent quantities, refer to the paragraph “Access via software to the units of measurement” for a list of these codes.

integer res = getUMC(integer UM)
given a integer (that identifies a quantity) method returns the selected UM for that quantity.

integer res = setUMC(integer UM, integer sel)
given two integers (the first identifies a quantity and the second the selection) method selects a UM for that quantity.

integer res = getUMN(integer UM)
given a integer (that identifies a quantity) method returns the number of different units of measurement available for that quantity.

integer str = MgetUMS(integer UM, integer sel, char *s, integer slm)
given two integers (the first identifies a quantity and the second the selection) fills string s with selected UM (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = getUMS(integer UM, integer sel)
given two integers (the first identifies a quantity and the second the selection) method returns as ANSI C type the string identifying the selected UM.

integer str = MgetSUMS(integer UM, char *s, integer slm)
given a integer UM for quantity fills string s with selected UM (eventually truncated to slm maximum lenght), this is the Microsoft specific method

char *str = getSUMS(integer UM)
given a integer UM for quantity this method returns as ANSI C type the string identifying the selected UM.

double res = UMCR(double value, integer UM, integer SEL)
given a value, the code for quantity and selection converts to reference and returns the result

double res = UMCS(double value, integer UM, integer SEL)
given a value, the code for quantity and selection converts from reference and returns the result

integer res = UMAU(double a, double b, char *name, integer UM)
given the code for a quantity, the parameters a, b required for conversion and the name adds a new (user defined, temporary) unit.

integer res = UMRAU(integer UM)
given the code for a quantity removes all additional (temporary) units

Additional methods

double p = getPatm()
returns the internal reference (user defined) for atmosferic pressure quantity.

Application examples

We present here some notes about Prode Properties applications in form of FAQ that should assist users to easily extend features or add interfaces.

Tips on creation of Prode Properties applications

- Include a command (menu', button etc.) for accessing the Properties Editor (method edS(stream))
- Ensure that units of measurement are those defined in Prode Properties or include methods to set the units.
- Use isSDef() method to test a streams validity before accessing the stream. Accessing an undefined stream generates a large numbers of errors.
- Include functions for controlling error messages if you have extended calculation sequences. When managing error messages ensure that you test at the end of the calculation sequence to capture any problems that may have occurred.
- When debugging always attempt to limit the complexity of problems and expand progressively to the full application, retesting at intervals as you expand the scope of your problem.

User defined ID for accessing the components in chemical's file

In Prode Properties each component in chemical's file includes a ID which permits to access that component, this feature permits to maintain a unique identification number also when the chemical file changes. The ID must be a integer type, as default the CAS number has been adopted but the user may edit and change this value from theProperties Editor. The functions for accessing components in Prode Properties require the component code, this code may vary in different versions of chemical file, to convert the ID to the component code utilize the method CompCID() and CompID() to convert from code to ID

```
' this example shows how to access data
' with  ID code
Dim code As Long , ID As Long, Pc As Double
ID      = 74840             ' CAS code for Ethane (but the user may define his own list of values)
code   = CompCID(ID)     ' get the code
Pc = CompPc(code)        ' and the critical pressure
```

How to define directly a stream (without accessing the Properties Editor)

Prode Properties includes methods to access (read and write) each different value in a stream, making it possible for the user to create procedure to define / edit / update directly each value without going through the Properties editor
Following list presents the methods for accessing all items

	read the value	**set the value**
• operating pressure,	getP()	setOp()
• operating temperature,	getT()	setOp()
• flow,	getW()	setW()
• vector [getMCNr() elements] with components codes	getCC()	putCC()
• vector [getMCNr() elements] Z vector, mole basis	getZ()	putZ()
• thermodynamic model (and related options)	getMP()	setMP()
• vector[getMBPNr() elements] of Ci	getCi()	putCi()
• vector[getMBPNr() elements] of Cj	getCj()	putCj()
• vector[getMBPNr() elements] of BIP matrix	getBIP()	putBIP()

When defining a stream one must follow these steps.

- call initS() method to clear all stream's data
- definedata
- call setS() method to validate the data

The following example shows how to define a 2 components stream

```
Call initS(Stream)
Call setMP(Stream, Fg, SRK, 0)   ' see the paragraph "Codes used in Prode library"
Call setMP(Stream, Fg, SRK, 1)
.........................        ' define the models for all required properties and states
Call putZ(Stream, 1, z1)
Call putCC(Stream, 1, cc1)
Call putZ(Stream, 2, z2)
Call putCC(Stream, 2, cc2)
Call setS(Stream)                ' validates stream

Call setW(Stream, W)             ' flow

Call loadSB(Stream,0)                    ' load VLE BIPS from database
```

How to save and restore streams to / from a file

Archives are files which contain a copy of all compositions, operating conditions, units of measurement, settings etc. , archives are useful for creating copies of your work otherways all data will be lost when leaving the application.
The library includes methods to load and save archives , see the paragraph “Methods to load / save archives” for the list.

```
Call AOpen()                    ' open a file as archive (browse for file)

Call AFOpen("e:/def.ppp")       ' open the specified file as archive

Call ASave()                    ' save a file as archive (browse for file)

Call AFSave("e:/def.ppp")       ' save the specified file as archive
```

Error messages

PROPERTIES may generate the following error messages. For some of these, an action is suggested

Memory allocation error
A limit in resources allocation (close applications, release memory and restart)

Corrupted file, error reading data file
PROPERTIES cannot access a file, this may depend from the file not being in the proper directory or being corrupted, it is suggested that You reinstall PROPERTIES.

Internal error
This error may depend from several different conditions, the most common is a wrong parameter in a function (i.e. an attempt to pass a value out of permitted range). Check Your code.

too many local variables
too many variables
a limit in resources allocation (see above)

calc. on undefined stream data
an undefined stream found while executing calc's (edit and define the stream)

undefined stream' s operating conditions
pressure, temperature or flow are undefined (edit and define the stream)

error calling thermo calc. procedure
wrong input value (calcs cannot converge) or calcs outside temperature range (check chemical's file for limits in temperature correlation's).

cannot converge calc' s loop
A wrong convergence condition has been specified (i.e in an adiabatic flash calcs a thermal condition that cannot be reached by varying temperature, pressure or liquid fraction ; a parameter is outside range limits etc.)

T, P values outside H, S range calcs
A wrong condition has been specified and a parameter in enthalpy /entropy calcs is outside range limits

too many comp' s in a stream
when two or more streams are mixed the total nr. of components may exceed the maximum
some inconsistences in stream's data

error accessing component' s data archive
unavailable data (a unspecified component) or calc's outside temperature range.

Stack error (no memory), reload procedure
a limit in resources allocation (see above)

Method not available in this version
Attempt to define a method not available in that version, edit the stream and define a new method

A stream with Steam Tables model must have only 1 component
You should specify a stream with one component only in order to apply ASME Steam Tables model

Calculation basis

The user can specify which method to use selecting the models.
Please refer to the paragraph “reference literature” and “Models” for additional information about the methods.

Fugacity	calculated according selected model
Enthalpy	calculated according selected model
Entropy	calculated according selected model
Volume	calculated according selected model

Viscosity

gas
low pressure mixing rule according Wilke (1950) , operating conditions correction according Stiel and Thodos (1964).
liquid
logarithmic average mixing rule, pressure correction according Lucas (1981)

Thermal conductivity

gas
low pressure mixing rule according Mason and Saxena (1958), operating conditions correction according Stiel and Thodos (1964)
liquid
mixing rule according Li (1976)

Surface tension

mixing rule according MacLeod-Sugden

Heat of combustion

weight average mixing rule according ISO std. (database contains values in Kj/Kg)

Flammability limits

mixing rule according Le Chatelier as discussed by Coward & Jones (1952)

Limits in thermodynamic calc’s

Enthalpy, Entropy calc’s

In Prode Properties the user can specify different initial conditions for enthalpy and entropy, see the paragraph “Config settings” for additional details.

Temperature, pressure ranges

Temperature range	1 K - 5000 K
Pressure range	1 Pa – 1000 Bar

Chemical's File format

Note : all data dependent correlation's in chemicals file have a range of temperature for application, outside this range they may provide inconsistent results. Prode Properties checks for this range (as defined by high and low limits in chemicals file) and attempts to extend data when required (when operating conditions are outside the range of application of correlations), this may produce in some cases inconsistent results with simple models and properties which require differentiation, for example specific heat capacity.

Flexible data format

Prode Properties utilizes proprietary code which allows up to 30 correlations and custom units to define each temperature dependent property, all major standards including DIPPR and others are supported.

Chemical' s data file

Prode Properties base version adopts the following format

```
Formula string 12 chars max
Name (1) (main list) string 40 chars max
Name (2) (user defined list) string 40 chars max
Name (3) (user defined list) string 40 chars max
Identification number (CAS as default)
Molecular weight
Critical temperature
Critical pressure
Critical volume
Acentric factor
Dipole Moment
Radius of Gyration
Solubility parameter
Standard enthalpy of formation (298 K)
Gibbs free energy of formation (298 K, 1 atm)
Enthalpy of fusion
Normal boiling point
Melting point
Flammability lean limit % (range 0-100)
Flammability rich limit % (range 0-100)
Autoignition temperature
Net heat of combustion
Flash Point

Gas heat capacity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Vapor viscosity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Vapor thermal conductivity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)
```

Heat of vaporization correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Liquid vapor pressure correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Surface tension
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Liquid density correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Liquid viscosity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Liquid thermal conductivity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Liquid heat capacity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Solid vapor pressure correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Solid density correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Solid thermal conductivity correlation
type of equation
unit for property
unit for temperature
low temperature limit
high temperature limit
A-E (5 parameters)

Solid heat capacity correlation
type of equation
unit for property
unit for temperature
low temperature limit
igh temperature limit
A-E (5 parameters)

Sources of data

Data in chemical data file come from several sources including :

- “Dechema Chemistry Data ser.” text books
- “DIPPR data collection” text books
- “Technical Data Book, Petroleum Refining”

Due to the large differences in critical and transport properties found in different sources, DIPPR (AICHE Design Institute for Physical PRoperty Data) reference has been selected as a default.

Component’s identification

Components are identified by name (from DIPPR list) , chemical formula and Identification number.

Regression procedures and results

Coefficients in correlations have been calculated with a custom program that uses a modified version of Levenberg-Marquardt algorithm , reported errors (at each fitting point) are usually lower than 1 % of input values for the most complex correlations (i.e. vapor pressure),), however in some cases they may be higher.

Consistency tests

When relations exist between thermodynamic properties (i.e. acentric factor and critical pressure and temperature, vapor pressure and heat of vaporization etc.) a consistency test has been performed.

Comparing Prode Properties results against those of different process simulators

When comparing data from different tools one must verify that

- the different tools do use the same thermodynamic models
- properties in databanks have siimilar values
- lists and values of BIPs and other parametres which can influence results have similar values

Models

Prode Properties includes a complete set of thermodynamic models, see also the table “Features available vs. Versions”

Liquid activities
Wilson
NTRL
UNIQUAC

Predictive
UNIFAC

Electrolytes
Pitzer
CPA-electrolyte
SAFT-electrolyte

Cubic EOS
Soave-Redlich-Kwong, Peng-Robinson with std. alpha function and VdW mixing rules,
Extended versions of SRK and PR including parameters calculated to fit experimental data (saturation pressures, densities, heat capacities etc.) and different mixing rules to combine equations of state with activity models
Std. and Modified versions of Huron Vidal (HV) rule
Std. and Modified versions of Linear Combination of Vidal and Michelsen (LCVM) rule
Std. and Modified versions of Michelsen-Huron-Vidal (MHV2) rule
Std. and Modified versions of Wong Sandler (WS) rule
etc...

Advantages of modified mixing rules (EOS+Activity Models)
These rules allow to reproduce the classical mixing rule (VdW) including the binary interaction coefficients for the terms where the parameters for activity model are not available, for example with mixtures containing a large number of hydrocarbons plus water, methanol or similar chemicals, the existing binary interaction coefficients for hydrocarbons can be preserved adding the flexibility of NRTL, Wilson or UNIQUAC expressions to model the non-ideal pairs.

Other models
Modified Benedict-Webb-Rubin
Benedict-Webb-Rubin-Starling
Lee-Kesler
Lee-Kesler-Plocker

Models based on associating fluid theory
Different versions of CPA Cubic Plus Association based on Soave Redlich Kwong and Peng Robinson models with VdW mixing rules and several others to combine equations of state with activity models
Std. and Modified versions of Huron Vidal (HV) rule
Std. and Modified versions of Linear Combination of Vidal and Michelsen (LCVM) rule
etc...

Different versions of SAFT (Perrturbed Chain Statistical Associating Fluid Theory)

SOLIDS
SPM (Solid Pure Model) solid phase treated as single component
SSM (Solid Solution Model) solid phase treated as homogeneous solution
WAX solid phase treated as homogeneous solution (with specific parameters)
Asphaltene
Hydrates (based on Van der Waals and Plateeuw theory with a std. model and a complex model)

STANDARDS
GERG 2008 (ISO 20765)
AGA 2017 (2017 version with GERG 2008 formulations)
Steam tables (IAPWS 1995) Water / steam properties calculated according IAPWS 1995 formulation

The models export derivatives of Fg, H, S, V vs. W, P, T

UNIFAC functional groups

The underlying idea in UNIFAC method is that a molecule can be considered as a collection of functional groups. The main advantage of this approach is that from a relatively small number of functional groups the properties of many different molecules can be predicted. The UNIFAC model is useful for estimating solution behaviour in the absence of experimental data. Prode Properties incorporates the UNIFAC Group Contribution revision 5 (January 1992, J.P.Baker).
Following the main groups and subgroups table :

Code	Main	Subgroup	Example
1	CH2	CH3	Hexane
2		CH2	n-Hexane
3		CH	2-Methylpropane
4		C	Neopentane
5	C=C	CH2=CH	1-Hexene
6		CH=CH	2-Hexene
7		CH2=C	2-Methyl-1-butene
8		CH=C	2-Methyl-2-butene
70		C=C	2,3-Dimethylbutene
9	ACH	ACH	Naphthaline
10		AC	Styrene
11	ACCH2	ACCH3	Toluene
12		ACCH2	EthylBenzene
13		ACCH	Cumene
14	OH	OH	n-Propanol
15	CH3OH	CH3OH	Methanol
16	H2O	H2O	Water
17	ACOH	ACOH	Phenol
18	CH2CO	CH3CO	Butanone
19		CH2CO	Pentanone-3
20	CHO	CHO	Propionic aldehyde
21	CCOO	CH3COO	Butyl acetate
22		CH2COO	Methyl propionate
23	HCOO	HCOO	Ethyl formate
24	CH2O	CH3O	Dimethyl ether
25		CH2O	Diethyl ether
26		CHO	Diisopropyl ether
27		THF	Tetrahydrofuran
28	CNH2	CH3NH2	Methylamine
29		CH2NH2	Ethyl amine
30		CHNH2	Isopropylamine
31	CNH	CH3NH	Dimethylamine
32		CH2NH	Diethyl amine
33		CHNH	Diisopropylamine
34	(C)3N	CH3N	Trimethylamine
35		CH2N	Triethylamine
36	ACNH2	ACNH2	Aniline
37	Pyridine	C5H5N	Pyridine
38		C5H4N	2-Methyl pyridine
39		C5H3N	2,3-Dimethylpyridine
40	CCN	CH3CN	Acetonitrile
41		CH2CN	Propionitrile
42	COOH	COOH	Acetic acid
43		HCOOH	Formic acid
44	CCl	CH2Cl	Butane-1-chloro
45		CHCl	Propane-2-chloro
46		CCl	2-Methylpropane-2-chloro
47	CCl2	CH2Cl2	Methane-dichloro
48		CHCl2	Ethane-1,1-dichloro
49		CCl2	Propane-2,2-dichloro
50	CCl3	CHCl3	Chloroform
51		CCl3	Ethane-1,1,1-trichloro
52	CCl4	CCl4	Methane-tetrachloro
53	ACCl	ACCl	Benzene-chloro
54	CNO2	CH3NO2	NitroMethane
55		CH2NO2	Propane-1-nitro
56		CHNO2	Propane-2-nitro

Code	Main	Subgroup	Example
57	ACNO2	ACNO2	Benzene-nitro
58	CS2	CS2	Carbon Disulfide
59	CH3SH	CH3SH	Methanethiol
60		CH2SH	Ethanethiol
61	Furfural	Furfural	Furfural
62	DOH	DOH	1,2-Ethanediol
63	I	I	Iodoethane
64	Br	Br	Bromoethane
65	C-C	CH-C	Hexyne-1
66		C-C	Hexyne-2
67	DMSO	DMSO	Dimethylsulfoxide
68	ACRY	Acrylnitril	Acrylnitrile
69	ClCC	Cl-(C=C)	Ethene-trichloro
71	ACF	ACF	Hexafluorobenzene
72	DMF	DMF-1	N,N-Dimethylformamide
73		DMF-2	N,N-Diethylformamide
74	CF2	CF3	Perfluorohexane
75		CF2	
76		CF	Perfluoromethylcyclohexane
77	COO	COO	Methyl acrylate
78	SiH2	SiH3	Methylsilane
79		SiH2	Diethylsilane
80		SiH	Heptamethyltrisiloxane
81		Si	Heptamethyldisiloxane
82	SiO	SiH2O	1,3-Dimethyldisiloxane
83		SiHO	1,1,3,3-Tetramethyldisiloxane
84		SiO	Octamethylcyclotetrasiloxane
85	NMP	NMP	N-methylpyrrolidone
86	CClF	CCl3F	Trichlorofluoromethane
87		CCl2F	Tetrachloro-1,2-difluoroethane
88		HCCl2F	Dichlorofluoromethane
89		HCClF	1-Chloro-1,2,2,2,-tetrafluoroethane
90		CClF2	1,2-Dichlorotetrafluoroethane
91		HCClF2	Chlorodifluoromethane
92		CClF3	Chlorotrifluoromethane
93		CCl2F2	Dichlorodifluoromethane
94	CON	CONH2	Acetamid
95		CONHCH3	N-Methylacetamid
96		CONHCH2	N-Ethylacetamid
97		CON(CH3)2	N,N-Dimethylacetamid
98		CONCH3CH2	N,N-methylethylacetamid
99		CON(CH2)2	N,N-Diethylacetamid
100	OCCOH	C2H5O2	2-Ethoxyethanol
101		C2H402	2-Ethoxy-1-propanol
102	CH2S	CH3S	Dimethylsulfide
103		CH2S	Diethylsulfide
104		CHS	Diisopropylsulfide
105	Morpholine	MORPH	Morpholine
106	Thiophene	C4H4S	Thiophene
107		C4H3S	2-Methylthiophene
108		C4H2S	2,3-Dimethylthiophene