# **Prode Properties**

Properties of pure fluids and mixtures

User's Manual rel. 1.2 Microsoft Windows version

PRODE www.prode.com

# Contents

License agreement	3
Customer support	
Introduction	4
What's new	
Installing the program	7
Prode Properties Quick Start	8
Data files folder	
Make sure all users can access data files folder	8
Avoid errors in read / write operations	
Getting Started from Microsoft Excel, part 1	
Prode Properties Add-In menu	10
Working with archives	
Getting Started from Microsoft Excel, part 2 operations with editor	
Getting Started from Microsoft Excel, part 3 working with predefined pages	
Getting started from LibreOffice	
Getting started from Python	
Getting started with Microsoft NET (VB , C) applications	
Solving problems (introduction)	
Working with archives, save and load data, default settings	
Properties editor	
Stream operating	44
Predefined Operations	
Stream Composition	46
Stream Models	
BIPs	
Config Units	
Config Settings	
Chemicals data	
Chemicals Settings	
Regress raw data	
Binary Interaction Parameters (BIP)	
Regress VLE-LLE-SLE data	
Parameters of models	
Accessing Prode Properties library	
Translate resources to different languages	
Microsoft Applications and Strings	64
Define models, compatibility with old verions	64
Units of measurement	
Introducing Prode Properties library methods	
Methods for thermodynamic calc's	
Methods for stream's data access	
Methods for stream's definition	
Methods to define stream's operating conditions	
Copy of streams	12
Methods for solving staged columns	
Methods for Reactors	
Methods for fluid flow problems	
Methods for solving a Polytropic operation	
Methods for orifices, relief valves design / rating Methods for calculating equilibrium lines in phase diagrams	
Methods for direct access to properties (F,H,S,V) and derivatives (T,P,W)	
Extended methods for accessing stream's properties	
Methods for chemical's file access	
Methods to set / access different options	
Codes used in Prode library	
Methods to define thermodynamic models	
Methods to define base values for Enthalpy and Entropy	
Methods to set / access stream's names	
Methods to access Model's data	
Methods to control error's messages	
Methods for accessing data-editing windows	
Methods to load / save archives	
Methods for accessing / defining the units of measurement	
Additional methods	
Application examples	
How to define directly a stream (without accessing the Properties Editor)	
How to save and restore streams to / from a file	
Error messages	
Calculation basis	
Limits in thermodynamic calc's Chemical's File format	
Chemical s File format	
Comparing Prode Properties results against those of different process simulators	
Models	
UNIFAC functional groups	

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**

Prode Properties library includes C++ ANSI ISO standard code with no limits on number of concurrent threads, the version for Windows (the same code is available for Android, Linux etc.) is released in form of compact Dynamic Libraries 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...
- <sup>4</sup> 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 (for Windows) distribution includes two libraries, ppp.dll (core methods) and pppx.dll (graphic interface).

#### **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 [2013-2019] 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		х	x
Database with more than 30000 BIPs		x	x
Complete set of thermodynamic models		х	x
SAFT models			EX
Asphaltene models			EX
Wax models			EX
GERG (2008)			EX
Electrolytes			EX
Hydrates Std. model	х	х	x
Hydrates Complex model			EX
Derivatives vs. P,T,W of Fg, H, S, V	х	х	х
Properties of fluids and mixtures	х	х	х
Multiphase flash with specified T, P, H, S, V	х	х	х
Additional flash operations			EX
Vapor-Liquid-liquid phase diagrams	х	х	x
Vapor-Liquid-Solid phase diagram			EX
VLE-LLE-SLE data regression	х	х	x
Raw data regression utility	х	х	x
Characterization of petroleum fractions			EX
Multiphase (gas,liquid) pipeline with heat transfer			EX
Isentropic nozzle HEM . HNE	х	х	x
Isentropic nozzle HNE-DS , NHNE			EX
Polytropic stage, vapor-liquid (gas+liquid)		х	x
Distillation (vapor-liquid)	х	х	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

#### Installation procedure

1) if previous versions are installed, login as admin and uninstall Prode Properties

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 a commercial license**

Your personal copy of Prode Properties doesn't include all the features available with a full license, you can purchase a Software License (single installation) or a Network Based License

#### Order a software license

this license is based on the Installation Code which appears in Prode Properties Editor

- Install Prode Properties on your computer (see the description above)
- install the add-in for Microsoft Excel following the instructions in paragraphs "Getting Started from Microsoft Excel"
- Run Prode Properties Editor from Excel, select the License::Data page , in this example the ID is ADAC6J7D27779A3M
- order a software license and include your ID (to avoid errors you can copy and paste the text)

Chemicals::Data Chemicals::Settings Chemicals::Regress BIPs::Data BIPs::Regress Models::Data Lic	ettings se::Data
1	se::Data
ADAC6J7D27779A3M	
ADAC6J7D27779A3M	
A	
	-

#### Activate a license

After the order, you will receive from Prode a text file containing a license key

• Run Prode Properties Editor from Excel, select the License::Data page and enter (to avoid errors you can copy and paste the text) your license key (see below), then click on button Store License

Stream::Operatin	g Stream::Con	nponents	Stream::Moo	dels Stre	am::BIPs	Co	nfig::Units	Config::Sett
Chemicals::Data	Chemicals::Setting	gs Chem	icals::Regress	BIPs::Data	BIPs::Reg	gress	Models::Da	ata License:
ionnealsb'ata	onennediboetang	jo onem	iculo togrooo	Dir G.D'did	Dir G.irtoş	9.000	modelabr	
ADAC6J7D2777	20.4-04.4							
ADAG6J7D2777	9A3WI							
71 128222223324	U2F5U7K3H28A29J	649B35624	1537237453725	57749F4B795		6461 131	/205000256	
Contraction of the second second					75451 014020	040031	121 3446200	NONIJJALJTA
8QA38J2M434P	4D9S2B6M2M67AF7	/03440904	P4V9A092E0930	TOUZPON				

the software will report "License Key Stored", at that point close Prode Properties Editor and stop Excel, the license will be activated the next time you'll run Prode Properties

## 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

ccel Options				?
General	View and manage Microsoft Office Add	-ins.		
Formulas				
Proofing	Add-ins			
Save	Name 🔺	Location	Туре	
anguage	Hidden Worksheets	C:\Program Files\Microsoft Office\Office14\OFFRHD.DLL C:\Program Files\Microsoft Office\Office14\OFFRHD.DLL	Document Inspector	
Advanced	Invisible Content Microsoft Actions Pane 3	C:\Program Files\Microsoft Office\Office14\OFFRHD.DLL	Document Inspector XML Expansion Pack	
Customize Ribbon Quick Access Toolbar		C:\Program Files\Prode\Excel\properties.xla C:\ft Office\Office14\Library\SOLVER\SOLVER.XLAM	Excel Add-in Excel Add-in	•
Add-Ins	Add-in: Properties Publisher:			
Trust Center	Compatibility: No compatibility information available Location: C:\Program Files\Prode\Excel\properties Description:	s.xla		
	Manage: Excel Add-ins			
			OK Can	cel

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.

Add-Ins available:     Analysis ToolPak     OK     Cancel     Browse     Autornation     Autornation     Properties	Add-Ins	🔣 Browse	×
Analysis ToolPak   Analysis ToolPak   Corganize ▼ New folder     Properties     Browse   Automation     Properties     Properties <td>Add-Ins available:</td> <td>G → Local Disk (C.) ▼ Program Files ▼ Prode ▼ Excel ▼ 🕼 Search Excel</td> <td></td>	Add-Ins available:	G → Local Disk (C.) ▼ Program Files ▼ Prode ▼ Excel ▼ 🕼 Search Excel	
14 15 File name: properties.xla Add-Ins (*.xlam;*.xla;*.xll)	Analysis ToolPak - VBA Euro Currency Tools Properties Solver Add-in Automation	Organize V New folder	ate modified  1/23/2016 4:38
	14		· ·
16	15	File name: properties.xla 💽 Add-Ins (*.xlam;*.xla	a;*.xII) ▼
Took V OK V Cancel	16	Tools - OK -	Cancel
	17		

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

X 🖬 🖻	) - (% -	Ŧ						Book2 - Micro	osoft Excel	
File	Home	Insert	Page Layout	Formulas	Data	Review	View	Developer	Add-Ins	
Proper	ties *									
Edit	Properties									
Ope	n Archive									
Save	a Archive						_			
3										

then select the folder and the name of file

ganize 🔻 New folder			800 -	. 🗌 📀
- Documents	Name *	Date modified	Туре	Size
Avorites     Microsoft     Microsoft Help     prode     Start Menu	def.ppp	6/29/2016 7:23 PM	PPP File	27
Templates     Recovery     System Volume Information     Users	1			
Windows				1
Network	<u>+</u>			

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

	) ~ (~ ~	•						Book2 - <mark>M</mark> icro	osoft Excel	
File	Home	Insert	Page Layout	Formulas	Data	Review	View	Developer	Add-Ins	
Proper	ties -									
Edit	Properties									
Ope	n Archive									
Save	a Archive									
3		-								

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, there are different versions for Windows, Linux and Android, your version may include a tab selector or a tree menu (see herebelow), select Operating tab and stream 8

the second se	lected Stream		8	~			Sa	ve
Components Models	eration to solv	/e	T-P Flash		~		Compute	
onfig Fee	ed(s)	<u>ou</u>	8		~	1 Test Case 1		
	ec. (IN)		288.15	1	K	101327		Pa.a
	ec. (OUT)			2	Pa.a			kW
lodels				Ç.				
icence Str	eam Operating	9			ĸ			Pa.a
Flo	w units		Flows (mole)		~			
Pha	ase	Feed						
Flo	w ( kmol/s )	0.0277778	0	0	0	0	0	0
Fra	action (molar)	1	0	0	0	0	0	0
CH	4	0.6	0	0	0	0	0	0
C2	H6	0.2	0	0	0	0	0	0
C3	H8	0.2	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0

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

LF (0-1)	METHA	NE Co	
nts Sorting criteria		/ first name	
-			
Add	Remove		Clear
Units	Molar flow		
Flow (stream)	0.0277778		kmol
Reaction set	Reaction set 1		
Chemical equations		Balance	
	mponent	Reaction set 1	and the second se
METHANE	nponent	No	0.6
METHANE	mponent	No v	0.6
METHANE	mponent	No Vo Vo Vo No Vo	0.6 0.2 0.2
METHANE ETHANE	mponent	No         >           No         >           No         >           No         >	0.6 0.2 0.2 0
METHANE ETHANE	mponent	No         >           No         >           No         >           No         >           No         >	0.6 0.2 0.2 0 0
METHANE ETHANE	mponent	No         >	0.6 0.2 0.2 0 0 0 0
METHANE ETHANE	mponent	No         >	0.6 0.2 0.2 0 0 0 0 0 0
METHANE ETHANE	mponent	No         >	0.2 0.2 0 0 0 0 0 0 0
METHANE ETHANE	mponent	No         >           No         >	0.6 0.2 0.2 0 0 0 0 0

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

ream Operating	LF (0-1)	METHA	NE	
Components Models	Sorting criteria	Sort by	first name	
BIPs onfig hemicals	Add	Remove	C	Jear
Ps	Units	Molar flow		
odels	Flow (stream)	0.0277778	ß	kmol/s
cence	Reaction set	Reaction set 1		
	reduition set	reaction set 1		
	Chemical equations Compose		Balance Reaction set 1	Molar fraction
	Chemical equations Compose		Reaction set 1	
	Chemical equations Compore METHANE		Reaction set 1	0.6
	Chemical equations Compose		Reaction set 1	0.6 0.2
	Chemical equations Compose METHANE ETHANE		Reaction set 1       No     V       No     V	0.6 0.2 0.2
	Chemical equations Compose METHANE ETHANE		Reaction set 1       No     V       No     V       No     V	0.6 0.2
	Chemical equations Compose METHANE ETHANE		Reaction set 1           No         ✓           No         ✓           No         ✓           No         ✓           No         ✓	0.6 0.2 0.2 0
	Chemical equations Compose METHANE ETHANE		Reaction set 1       No     >       No     >       No     >       No     >       No     >       No     >	0.6 0.2 0.2 0 0
	Chemical equations Compose METHANE ETHANE		Reaction set 1       No     >	0.6 0.2 0 0 0 0 0 0 0
	Chemical equations Compose METHANE ETHANE		Reaction set 1       No     V       No     V	0.6 0.2 0 0 0 0 0 0 0 0 0

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

- Operating	3	Predefined packa	ges	5 PRXIVE	W)	~	PRX(VDW	V)			Save	-
- Components - Models - BIPs				C								
Config				Vapor		Liq	uid		Solid		Hydrate	
Chemicals		Fugacity	PRX(	VDW)	Y	PRX(VDW)		~	SPRX-NRTL(P-HV)	Y	HPRX-NRTL(P-HV)	1
BIPs		Enthalpy	PRX(	VDW)	¥	PRX(VDW)		~	REGULAR	~	HPRX-NRTL(P-HV)	
- Models	De	Entropy	PRX(	VDW)	~	PRX(VDW)		~	REGULAR	~	HPRX-NRTL(P-HV)	>
- Licence	~	Volume	PRX(	VDW)	V	PRX(VDW)		~	REGULAR	~	HPRX-NRTL(P-HV)	~
		Multiphase equilib Multiphase initializ					Multiphas Standard		apor-liquid ts			
							In other the other state		March 1990 March			
			ation				Standard	tes	March 1990 March	and L	.iq.Dens.	-
		Multiphase initializ Detect Phase Stat Phase diagram, cl	ation te heck stability a	-			Standard From Gibt	tes bs (	its	and L	.iq.Dens.	-
		Multiphase initializ Detect Phase Stat	ation te heck stability a	-	15		Standard From Gibt Discard u	tes bs i inst	its or Isothermal Compr. (			

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	No multiphase, only two-phases	~
Multiphase initialization	Standard tests	~
Detect Phase State	From Gibbs or Isothermal Compr. and Liq.Dens.	~
Phase diagram, check stability against feed	Accept all solutions	~
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 structures 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

erating Predefined pac mponents	kal⊋s	30		~	Test	_			Save	
dels s		Vapor		Lit	quid		Solid		Hydrate	
cals Fugacity	PR(V	DW)	~	PR(VDW)			SPRX-NRTL(P-HV)	~	HPRX-NRTL(P-HV)	~
Enthalpy	PR(V	DW)	~	PR(VDW)		~	REGULAR	~	HPRX-NRTL(P-HV)	~
Entropy	PR(V	DW)	~	PR(VDW)		~	REGULAR	~	HPRX-NRTL(P-HV)	~
Volume	PR(V	DW)	~	PR(VDW)		~	REGULAR	~	HPRX-NRTL(P-HV)	~
Detect Phase S Phase diagram,	check stability a	gainst feed			Accept all	-	or Isothermal Compr. a olutions			2 2
	specified phase	-				-	rossing phase bounda	in li	nee	2 2
Hydrate structu		in a contraint				_	al structures generate			-

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

eam Operating	Edit BIPs				Use edited BIPS			
Components	Select BIPs D	ata Set			VLE BIPs			
Models	Get BIPs					Get BIPs from da	atabase	
BIPs	Select the mo	del			PRX(VDW)			
nfig	1							
emicals								
s idels	C1	C2	K12	K12(T)	K12(T2)	L12		
ence	1	2	-0.0149816	0	0	0.0416792		
	1	3	0.0189855	0	0	0.00640015		
	2	3	-0.0222949	0	0	-0.0252287		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	0	0	0	0	0		
	0	n	0	n	n	n		
	<							1

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

and the second second second second	ected Stream		8	~			Sav	e
- Components - Models	eration to solv		T-P Flash		1000		Compute	
BIPs		/e	8		~ ~	1 Test Case 1	Compute	15
ontig	ed(s)		288.15		ĸ	101327		Pa.a
ricificals	ec. (IN)		200.15		Pa.a	101327		
	ec. (OUT)				Pala			kW
lodels icence								10000
Sue	eam Operating	9			к			Pa.a
Flov	w units		Flows (mole)		~			
Pha	ISC	Feed						
Flow	w ( kmol/s )	0.0277778	0	0	0	0	0	0
Frad	ction (molar)	1	0	0	0	0	0	0
CH4	4	0.6	0	0	0	0	0	0
C2H	16	0.2	0	0	0	0	0	0
СЗН	18	0.2	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0

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

erating Selected Strea	m	8	~			Save	
mponents	**					0010	
dels Operation to ac	lve	T-P Flash		~		Compute	II
5 Ecod(c)		8		~	1 Test Case 1	ana ana amin'ny solar	~
cals Spec. (IN)		200		ĸ	5		bar.a
Spec. (OUT)				Pa.a			kW
					1		
Stream Operati	ng	200	2	к	500000		Pa.a
Flow units		Flows (mole)		~			1
Phase	Feed	Vapor	Liquid	Not present	Not present	Not present	Not prese
Flow ( kmol/s )	0.0277778	0.019494	0.00828377	0	0	0	0
Fraction (molar	) 1	0.701784	0.298216	0	0	0	0
CH4	0.6	0.821153	0.0795662	0	0	0	0
C2H6	0.2	0.149549	0.318726	0	0	0	0
СЗН8	0.2	0.0292984	0.601708	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

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

Operating	Pressure	Pa.a	~
- Components	Pressure (dp)	Pa	~
Models	Temperature	ĸ	~
- BIPs	Temperature (dt)	к	~
onfig	Calorific Value	kJ/kg	~
- Units - Settings	Calorific Value (molar)	kJ/kmol	~
nemicals	Enthalpy (Streams)	kW	~
Ps	Entropy (Streams)	kJ/(K*s)	~ ~
odels	Heat Capacity	kJ/(kg*K)	~
ence	Heat Capacity (molar)	kJ/(kmol*K)	~
	Flow (mass)	kg/s	~
	Flow (gas, mass)	kg/s	~
	Density	kg/m3	~
	Density (molar)	kmoVm3	~
	Specific Volume	m3/kg	~
	Specific Volume (molar)	m3/kmol	~
	Thermal Conductivity	W/(m*K)	~
	Viscosity (dynamic)	Pa*s	~
	Surface Tension	N/m	~
	Lenght	m	~
	Area	m2	~
	Volume	m3	~
	Happ	ka	

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 🔫 🤇	° ∫x	=EStrLf	(8,B1,B2)		B4 ▼ (	f <sub>x</sub>	=EStrL[	0(8, <mark>B1,</mark> B2)
2	A	В	С	D	1	A	В	С	D
1	Temperature	150			1	Temperature	150		
2	Pressure	5			2	Pressure	5		
3	Liquid Fraction	0.69282			3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395			4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528			5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377			6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965			7	Gas Heat Capacity	2.208965		

	B5 👻 🔍	fx fx	=EStrLc	p(8,B1,B2)		B6 🔫 🤇	• fx	=EStrG	D(8,B1,B2)
1	A	В	С	D		A	В	С	D
1	Temperature	150			1	Temperature	150		
2	Pressure	5			2	Pressure	5		
3	Liquid Fraction	0.69282			3	Liquid Fraction	0.69282		
4	Liquid Density	543.5395			4	Liquid Density	543.5395		
5	Liquid Heat Capacity	2.399528			5	Liquid Heat Capacity	2.399528		
6	Gas Density	6.99377			6	Gas Density	6.99377		
7	Gas Heat Capacity	2.208965			7	Gas Heat Capacity	2.208965		

	B7 💌	fx fx	=EStrG	cp(8,B1,B2)
4	A	В	С	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.

ng Selected Stream		8		+		S	ave
nents			1	-		-	
Operation to sol	/e	T-P VL		•	-	Compute	
Feed(s)		8			1 Test Case 1		
Spec. (IN)		170		K	3		bar.a
Spec. (OUT)				bar.a	J		kW
Stream Operatin	g	170		к	3		bar.a
Flow units		Flows (volume)					0
Phase	Feed	Vapor	Liquid	Not present	Not present	Not present	Not present
Flow (m3/s)	0.0757012	0.0750186	0.00068263	0	0	0	0
Fraction (molar)	1	0.59355	0.40645	0	0	0	0
CH4	0.6	0.93628	0.10892	0	0	0	0
C2H6	0.2	0.0598366	0.404685	0	0	0	0
C3H8	0.2	0.00388291	0.486396	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
•							

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 • (	fs	=getT	(8)
1	A	В	С	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
б	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 D5 enter =StrLD(8) to obtain inquid 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

Operating	Selected Stream		8 test		test		S	ave
Components Models	Operation to solv		H-P VLL		Ŧ			
BIPS		2	8 test			1 Test Case 1	Compute	
onfig	Feed(s)		8 test 270			50		
Units	Spec. (IN)		5		K	50		bar.a
Settings iemicals	Spec. (OUT)		5		bar.a			kW
Ps	Stream Operating	1	220 869		к	500000		Pa.a
Data	Flow units		Flows (volume)					1 d.d
Regress	1							
odels	Phase	Feed	Vapor	Liquid	Not present	Not present	Not present	Not present
ence	Flow (m3/s)	0.0832156	0.0829232	0.000292377	0	0	0	0
	Fraction (molar)	1	0.849711	0.150289	0	0	0	0
	CH4	0.6	0.698364	0.0438658	0	0	0	0
	C2H6	0.2	0.198811	0.206724	0	0	0	0
	C3H8	0.2	0.102825	0.74941	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0

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 💌 🤄	$f_{x}$	=getT(8)	
1	A	В	С	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

ating Selected Strea	m	9 Test Case 4		Test Case 4		S	ave
oonents					N		
Operation to so	ilve	Mixer				Compute	
Feed(s)		4 Test Case 4		<u>_</u>	5 COLUMN FEI	ED 1	
Spec. (IN)				K			bar.a
Spec. (OUT)		2		bar.a			kW
Stream Operat	22	256.28		K	2		bar.a
Flow units	ng	Flows (mole)		К	2		par.a
					J	1	
Phase	Feed	Vapor	Liquid	Solid	Not present	Not present	Not present
Flow ( kmol/s )	0.0648143	0.0562894	0.00761231	0.000912637	0	0	0
Fraction (mola	· · · · · · · · · · · · · · · · · · ·	0.868471	0.117448	0.0140808	0	0	0
CH4	0.581108	0.667679	0.0106232	3.81986e-013	0	0	0
C2H6	0.0952145	0.107486	0.0158905	1.47052e-014	0	0	0
C3H8	0.11439	0.118615	0.0968662	6.23398e-013	0	0	0
CO2	0.0149002	0.0169151	0.0017875	5.37252e-012	0	0	0
C4H10	0.043778	0.0384516	0.0884129	0	0	0	0
H2O	0.0149002	0.000714668	0.00169212	1	0	0	0
C4H10	0.0547597	0.0422671	0.153702	0	0	0	0
C6H14	0.0809492	0.00787195	0.631025	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
•							

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

Selected Stream	n	6 Test Hydrate		▼ Test Hydrate		c	ave
ents		o restriyurate	3	Test Hydrate		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	dve
Operation to so	ve	T-P Flash			•	Compute	
Feed(s)		6 Test Hydrate	i.		• 1 Test Case 1	48	
Spec. (IN)		250		ĸ	5		bar.a
Spec. (OUT)				Pa.a	1		kW
Stream Operati	ng			к	1		Pa.a
Stream Operati	ıg			K			Pa.a
Flow units		Flows (mole)			<u> </u>		
Phase	Feed	Not present	Not present	Not present	Not present	Not present	Not present
Flow ( kmol/s )	0.0558521	0	0	0	0	0	0
Fraction (molar)	1	0	0	0	0	0	0
CH4	0.90375	0	0	0	0	0	0
C2H6	0.05	0	0	0	0	0	0
C3H8	0.02	0	0	0	0	0	0
0.5110						-	-
CO2	0.02	0	0	0	0	0	0
	0.02	0	0	0	0	0	0

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

ream					I power :			-	_
Operating	Predefined packages			PRXCPA-NRTL(P-HV) with hy	PRXCPA	-NRT	L(P-HV) with hydrate	Save	
Components Models BIPs			2 3	SRK(VDW) SRKX (VDW) PR (VDW)	-				
onfig				PRX (VDW) PRX-NRTL(P-HV) with hydrate	uid		Solid	Hydrate	
nemicals	Fugacity	PRX		SRKX-NRTL(P-HV)	-(P-HV)	•	SPRXCPA-NRTL(P-HV)	HPRXCPA-NRTL(P-HV)	
Ps	Enthalpy	PRX	-NI 7	PRXCPA(VDW) SRKXCPA(VDW)	HV)	-	REGULAR	HPRX-NRTI (P-HV)	-
odels cence	Entropy	PRX	NI 9	PRXCPA-NRTL(P-HV) with hy	HV)	÷	REGULAR	HPRX-NRTL(P-HV)	
ence	Volume	PRX	CF 10	0 Lee Kesler Plocker 1 Benedict-Webb-Rubin-Starling	-(P-HV)	-	REGULAR	HPRX-NRTL(P-HV)	Ŧ
	Detect Phase State Phase diagram, check	stability ag	air 2				r Isothermal Compr. and L ble solutions	iq.Dens.	
	Phase diagram, check	stability ag							-
	Phase diagram, specifi	ed phase fr		2 WILSON 3 UNIQUAC	End when	n cro	ssing phase boundary line	S	and the second
	Hydrate structures inclu	usion		4 UNIFAC	Include n	orma	I structures generated by	formers	Contraction of the second
			20	9	J				
							ОК	Cancel A	opl

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

Operating	Edit BIPs				Use edited BIP	5			*
Components	Select BIPs D	ata Set			Hydrate BIPs				-
Models	Get BIPs					Get BIPs f	rom database		
BIPs	Select the mo	del			PRXCPA-NRTL	(P-HV)			-
nemicals									
Ps									
odels cence	C1	C2	K12	U12	U21	U12 T	U21 T	A12	
Lence	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

- Stream						
- Operating	LF (0-1)	METHAN	E	~		
Components	Sorting criteria	Sort by	first name			
Models	-					
BIPs	Add	Remove	C	Jear		
- Config				().633.947		
Settings	Units	Molar flow				
- Chemicals	Flow (stream)	0.0558521	N	kmol/s		
BIPs	Reaction set	Reaction set 1				
Models	Chemical equations		Balance			
Licence						
	Component	at	Practice set 1	Molar fraction		
	Componer	nt	Reaction set 1	Molar fraction		
	METHANE	nt	No	0.906		
	METHANE	nt	No v	0.906 0.05		
	METHANE	nt	No V No V No V	0.906		

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

Selected Stream		6 Test Hydrate		<ul> <li>Test Hydrate</li> </ul>		S	ave	
L	·		·					-
Operation to solv	e	T-P Flash			-	Compute		
Feed(s)		6 Test Hydrate	l.		•			
Spec. (IN)		277		K	15		bar.a	
Spec. (OUT)				Pa.a			kW	
r					1			
Stream Operating	9	277		K	15		bar.a	
Flow units		Flows (mole)			<u> </u>			
Phase	Feed	Vapor	Hydrate	Not present	Not present	Not present	Not present	
Flow ( kmol/s )	0.0558521	0.0555753	0.000276745	0	0	0	0	
Fraction (molar)	1	0.995045	0.00495497	0	0	0	0	
CH4	0.905095	0.909258	0.0691173	0	0	0	0	
C2H6	0.04995	0.0500998	0.0198874	0	0	0	0	
C3H8	0.01998	0.0200028	0.0153972	0	0	0	0	
CO2	0.01998	0.0200626	0.00339169	0	0	0	0	
	0.004995	0.000577008	0.892206	0	0	0	0	1
H2O	0.004995	0.000311000	0.002200		-			

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

itream					
- Operating	LF (0-1)	METHAN	IE		
Components	Sorting criteria	Sort by	first name		
- Models BIPs					
Config	Add	Remove		C	Jear
Units					
Settings	Units	Molar flow			
Chemicals	Flow (stream)	0.0558521			kmol/s
BIPs	Reaction set	Reaction set 1			
Models	Chemical equations		Bala	nce	
icence					
	Compone	nt	Reaction	set 1	Molar fraction
	Compone	nt	Reaction :		Molar fraction 0.90375
		nt	the second	~	
	METHANE	int	No	~	0.90375
	METHANE ETHANE	nt	No No	* * *	0.90375 0.05
		ent	No No No		0.90375 0.05 0.02

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

Operating	elected Stream		6 Test I lydrate		Test I lydrate		0	ave
Components	Selected Stream		o rest riyulate		Test Hydrate			ave
- <u>Models</u>	peration to solve	)	I-P Flash			-	Compute	
BIPs F	eed(s)		6 Test Hydrate	ļ		▼ 1 Test Case 1		
	Spec. (IN)		277		к	30		bar.a
	Spec. (OUT)				Pa.a			kW
Models								
Licence	Stream Operating	E	277		ĸ	3e+06		Pa.a
F	low units		Flows (mole)			<u> </u>		
P	hase	Feed	Vapor	Liquid	Not present	Not present	Not present	Not present
F	low ( kmol/s )	0.0558521	0.0555638	0.000288244	0	0	0	0
F	raction (molar)	1	0.994839	0.00516085	0	0	0	0
C	CH4	0.90375	0.908438	0.000132679	0	0	0	0
C	2H6	0.05	0.0502594	1.24935e-06	0	0	0	0
C	C3H8	0.02	0.0201038	7.43526e-08	0	0	0	0
C	02	0.02	0.0201037	2.82235e-06	0	0	0	0
H	120	0.005	0.000287665	0.913381	0	0	0	0
	H4O	0.00125	0.00080785	0.0864819	0	0	0	0

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)

ating Selecte	d Stream		6 Test Hydrate		Test Hydrate		5	ave
els Operativ	on to solve	e	H-P Flash			1	Compute	
Feed(s)	(		6 Test Hydrate			1 Test Case 1		
als Spec. (I	N)		293.15		ĸ	50		bar.a
Spec. (	OUT)		15		bar.a			kW
Stream	Operating	1	274.716		к	1.5e+06		Pa.a
Flow un			Flows (mole)		-			
Phase		Feed	Vapor	Liquid	Not present	Not present	Not present	Not preser
Flow ( k	mol/s )	0.0558521	0.0555784	0.000273689	0	0	0	0
Fraction	(molar)	1	0.9951	0.00490025	0	0	0	0
CH4		0.90375	0.9082	6.4564e-05	0	0	0	0
C2H6		0.05	0.0502462	6.31677e-07	0	0	0	0
C3H8		0.02	0.0200985	3.85708e-08	0	0	0	0
CO2		0.02	0.0200985	1.42136e-06	0	0	0	0
H2O		0.005	0.000454894	0.92798	0	0	0	0
CH4O		0.00125	0.00090183	0.0719532	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
•								

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)

ł	ile Home	Insert	Page Layout	Formulas	Data	Rev
	Properties *					
Men	u Commands					
	B2		f <sub>x</sub>	=PfTF(2,B1,	0,1,1)	
1	A	В	C	D	E	
1	Temperature	190.20	8			
2	P (1)	16.2391	6			
3	P (2)	30.0487	6			
4	P (3)	44.4912	3			

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:

	File Home	Insert	Page Layout	Formulas	Data	Rev
	Properties *					
Mer	u Commands					
	B2	• (*	$f_{\mathbf{x}}$	=PfTF(2,B1,	0,1,1)	
1	A	В	C	D	E	
1	Temperature	190.	1			
2	P (1)	15.8768	9			
3	P (2)	30.4252	4			
4	P (3)	44.3025	2			

# 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)

	File Home In	isert Page Layout	Formulas Data R	eview Vie	w Develope	r Add-I	Ins
	Arial	- 10 - A	_A <sup>*</sup> ≡ ≡ ≫ -		Number	.*	
Pa		U •   🖽 •   💩 •		💷 -	\$ • % •	•.0 .00 .00 →.0	Conditional Format a: Formatting * Table *
Clip	board 🖫	Font	Alignment	ľ%	Number	l5i	Styles
	B1	▼ ( f <sub>x</sub> 3	1				
	A	В	C	C	)	Е	F
1	Stream	3		Note : y	ou may nee	d to load	d the add-in prope
2	Temperature	187.0000	K	1) from	Properties I	nenu se	lect the editor and
				2.2			
3	Pressure	4154420.0000	Pa.a	rememb	er to prope	rly set th	ne multiphase opti
3	Pressure Errors	4154420.0000 No errors	Pa.a		25 85	1.1	ne multiphase opti e units defined in e
	- Constant of the second		Pa.a		25 85	1.1	25
4	Errors				25 85	1.1	25
4 5	Errors	No errors			25 85	rding the	25
4 5 6	Errors	No errors		2) spec	fy p, t accor	rding the	e units defined in ε
4 5 6 7	Errors	No errors	shatp,t	2) spec	fy p, t accor	rding the	e units defined in e
4 5 6 7 8	Errors Comp	No errors ute Isothermal Flas	sh at p, t Molar Fraction	2) spec	fy p, t accor	rding the IIId 104	e units defined in e
4 5 6 7 8 9	Errors Comp	No errors ute Isothermal Flas Component	sh at p, t Molar Fraction Formula	2) spec Feed 1.0000	fy p, t accor Liqu 0.79	rding the lid 04	e units defined in e Liquid 0.2096
4 5 6 7 8 9 10	Errors Comp	No errors ute Isothermal Flas Component METHANE	sh at p, t Molar Fraction Formula CH4	2) spec Feed 1.0000 0.9000	ify p, t accor Liqu 0.79 0.99	rding the lid 04	e units defined in ∈ Liquid 0.2096 0.5442
4 5 7 8 9 10 11	Errors Comp	No errors ute Isothermal Flas Component METHANE	sh at p, t Molar Fraction Formula CH4	2) spec Feed 1.0000 0.9000	ify p, t accor Liqu 0.79 0.99	rding the lid 04	e units defined in ∈ Liquid 0.2096 0.5442

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.

Pas	- 6 -	- 10 - A A ⊻ -   ⊞ -   3 - Font		計 Num	nber - - % •	Conditional Forr Formatting + Ta Style	ble * Styles *	ia Insert Insert I Delete II Forma Cells
	B2	• (* f* 7	70		14			
A	A	В	С	D	E	F	6	3
1	Stream	4			may need to loa	-		
		070 0000		A) france Dece	operties menu se	lact the editor	and define c	composit
2	Temperature	270.0000	K	and the second second second	the second second the second second second			and the second sec
3	Pressure	500000.0000	Pa.a	remember	to properly set t	he multiphase	options whe	n multip
3 4	Carlos and the second second			remember	the second second the second second second	he multiphase	options whe	n multip
3 4 5	Pressure Errors	5000000.0000 No errors	Pa.a	remember	to properly set t	he multiphase	options whe	n multip
3 4	Pressure Errors	500000.0000	Pa.a	remember	to properly set t o, t according the	he multiphase e units defined	options whe	n multip
3 4 5 6	Pressure Errors	5000000.0000 No errors	Pa.a	remember 2) specify p	to properly set t	he multiphase	options whe I in editor the	n multip
3 4 5 6 7	Pressure Errors	5000000.0000 No errors	Pa.a hatp,t	remember 2) specify p Feed	to properly set t o, t according the Vapor	he multiphase e units defined Liquid	options whe I in editor the Solid	n multip
3 4 5 6 7 8 9	Pressure Errors	5000000.0000 No errors ute Isothermal Flas	Pa.a h at p, t Molar Fraction	remember 2) specify p Feed	to properly set t o, t according the Vapor	he multiphase e units defined Liquid	options whe I in editor the Solid	n multip
3 4 5 6 7 8 9 10	Pressure Errors	5000000.0000 No errors ute Isothermal Flas	Pa.a h at p, t Molar Fraction Formula	remember 2) specify p Feed 1.0000	to properly set t b, t according the Vapor 0.9561	he multiphase e units defined Liquid 0.0241	options whe l in editor the Solid 0.0198	n multip
3 4 5 6 7 8 9 10 11	Pressure Errors	5000000.0000 No errors ute Isothermal Flas Component METHANE	Pa.a h at p, t Molar Fraction Formula CH4	remember 2) specify p Feed 1.0000 0.7800	Vapor 0.9561 0.8085	he multiphase e units defined Liquid 0.0241 0.2896	options whe l in editor the Solid 0.0198 0.0000	n multip
3 4 5 6 7 8 9 10 11 11	Pressure Errors	5000000.0000 No errors ute Isothermal Flas Component METHANE ETHANE	Pa.a Molar Fraction Formula CH4 C2H6 C3H8	remember 2) specify p Feed 1.0000 0.7800 0.1000	Vapor 0.9561 0.8085 0.1002	he multiphase e units defined Liquid 0.0241 0.2896 0.1722	Solid           0.0198           0.0000           0.0000	n multip
3 4 5 6 7 8	Pressure Errors	5000000.0000 No errors ute Isothermal Flas Component METHANE ETHANE PROPANE	Pa.a Molar Fraction Formula CH4 C2H6 C3H8	remember 2) specify p Feed 1.0000 0.7800 0.1000 0.0500	Vapor 0.9561 0.8085 0.1002 0.0459	he multiphase e units defined 0.0241 0.2896 0.1722 0.2517	Solid         0.0198           0.0000         0.0000           0.0000         0.0000	n multip

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.

1	A	В	С	D	E	F	G	Н	1	J	K	L	М
1	Stream	1				Note : you m	av need to l	oad the add-	n properties	xla to insta	Il the custo	m menu for	r propert
2	t. min	100 P	<			1) from Prop							
3	t. max	300		Compute	2	options for t				A CONTRACTOR OF THE		1000	
4		100000 F	10. The second se			2) on this pa	-				-		the table
4 5	pressure	TUUUUU	aa			z) on this pa	ge specily o	peraung pres	sure and a	ange of ten	iperatures t	o compute	the table
6	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
7	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
8	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
9	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
10	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
11	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
12	211.11	0.00	0.00E+00	1.17E+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
13	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
14	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
15	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
16	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
17													
18	Result :	No errors											
19	Notes :	errors may be o	priginated whe	n accessing th	e chemical's da	ta base with wr	ong parameter	s					
20	2 10 10 mg	i.e. calculating li	10 C				51						
21		ALL AND ALL AN	to the second second	132 V 80 80									
		above critical po	Jint etc. Il Tou	ind enois inni	t the temperatu	re range.							
22													
23					and the second second								
24		Liquid fraction	vs tempera	ture (click on	rectangle)			10 mm					

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.

q 👗	hasenv.xls	[Read-Only] [Compa	atibility Mode] -	Microsoft	Excel									- d	er 23
1	A	В	С	D	E	F	G	Н		J	K	L	M	N	
5		Stream	1		co	mpute phase	e diagram					Pc (2)			
6		Phase Fract.	0.30	85				-	5			Tc (2)			
7												CricoT	258.639	K	-
8												CricoP	9470711.000	Pa.a	-
9															
10												PHASE	DIAGRAM		
11	10000	000.00000										BP-DP		BP-DP	
12	10000	000.000000										Vap Liq Liq		Vap Liq	
13									4			к	Pa.a	к	F
14	90000	000 00000								1					
15										1		111.952990	101327.400000	181.666717	-
16	8000	000,00000										118.331776	163273.669469	193.335516	2
17	0000	000.000000							11			123.394276	229258.610426	203.460516	4
18												130 144276	344223 083486	208 523016	£
19	7000	000.00000							11			135.206776	453064.403602	213.585516	7
20												142.800526	656452.543031	219.913641	ę
21	6000	000.000000							//			151.597401	957903.486577	228.710516	
22												156.597401	1163578.588302	233.710516	-
23	5000	000.00000						/					1395881.352050	239.673141	- 1
24	50000	000.00000						-/	/ /				1655682.007749	241.328904	2
25 26 27								11	(				1943273.665417	242.861006	2
26	4000	000.00000						-//					2258482.788146 2276794.832354	244.282814 245.605268	3
28								11				1/0.0/51/0	2210194.032354	246.837428	-
29	3000	000.00000						/ /	/					247.986849	-
30	30000	000.000000					1	/						249 059883	-
31							/	/	1					250.061891	2
32	2000	000 00000					//		/					250.997409	2
33							11		1					251.870290	1
34 35	1000	000.00000					//		/					252.683774 253.440598	4
36						$\sim$		/	/					253.440598	
36 37		0.000000				//	)							254.792922	
	► H Fo	glio1							•		10				× i

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	No multiphase, only two-phases	~
Multiphase initialization	Standard tests	~
Detect Phase State	From Gibbs or Isothermal Compr. and Liq.Dens.	~
Phase diagram, check stability against feed	Accept all solutions	~
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

p 🕺	hasenv.xls	[Read-Only] [Compa	atibility Mode] -	Microso	ft Excel										- 6	F 23
4	A	В	С	D	E	F	G	Н	1		J	K	L	M	N	
5		Stream	4		С	ompute ph	ase diagram	1					Pc (2)			
6		Phase Fract.	0.05	85			and a subset of the second						Tc (2)			
7													CricoT	467.400	K	
8													CricoP	100012634.233	Pa.a	12
8 9															and the second	
10													PHASE	DIAGRAM		_
11													BP-DP		BP-DP	
12	120000	0000.00000											Vap Liq Liq		Vap Liq	
13													K	Pa.a	ĸ	F
14														1.4.4		
15													113 658026	101327,400000	290,739561	4
16	100000	000000.0000									1			148234.453723	351.481487	2
17														211683.617936	366.714752	5
18											1			299810 459950	376 249955	2
and a second														411969.875315	383.239411	
19	80000	000000000									1					Ĵ
20											1			566920.839368	388.785738	
21											1			721065.977961	393.411267	
22														902099.824119	397.405263	1
23	60000	0000000000								/				1149852.686175	400.943877	1
24 25										1				1395484.960538 1673344.710936	404.141427 406.818181	2
25										1				1984538.845093	409.319193	4
27										1				2329789.642109	411.675503	4
28	10000	0000 000000								/				2709350,482709	413.910731	-
29	40000	000000000								1				3342135.520814	416.043219	3
30										1				3998360.061195	418.087447	3
31										1				4198360.061195	420.055012	3
32										/			200.837998	4398360.061195	421.955336	3
33	20000	0000000000							/				202.665622	4598360.061195	423.796170	3
34									/				204.458588	4798360.061195	425.583961	2
35									/				206.221237	4998360.061195	427.324129	2
36						/			1					5198360.061195	429.021273	2
37		0 00000	a n in 2 in	2 2 2 2			/		-	ingen in	ica 2 id		209.671806	5398360.061195	430.679331	27
14 4	F H Fog	glio1 🥂 🕽								4		101			)	

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,

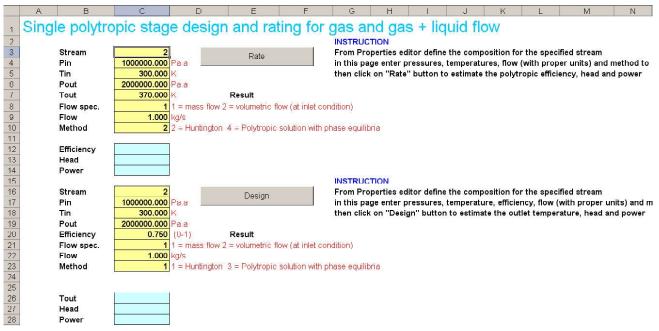
enter 2 as stream, 0.01 as liquid fraction then click on button "Compute phase diagram" to generate the graph

i sil	А	В	С	D	E	F	G	Н	Ť	J	K	L	М	N	
4			<b></b> ]	-				T.				Tc (1)	191.366	K	
5		Stream	2		COL	pute phase	e diagram					Pc (2)			
6		Phase Fract.	0.01									Tc (2)			=
7												CricoT	191.366	K	
8												CricoP	4671132.833	Pa.a	
9															-0
10												PHASE	DIAGRAM		
11	500000	0.00000										BP-DP		BP-DP	
12	500000	0.00000										Vap-Liq		Vap-Liq	
13												к	Pa.a	К	Pa
14	450000	0.000000							T						
15												111,596909	101327,400000	167.311099	10
16	400000	0 000000										117 894497	165273.130603	172,915542	19
17	400000	0.00000							//				249498,246808	177,978042	34
18													344345.881005	183.040542	
19	350000	0.000000						/	/ \				533614.406378	188.103042	11
20													789903.787901	189.164481	13
21	200000	0.000000						//					1001216.702757	189.951062	15
	300000	0.00000											1250573.496892	190.514820	15
22 23								//					1541680.653292	190.514620	17
24	250000	0.000000						/					1878321 089527	190.009250	21
25								//					2264352.826633	191.147103	23
25 26 27	000000	0.000000						//					2862666.872286	191,149014	23
27	200000	0.000000						//				180.111921	3313360.132833	191.048707	25
28								//				181.927831	3513360.132833	190.799166	27
29 30	150000	0.000000						//					3713360.132833	190.389204	29
							1	//					3913360.132833	190.799119	27
31	100000	0.000000					11		1			186.980957	4113360.132833	189.799444	31
32	100000	0.00000					//		1					188.998470	33
33 34							//		/					187.969280 187.167849	35 37
34	50000	0.000000					/	/	-					187.167849	37 40
36						/	/							188.847091	
14 4	► H Fog	lio1 /ta				-		<i>(</i>	•		80			100.041031	42 *

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

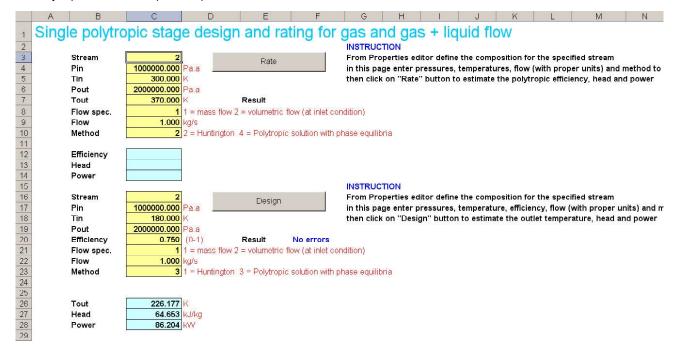


the page contains two sections, the first permits to calculate the polytropic efficiency of a single compression stage given the inlet temperature and pressure.

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

-			-	5765			1					
1	A	B	C	D	E	F	G	H	1	J	K	L
4	select the most suitable mod	del (1 = HEM, 2 =	HNE, 3 = HNE	-DS, 4 = NHN	IE) and the param	eter (when re	quired)					
5	the procedure estimates the	(maximum, isent	ropic) nozzle	flux and retu	rns the required a	rea						
6	Stream	5										
7	Model	2	1	1 = HEM, 2	= HNE, 3 = HNE-	DS, 4 = NHNE						
8	Model parameter	0.7500		model para	meter as defined	in operating r	nanual					
9	Pin	2.000E+06	Pa.a				1					
10	Tin	3.400E+02	К		Calculate solution	٦						
11	Pout	1.013E+05	Pa.a									
12	Flow	1.2300	kg/s									
13	Corrections Ka*Kb*K	0.9000	0.3-1	Result :	No errors							
14			20 20									
15	Estimated tout	274.7390	K									
16	Calculated area	4.229E-05	m2									
17	Required Area	4.699E-05	m2									
18												

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

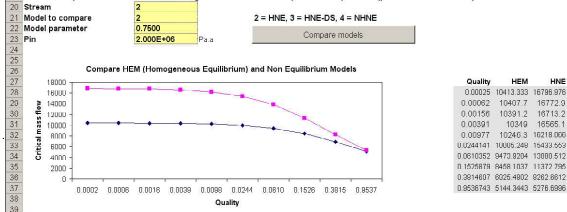
Please follow fhese steps to compare:two models,

1) from Properties editor define the composition, models, BIPs (for mixtures)

2) enter the pressure, model and parameter

3) click on button "Compare Models"

19 You can compare the results of HEM model against another model at specified inlet pressure (pout set at a fraction of pin to evaluate the critical flow)

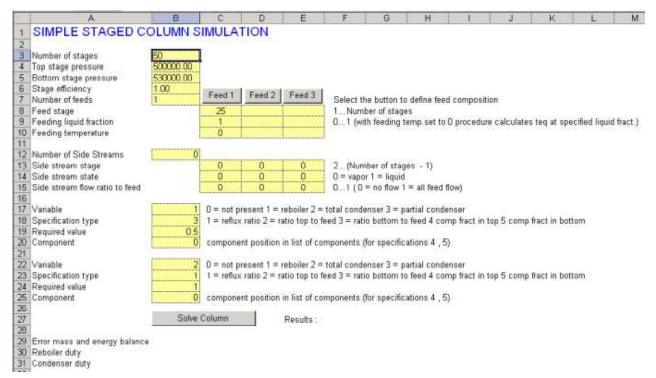


The Non Equilibrium models are mainly of interest for short nozzles where the final equilibrium condition (predicted by HEM models) is not reached cause the residence time of the fluid is too short.

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)

	Solve Column
Error mass and energy balance Reboiler duty	-8.342E-16 399.71527 kW
Condenser duty	312.58773 kW

Results : a numerical solution was found, please verify the results...

61.66 16811 16811 80906 04396 45478 99269 74269 15045 40037 57011 1.6946 79124	530000 500000 500000 500612.2 501224.5 501836.7 502449 503061.2 503673.5 504285.7 504285.7 504285.7 504898 505510.2	29.7486 29.7486 29.7486 28.7518 28.35777 28.15443 28.04284 27.97346 27.97346 27.87466 27.87699 27.87446	0 4.833105 4.833105 0.974546 0.51557 0.450963 0.435465 0.429428 0.426483 0.426483 0.424013 0.424013 0.423533	6.474781 6.439535	0.543766 4.455996 4.455996 8.221962 9.311468 9.23552 8.826832 8.402213 8.052877 7.789158 7.597766	10.31683 2.460342 2.460342 6.44846 9.132649 10.78431 11.80696 12.46848 12.90783 13.20706 13.41315	18.888 0 2.87E-23 8.8E-22 9.7E-21 9.02E-20 7.86E-19 6.65E-18 5.54E-17 4.57E-16 3.75E-15	58 50 58 106 57 903 57 791 57 722 57 675
16811 80906 04396 45478 99289 74269 15045 40037 57011 1.6946	500000 500612.2 501224.5 501836.7 502449 503061.2 503673.5 504286.7 504286.7 504898 505510.2	29.7486 28.7518 28.35777 28.15443 28.04284 27.97346 27.97346 27.92731 27.8959 27.87446	4 833105 0.974546 0.51557 0.450963 0.435465 0.429428 0.426483 0.426483 0.424897 0.424013	17.99915 13.10682 9.397879 7.683638 6.971584 6.673339 6.540123 6.540123 6.474781 6.439535	4 455998 8.221982 9.311468 9.23552 8.826832 8.402213 8.052877 7.789158	2 460342 6 44846 9 132849 10,78431 11,80896 12,46848 12,90783 13,20706	2.87E-23 8.8E-22 9.7E-21 9.02E-20 7.86E-19 8.66E-18 5.54E-17 4.57E-16	59,497 58,50 58,106 57,903 57,791 57,722 57,675
80906 04396 45478 99269 74269 15045 40037 57011 1.6946	500612.2 501224.5 501836.7 502449 503061.2 503673.5 504286.7 504898 505510.2	28.7518 28.35777 28.15443 28.04284 27.97346 27.92731 27.8959 27.87446	0.974546 0.51557 0.450963 0.435465 0.429428 0.426483 0.426483 0.424897 0.424013	13.10682 9.397879 7.683638 6.971584 6.673339 6.540123 6.474781 6.439535	8.221982 9.311468 9.23552 8.826832 8.402213 8.052877 7.789158	6.44846 9.132849 10.78431 11.80896 12.46848 12.90783 13.20706	8.8E-22 9.7E-21 9.02E-20 7.86E-19 6.65E-18 5.54E-17 4.57E-16	69,497 58,50 58,106 57,903 57,791 57,722 57,675
04396 45478 99289 74269 15045 40037 57011 1.6946	501224.5 601836.7 502449 503061.2 503673.5 504265.7 504898 505510.2	28.35777 28.15443 28.04284 27.97346 27.92731 27.8959 27.87446	0.51557 0.450963 0.435465 0.429428 0.426463 0.424897 0.424013	9.397879 7.683638 6.971584 6.673339 6.540123 6.474781 6.439535	9.311468 9.23552 8.826832 8.402213 8.052877 7.789158	9.132849 10.78431 11.80696 12.46848 12.90783 13.20706	9.7E-21 9.02E-20 7.86E-19 8.65E-18 5.54E-17 4.57E-16	58.106 57.903 57.791 57.722 57.675
45478 99269 74269 15045 40037 57011 1.6946	501836.7 502449 503061.2 503673.5 504285.7 504898 505510.2	28.15443 28.04284 27.97346 27.92731 27.8959 27.67446	0.450963 0.435465 0.429428 0.426483 0.426483 0.424897 0.424013	7.683638 6.971584 6.673339 6.540123 6.474781 6.439535	9.23552 8.826832 8.402213 8.052877 7.789158	10.78431 11.80896 12.46848 12.90783 13.20706	9.02E-20 7.86E-19 6.65E-18 5.54E-17 4.57E-16	57.791 57.722 57.675
99269 74269 15045 40037 57011 1.6946	502449 503061.2 503673.5 504285.7 504898 505510.2	28.04284 27.97346 27.92731 27.8969 27.87446	0.435465 0.429428 0.426483 0.424897 0.424013	6.971584 6.673339 6.540123 6.474781 6.439535	8.826832 8.402213 8.052877 7.789158	11.80696 12.46848 12.90783 13.20706	7.86E-19 6.65E-18 5.54E-17 4.57E-16	57.903 57.791 57.722 57.675
74269 15045 40037 57011 1.6946	503061 2 503673.5 504285.7 504898 505510.2	27 97346 27 92731 27 8959 27 87446	0.429428 0.426483 0.424897 0.424013	6.673339 6.540123 6.474781 6.439535	8.402213 8.052877 7.789158	12.46848 12.90783 13.20706	6.65E-18 5.54E-17 4.57E-16	57.722 57.675
15045 40037 57011 1.6946	503673.5 504285.7 504898 505510.2	27.92731 27.8959 27.87446	0.426483 0.424897 0.424013	6.540123 6.474781 6.439535	8.052877 7.789158	12.90783 13.20706	5.64E-17 4.57E-16	57.791 57.722 57.675 57.644
40037 57011 1.6946	504285.7 504898 505510.2	27.8959 27.87446	0.424897 0.424013	6.474781 6.439535	7.789158	13.20706	4.57E-16	57,675
57011 1.6946	504898 505510.2	27.87446	0.424013	6.439535		- ALCONSTRUCTS		
1.6946	505510.2				7.597766	13.41315	3.75E-15	57.644
1.00		27.85993	0.423533					
79124	and the local data in the loca		M. The shake of	6.419117	7.461665	13.55561	3.06E-14	57,623
1 M 1 S 4 1	506122.4	27.85021	0.423306	6.406866	7.366011	13.65403	2.5E-13	57.608
86972	506734.7	27.84387	0.423244	6.39955	7.299302	13.72177	2.03E-12	57.598
93595	507346.9	27.83988	0.423295	6.39641	7.253066	13.76811	1.65E-11	57.592
99376	507959.2	27.83753	0.423423	6.393393	7.221216	13.7995	1.34E-10	57.588
04574	508571.4	27.83633	0.423604	6.392825	7.199435	13.82047	1.08E-09	57.586
09365	509183.7	27.83593	0.423822	6.393254	7.184686	13.83417	8.74E-09	57.584
13872	509795.9	27.83608	0.424066	6.394374	7.17484	13.8428	7.06E-08	57.584
18181	510408.2	27.83662	0.424328	6.395974	7.168409	13.84791	5.7E-07	57.584
22351	511020.4	27.83743	0.424603	6.397904	7.164354	13.85056	4.59E-06	67.585
26426	511632.7	27.83833	0.424884	6.400042	7.161925	13.85145	3.7E-05	57.586
3046	512244.9	27.83869	0.42516	6.40214B	7.160414	13.85067	0.000296	57.586
	13872 18181 22351 26426	13872 509795.9 18181 510408.2 22351 511020.4	13872         509796.9         27.83608           18181         510408.2         27.83662           22351         511020.4         27.83743           26426         511632.7         27.83833	13872         509795.9         27.83608         0.424066           18181         510408.2         27.83662         0.424328           22351         511020.4         27.83743         0.424603           26426         511632.7         27.83833         0.424884	13872         509795.9         27.83608         0.424066         6.394374           18181         510408.2         27.83662         0.424328         6.395974           22351         511020.4         27.83743         0.424603         6.397904           25426         511632.7         27.83833         0.424884         6.400042	13872         509796.9         27.83608         0.424066         6.394374         7.17484           18181         510408.2         27.83662         0.424328         6.395974         7.168409           22351         511020.4         27.83743         0.424603         6.397904         7.164354           26426         511632.7         27.83833         0.424884         6.400042         7.161925	13872         509795.9         27.83608         0.424066         6.394374         7.17484         13.8428           18181         510408.2         27.83662         0.424328         6.395974         7.168409         13.84791           22351         511020.4         27.83743         0.424603         6.397904         7.164354         13.85056           26426         511632.7         27.83833         0.424884         6.400042         7.161925         13.85145	13872         509795.9         27.83608         0.424066         6.394374         7.17484         13.8428         7.06E-08           18181         510408.2         27.83662         0.424328         6.395974         7.168409         13.84791         5.7E-07           22351         511020.4         27.83743         0.424603         6.397904         7.164354         13.85056         4.59E-06           26426         511632.7         27.83833         0.424884         6.400042         7.161925         13.85145         3.7E-05

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

. al	A B	C	DE	F	G	H	Ĵ.	J	K	L	M	N	0	
1	HYDRATE	FORMA	TION CU	RVE										
2	From Properties edite	or define the compo	sition including at	least one h	vdrate for	mer (C1.	C2. C3. I	C4, nC4, 1	V2. CO2. H	2S) and	water amount			
3	Prode hydrate model	s support SI, SII, SI	H structures, you	can specify	different	inhibitors	as Metha	nol, Ethan	ol, Ethylene	glycol plu	s salts (with ele	ctrolyte mode	ls)	
4	Tmin	250 K	Con	pute Hydra	to Formal	ion Cunz	. 1							
5	Tmax	290 K	COIL	ipute nyun	ale runnai		5							
6	Model	1	1 = SI , SII	, SH 2	= SI 3 :	= SII								
7	Stream	6												=
8 9			Errors	Vo errors										
9														
10	245.00	250.00 255.00	260.00 265.0	0 270.00	275.00	280.00	285.00	290.00	295.00		Formation Co	onditions		
11	12000000.00									K 250.00	Pa.a 172999.12			
12 13										250.00	172999.12			
13										253.64	205662.73			
15	40000000.00							1		255.45	225192.57			
16	1000000.00							/		257.27	246062.69			
17							/	/		259.09	269919.24			
18										260.91	297086.51			
19	8000000.00						/_			262.73	330278.28			
20							/			264.55	367575.74			
21							/			266.36	411776.92			
22							/			260.10	494043.80			
23	600000.00					/	/			270.00	745281.84			
24 25						/				271.82	1128352.42			
						/				273.64	1656225.53			
26	4000000.00					/				275.45	2350792.19			
27	400000.00					/				277.27 279.09	3251426.21 4430230.61			
28 29					)	/				279.09	4430230.61 5491754.77			
30					/					282.73	6432516.23			
- 2.5		ś								202.75	5452510.25			Ŧ

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 LibreOffice

LibreOffice (and OpenOffice) include different methods for accessing external libraries as Prode Properties, LibreOffice BASIC allows to access Prode in Windows, while for Linux versions different interfaces (to Prode Properties) are available,

there are several Libre Office pages in folder /Prode/LibreOffice, since LibreOffice (and OpenOffice) include similar features of Microsoft Excel, the LibreOffice pages look (and work) as the equivalent Excel versions, to open the LibreOffice pages, start LibreOffice Calc and Open the page phasenv.xls

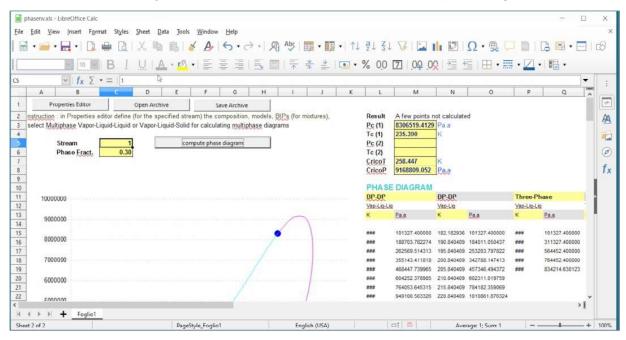
LibreOffice						-		×
Eile Jools Help	11							
Open File	Open						×	
Remote Files	← → → ↑ 🛃 → This PC → CODE (C:) → P	✓ O Sear	rch Libreoffice		P			
- Fairles Fridg	Organize • New folder				168 <b>•</b>		0	
Recent Files -	Microsoft Analysis Services	Name	Date modified	Туре	Size			
Templates 🔻								
1		B multiphase.xls	1/15/2020 1:28 AM					
Create:			1/15/2020 1:30 AM	Microsoft Excel 97				
Writer Dogument		18 phasenv.xls	1/15/2020 1:31 AM	Microsoft Excel 97				
Writer Document		g props.xls	1/15/2020 1:34 AM	Microsoft Excel 97	cel 97 39 KB			
Calc Spreadsheet	Excel	🗐 tpxy.xds	1/15/2020 1:37 AM	Microsoft Excel 97	27 KB			
	Libreoffice							
Impress Presentation	NET .							
		×						
Draw Drawing	[ Read-only							
	File name: phasenv.xls		✓ All	All files (".")		~		
Math Formula			Version	~	Onen	Cancel		
			TC/3/07/1		open	Concer	- 2	
Base Database	pen File ernote Files ernote Files ecent Files ecent Files ernote SQL Server mplates ernote SQL Server ernote SQL Server microsoft SQL Server mi							
Help Extensions								

you may receive a message "LibreOffice Security Warning" : The document contains document macros , click Enable Macros button and proceed

phasenv.xls - LibreOffice Calc								D X
File Edit View Insert Format Styles Sheet Data	a <u>T</u> ools <u>W</u> indow <u>H</u> elp	Rel 1.2d1 License 2	030 suppo	rt ends 20200101 user	r Prode test		×	×
	। । । । : : : : : : : : : : : : : : : :	Chemicals Data ( Stream Operating					Aodels Data Ucense Data Units Config Settings	8
		Selected Stream	1	1 Test Case 1	V Test Case 1		Save	
5 $f_X \Sigma = 1$	E F G H							-
		Operation to sol	ve 1	r-P Flash	U U	C	ompute	1^
1 Properties Editor Open Archive	ST. STREAM ST	Feed(s)	1	Test Case 1		1 Test Case 1	~	2
2 nstruction : in Properties editor define (for the spec		Spec. (N)	2	288.15	ĸ	101327	Pa.a	A
3 select Multiphase Vapor-Liquid-Liquid or Vapor-Liquid	uid-Solid for calculating multiphase diagram	Spec. (OUT)			Pa.a	]	kW	
5 Stream 1	compute phase diagram							
6 Phase Fract. 0.30		Stream Operatin			К	1	Pa.a	
7 8		Flow units	1	flows (mole)	~			
8								fx
9 10		Phase	Feed				^	1.00
11 12		Flow ( kmol/s )		177	0		0	
12		Fraction (molar)	1.1	0	0	C	0	
13		CH4 CO2	0.7	0	0 0		0	
15		H2S	0.15	0	0	5. S.	0	
15 16 10		123	0.15	0	0		0	
17 18			0	0	0	20	0	
18			0	0	0		0	
20 8			0	0	0	0 0	0	
			0	0	0		0 ¥	
22		<					,	
21 22 23 24 6								
ne la						Г	OK Cancel	
4 4 ▶ ▶ + Foglio1			_					
Sheet 2 of 2	PageStyle_Foglio1	English (USA)		=I   🖄	Av	erage: 1; Sum: 1		-+ 100%

the page includes several buttons

Properties Editor opens the editor dialog Open Archive opens a archive Save Archive saves a archive Compute phase diagram calculates the phase diagram for the specified stream



# To calculate the phase diagram define a stream and click the button Compute phase diagram

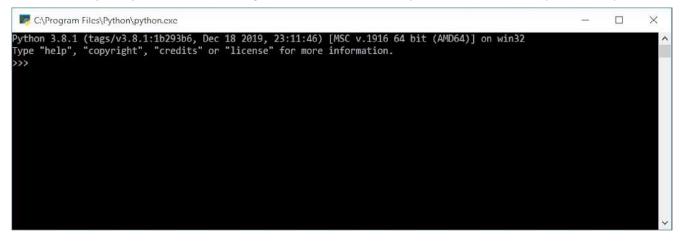
in the same way you can load the pages for solving different problems as discussed in Excel section

• 📾 •			9.6.		• ↔ · A					and the second	Ω·∰			
	10 10		A tokas	• = =		1	≝ <b>!</b>	• % 0,0	21.04	.02 =	三日・	nin • 🚄		
	~ fx Σ • :	=   4			-									-
A	8	C I	D	E		G	н		,	К	L	М	N	^
Prop	perties Editor	Open A	rchive	581	e Archive									
Hoat	Evcha	ndor D	rocos	e Dro	perties									
ileat		igel, r	Toces		pernes									
	SIDE 1			SIDE 2			0 1						8 9	
Stream	4	Edit	Stream	5	Edit	350.00	0 1	2	3	4 5	6	1	8 9	1.00
Flow	8.000 kg/		Flow	10.000		300.00							_	
Pin	200000.000 Pa	a	Pin	500000.000		250.00								
Tin	260.000 K		Tin	320.000		150.00								
dP Tout	50000.000 Pa. 270.000 K	a	dP Tout	50000.000	A CONTRACTOR OF	100.00								
rout	270.000		rout	314.062	ĸ	50.00								
	1	Calculate Prop	erties			0.00								
	L	Concurate 110	ACTURE!			Errors :	No errors							
								LIQUID						_
ZONE	P	÷	T(BUBBLE)	T(DEW)	FLOW	MW	CP	HV	DENS	VISC	тнс	FLOW	MW	
LUNE	Paa	ĸ	K K	K	kais		kJ/(kg*K)	kJ/kg	kg/m3	Pass	W/(m*K)	kg/s		\$410
Inlet	500000	320			6.39176423035026	69.8747336							48.8951541	
	495000 31	9 415756817069	288 976047	260 666206	6.41477397654861	69 8337522	2 43406959	304.613484	579 194203	0.00018617	0.10831215	3 58522602	48 8369297	1.893

# Getting started from Python

The different versions of Prode Properties work with the different versions of Python available for Windows, Linux and Android platforms and all the methods exposed by Prode library can be imported in Python applications, in Windows, to install Prode Properties library in your Python application, follow these steps

check whether your Python shell is executing in 32 or 64 bit mode, run Python and read the data (this is 64 bit)



for a 32 bit Python application copy the files from Prode\Python\32 folder for a 64 bit Python application copy the files from Prode\Python\64 folder

copy the file prode.py to your Python application in /Lib folder copy the file \_prode.pyd to your Python application in /DLLs folder

Prode	^	Name	Date modified	Type	Size
L C		🗟 _prode.pyd	1/23/2020 12:16 PM	Python Extension	219 K
Excel		🖗 prode.py	1/23/2020 12:06 PM	Python File	28 K
LIB		h-disease water			
Libreoffice					
NET NET					
Python					
32					
64					
system					
1 Python					
DLLs					
Doc					
include					
Lib					
libs					
📕 tcl					
Tools		$\triangleright$			

to import prode module in Python, type

>>> import prode

then you have access to all the methods exposed by Prode library, to run Prode Properties Editor on stream 1 enter

>>> prode.edS(1)

Program Files\Python\python.exe	Chemicals::Data C Stream::Operating			Regress BIPs::Data tream::Models Str			License::Da nfig::Setting
3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) help", "copyright", "credits" or "license" for more port prode	Selected Stream		1 Test Case 1	V Test Case 1		Save	
ode.edS(1)	Operation to solv	/e	T-P Flash	~		Compute	
	Feed(s)		1 Test Case 1	~	1 Test Case		~
	Spec. (IN)		288.15	к	101327	P	a.a
	Spec. (OUT)			Pa.a			w
	Stream Operating	a		к		P	a.a
	Flow units		Flows (mole)	~			
	Phase	Feed					
		0.043584	9 0	0	)	0 0	
	Fraction (molar)		0	0	)	0 0	
	TEXAL CONTRACTOR	0.7	0	0	)	0 0	<u>.</u>
	152.01	0.15	0	0		0 0	2
	H2S	0.15	0	0	)	0 0	1
		0	0	0	-	0 0	
		0	0	0	·	0 0	
		0	0	0	)	0 0	
		0	0	0	)	0 0	
		n	0	0	1	0	×
	<						>

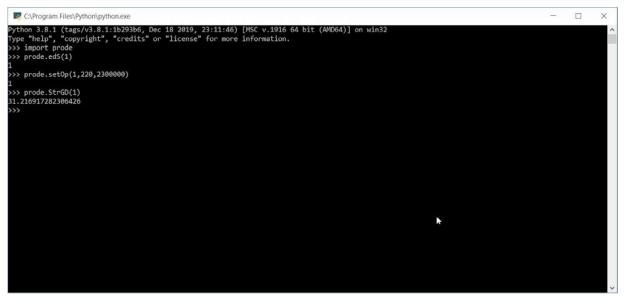
to obtain the gas density at specified temperature and pressure (K,Pa units) for stream 1 enter

>>> prode.setOp(1,220,2300000)

which returns 1 meaning the operation has been solved

>>> prode.StrGD(1)

which returns the value (units Kg/M3) 31.269

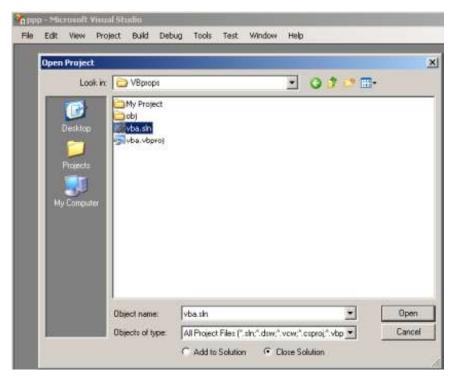


with Python language you can create procedures to calculate and print phase diagrams, solve distillations columns etc. as discussed in Excel paragraph

# Getting started with Microsoft NET (VB , C) applications

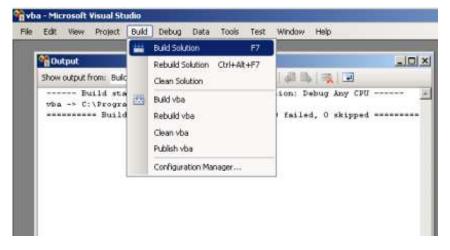
**IMPORTANT** Microsoft NET support files are located in the directory \Prode\NET

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
- a) define a suitable temperature and pressure (with proper units)
   a) click on button Compute Properties to print the properties

Edit View Project Build Debug Data Format To		아이지 이 것이라고.		
	rm.vb [Design]			_101
	nimal VBA applica	ition for Prode Pro	operties	×
E My Project				
- 🛐 AssemblyInfo.vb		1	Yode Properties editor	
- Je ppp.ko	tream code	6	-1	
voa_pas.vo				
😽 vha_frm.vb (Read Only)	mperature	0		
(General) Pi	cessure	0		
ShowPUM.Text =			Compute Properties	
* isothermal f				
res = setOp(St	lar fraction	Liquid	Vapor	
It (res = 0) T		0	þ	
	ensity	0	0	
ShowLF.Text = (	iscosity	0	0	
ShowGF.Text = ( The showLD.Text = ( The showLD		-		
ShowGD.Text =	nermal Cond.	0	0	
	eed od sound	0	0	
ShowDUM. Text =	ule Thomson	10	6	
ShowLV.Text = (		-		
ShowGV.Text = C	2	0	P	
um = 31 ' UM c	*	In	ln l	
ShowLC.Text =				100
ShowGC. Text = 1 1 To	offipt 🥥 Lab	el6 🕜 Label7		
um = 30 ' UM cours to	0.000000			
ShowCUM.Text = getSUM	(S(um)			
ShowLS.Text = CStr(St				
ShowGS.Text = CStr(St		m))		
um = 37 * UM code for	velocity			

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

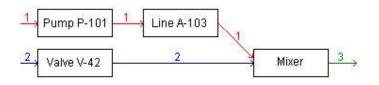
• **flash unit** (enthalpy, entropy or volume basis), calculates output temperature or pressure, with this unit you can simulate pipelines, valves, heat exchangers, pumps, compressors and many others operations.

- 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 projects.

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

Operating Components	Selected Stream		6 Test Hydrate		Test Hydrate		S	ave
Models	Operation to solve	e	T-P Flash		•		Compute	
onfig	Feed(s)		T-P Flash					
Units	Spec. (IN)		PF-P Flash PF-T Flash			40		bar.a
Settings	Spec. (OUT)		H P Flash					kW
iemicals			H-T Flash S-P Flash					
Ps	Stream Operating	1	S-T Flash			40		bar.a
odels	Flow units		V-P Flash V-T Flash					
ence			Copy Stream				1	
	Phase	Feed	Gas Separator			Not present	Not present	Not present
	Flow ( kmol/s )	0.0558521	Liquid Separator Mixer			0	0	0
	Fraction (molar)	1	Mixer 0.99436	0.00563958	0	0	0	0
	CH4	0.90375	0.908873	0.00048084	0	0	0	0
	C2H6	0.05	0.0502835	1.99525e-005	0	0	0	0
	C3H8	0.02	0.0201134	4.65437e-006	0	0	0	0
	CO2	0.02	0.020112	0.000250267	0	0	0	0
	H2O	0.005	0.000224581	0.846993	0	0	0	0
	CH4O	0.00125	0.000393585	0.152251	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0
		0	0	0	0	0	0	0

# 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 LF-T	Phase Fraction, Pressure, Flash Operation Phase Fraction, Temperature, Flash Operation
H-P Flash H-T Flash	Enthalpy, Pressure, Flash Operation Enthalpy, Temperature, Flash Operation
S-P Flash S-T Flash	Enttropy, Pressure, Flash Operation Entropy, Temperature, Flash Operation
V-P Flash V-T Flash	Volume, Pressure, Flash Operation 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.

am Operating	LF (0-1)	WA	TER				
Components	Sorting criteria	Sort	by first name				
Models BIPs frig micals	Add	Remove		C	Clear		
micals	Units	Molar flow					
dels	Flow (stream)	0.0435849	0.0435849				
cence	Reaction set	Reaction set	Reaction set 1				
	Charles and the second second		Balance				
	Chemical equations 1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo		Reaction set		Molar fraction		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) +				Molar fraction		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE		Reaction set	1	0.7		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE		Reaction set Reactant Reactant	1	0.7		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN		Reaction set Reactant Reactant Reactant	1 > > >	0.7 0.2 0.1		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN CARBON DIOXIDE		Reaction set Reactant Reactant Reactant Product	1 > > > >	0.7 0.2 0.1 0		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN		Reaction set Reactant Reactant Reactant Product Product	1 2 2 2 2 2	0.7 0.2 0.1 0		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN CARBON DIOXIDE		Reaction set Reactant Reactant Reactant Product Product No	1 2 2 2 2 2 2	0.7 0.2 0.1 0 0 0 0 0 0		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN CARBON DIOXIDE		Reaction set       Reactant       Reactant       Reactant       Product       Product       No       No	-	0.7 0.2 0.1 0 0 0 0 0 0		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN CARBON DIOXIDE		Reaction set       Reactant       Reactant       Reactant       Product       No       No       No	1 2 2 2 2 2 2 2 2 2	0.7 0.2 0.1 0 0 0 0 0 0 0 0		
	1 (CH4) + 2 (C2H6) + 9 (O2) -> 5 (CO2) + Compo METHANE ETHANE OXYGEN CARBON DIOXIDE		Reaction set       Reactant       Reactant       Reactant       Product       Product       No       No	1	0.7 0.2 0.1 0 0 0 0 0 0		

# 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)

### Define reactions sets (for reactive flash operations)

• select the reaction set and define reactants and products (for the selected components)

#### **Balanca chemical equations**

· select a reaction set, define products and reactants, then click on Balance button to balance chemical equations

- define many different packages with user defined models and options
- set models per each propertty and state (gas,liquid,solid)
- set different options

<ul> <li>Stream</li> <li>Operating</li> </ul>	Predefined packages				1	SRK standar	4	-	Save	-
Components	Predenned packages			SRK standard	-	SRN standar	a		Save	_
Models BIPs ⊕ Config ⊕ Chemicals				2 SRKX (SRK Extended) 3 PR Standard 4 PRX (PR Extended)						
10 Sec. 90				PRX-NRTL-HV (incl. hydrate)	1	uid	Solid		Hydrate	-
Inemicals IIPs	Fugacity	10122	17	6 SRKX-NRTL-HV			SP-PR-NRTL-HV		HYD-PR-NRTL-HV	-
Aodels	Enthalpy	PRI	PR VD 7 CPA-PRX (incl hydrate) 8 CPA-SRK PR VD 9 PRX-GMR PR VD 10 Lee Kesler Plocker			REGULAR	37	HYD-PR-NRTL-HV	7	
icence	Entropy	PR				REGULAR	*	HYD-PR-NRTL-HV	-	
and the second	Volume	PR \	VD	0 Lee Kesler Plocker 1 Benedict-Webb-Rubin-Starling		3	REGULAR		HYD-PR-NRTL-HV	•
	Multiphase initialization Detect Phase State Phase diagram, check stability agai			19 PRX-NRTL LCVM 20 PRX-UNIQUAC LCVM	From Gibbs or Isothermal Compr. and Liq.Dens. Discard unstable solutions				Dens.	61.6
	Detect Phase State Phase diagram, check stability again Phase diagram, specified phase frac		20 PRX-UNIQUAC LCVM							
			2	22 WILSON	End when crossing phase boundary lines					+ +
	11 - 1			S OTHEROPIC	Include normal structures generated by formers					-
	Tryulate structures inci	USION			include normal structures generated by formers					1
			2223333333	21 NR1L 22 WILSON 23 UNIQUAC 24 UNIFAC 26 ISO 18453 (GERG) 26 ISO 20765 (AGA B) 27 STEAM TABLES 28 29 30 30 31 32 33 34 35						

# 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)

• 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

ng E	dit BIPs				Use edited BIPS
	elect BIPs Da	ta Sat			VLE BIPs
	Set BIPs				Get BIPs from database
	Select the mod	ol			UNIQUAC
12	select the mou	ei			UNIQUAC
-					
	C1	C2	U12	U21	
1		2	538.578	-354.352	
0	( <u> </u>	0	0	0	
0		0	0	0	
0		0	0	0	
0		0	0	0	
0		0	0	0	
0		0	0	0	
0	l.	0	0	0	
0	6	0	0	0	
0		0	0	0	
0		0	0	0	
0		0	0	0	
0		0	0	0	
0	(	0	0	0	
0	i.	0	0	0	
	6	0	0	0	

#### **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.

· define the unit of measurement for the different properties

Pressure	Pa.a	
Temperature (dt)	Paa	
Temperature	Pa.g	
Calorific Value	mbar g	
Calorific Value (molar)	KPa.a KPa.g	
Enthalpy (Streams)	bar a	
Entropy (Streams)	bar.g	
Heat Capacity	igficing a igficing g	
Heat Capacity (molar)	psi.s	
Flow (mass basis)	psig atm a	
Flow (gas, mass basis)	atm.g	
Density	mmH2O.a mmH2O.g	
Denotity (molar)	inH20 a	
Specific Volume	inH20.g mmH93.a	
Specific Volume (molar)	mmHG.g	
Thermal Conductivity	V0(mK)	
Viscosity (dynamic)	Pa*s	
Surface Tension	Nên	
Lenght	m	
Area	m2	
Volume	m3	
Mass	kg	
Maloothy	in fo	

# **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.

Operating       Max number of streams       50         Components       Max number of components per stream       50         Models       Max number of interaction coefficents per stream       250         BIPs       Reference temperature (for normal or standard conditions)       288.15       K         Units       Reference pressure (for normal or standard conditions)       101327       Pa.a         Settings       Base value for enthalpy calc.       Specified value and temperature       *         Chemicals       Base temperature for enthalpy       1       K         Models       Base value for enthalpy       5000       kJ/kg         Models       Base value for entropy calc.       Specified value and temperature       *         Chemicals       Base value for entropy calc.       Specified value and temperature       *         Base value for entropy calc.       Specified value and temperature       *       *         Base value for entropy calc.       Specified value and temperature       *       *         Base value for entropy       50       kJ/kg       *       *         Models       Base value for entropy       50       kJ/kg       *         Max allowed time for solving operations       60       s       * <td< th=""><th>Pa.a K</th><th>•</th></td<>	Pa.a K	•
Models       Max number of interaction coefficients per stream       250         BIPs       Max number of interaction coefficients per stream       250         config       Reference temperature (for normal or standard conditions)       288.15       K         Units       Base value for enthalpy calc.       Specified value and temperature       *         chemicals       Base value for enthalpy calc.       Specified value and temperature       *         Chemicals       Base value for enthalpy       5000       kJ/kg         Models       Base value for enthalpy       5000       kJ/kg         Models       Base value for entropy calc.       Specified value and temperature       *         Models       Base value for entropy calc.       Specified value and temperature       *         Base value for entropy calc.       Specified value and temperature       *         Base value for entropy calc.       Specified value and temperature       *         Base value for entropy calc.       Specified value and temperature       *         Base value for entropy calc.       Specified value and temperature       *         Base value for entropy       50       kJ/kg       Specified value and temperature         Base value for entropy       50       Specified value and temperature       * <td>Pa.a K</td> <td></td>	Pa.a K	
BPS       Reference temperature (for normal or standard conditions)       288.15       K         Config       Reference temperature (for normal or standard conditions)       101327       Pa.a         Settings       Base value for enthalpy calc.       Specified value and temperature       *         chemicals       Base temperature for enthalpy       1       K         RPS       Base value for enthalpy calc.       Specified value and temperature       *         Internals       Base value for enthalpy       5000       kJ/kg         RPS       Base value for enthalpy calc.       Specified value and temperature       *         Kodels       Base value for entropy calc.       Specified value and temperature       *         icence       Base temperature for entropy       1       K         Base value for entropy       50       kJ/kg*K)       K         Base value for entropy       50       kJ/(kg*K)       K         Base value for entropy       50       kJ/(kg*K)       K         Base value for entropy       60       s       s         Idowed time for solving operations       60       s       s         Flow units       Mass flows       *       *	Pa.a K	1.
config       Reference pressure (for normal or standard conditions)       101327       Pa.a         Settings       Base value for enthalpy calc.       Specified value and temperature       •         chemicals       Base temperature for enthalpy       1       K         RP's       Base value for enthalpy       5000       kJ/kg         fodels       Base value for enthalpy calc.       Specified value and temperature       •         identicals       Base value for enthalpy       5000       kJ/kg         identicals       Base value for entropy calc.       Specified value and temperature       •         identicals       Base value for entropy calc.       Specified value and temperature       •         identicals       Base value for entropy       1       K         Base value for entropy       50       kJ/(kg*K)       K         Base value for entropy       50       s       kJ/(kg*K)         Convergence tolerance on specifications       1e.009       s       s         Max allowed time for solving operations       60       s       s         Flow units       Mass flows       •       •         Minimum Liquid Density to validate liquid state       Mass flows       •	Pa.a K	•
Base value for enthalpy calc.         Specified value and temperature           Internicals         Base temperature for enthalpy         1           RP's         Base value for enthalpy         6000           Models         Base value for entropy calc.         Specified value and temperature           Jointe         Base value for entropy calc.         Specified value and temperature           Jointe         Base value for entropy calc.         Specified value and temperature           Jointe         Base value for entropy calc.         Specified value and temperature           Jointe         Base value for entropy         1         K           Base value for entropy         50         kJ/(kg*K)         K           Base value for entropy         50         s.         K//(kg*K)           Convergence tolerance on specifications         1e.009         s.         S           Max allowed time for solving operations         60         s.         S           Flow units         Mass flows         V         Mass flows         V	к	-
Base temperature for enthalpy     1     K       Ardels     Base value for enthalpy     5000     kJ/kg       Models     Base value for entropy calc.     Specified value and temperature     •       Base temperature for entropy     1     K       Base value for entropy     1     K       Base value for entropy     50     kJ/kg*/       Convergence tolerance on specifications     1e.009       Max allowed time for solving operations     60     s       Flow units     Mass flows     •       Minimum Liquid Density to validate liquid state     Mass flows     •		
Base value for enthalpy     5000     kJ/kg       Aodels     Base value for entropy calc.     Specified value and temperature     •       Joence     Base temperature for entropy     1     K       Base value for entropy     50     kJ/(kg*K)       Convergence tolerance on specifications     10:009       Max allowed time for solving operations     60     s       Flow units     Mass flows     •       Minimum Liquid Density to validate liquid state     Mass flows		-
Addels     Base value for entropy calc.     Specified value and temperature     *       Base temperature for entropy     1     K       Base value for entropy     50     kJ/(kg*K)       Convergence tolerance on specifications     10:009       Max allowed time for solving operations     60     s       Flow units     Mass flows     *       Minimum Liquid Density to validate liquid state     Mass flows	KJ/Kg	
Base temperature for entropy Calc.     Specified value and temperature       Base temperature for entropy     1       Base value for entropy     50       Convergence tolerance on specifications     1e-009       Max allowed time for solving operations     60       Flow units     Mass flows       Minimum Liquid Density to validate liquid state     Mass flows		
Base value for entropy     50     kJ/(kg*K)       Convergence tolerance on specifications     1e-009       Max allowed time for solving operations     60     s       Flow units     Mass flows     ✓       Minimum Liquid Density to validate liquid state     Mass flows     ✓	44	-
Convergence tolerance on specifications 1e 009 Max allowed time for solving operations 60 s Flow units Mass flows  Minimum Liquid Density to validate liquid state Mass flows		-
Max allowed time for solving operations     60     s       Flow units     Mass flows        Minimum Liquid Density to validate liquid state     Mass flows	kJ/(kg*K)	_
Flow units Mass flows  Minimum Liquid Density to validate liquid state Mass flows		_
Minimum Liquid Density to validate liquid state Mass flows	5	
		*
Tricial Herro		

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

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

- Stream - Config	ACETYLENE		-					
- Chemicals	Sort by first name							
Data	Sort by first name							
- Settings Regress BIPs	Sort by second name Sort by third name Sort by formula							
E-Licence	Name (1)	ACETYLENE	_					
	Name (2)	ETHYNE						
	Name (3)							
	CAS / Identification number	74862						
	Molecular weight	26.0379						
	Critical temperature	308.325	к					
	Critical pressure	6.139e+006 Pa.a						
	Critical volume	0.113	m3/kmoi					
	Acentric factor	0.187642						
	Electric dipole moment.	C-m						
	Radius of gyration	1.0945e-010 m						
	Solubility parameter	590.713	(kJ/m3)1/2					
	Std.Enthalpy form.	226766	kJAmol					
	Gibbs Energy of form.	209940	kJAsnol					
	Enthalpy fusion	3770	kJAonol					
	Normal boiling point	189	K					

#### 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 .

Operating	Nitrogen	7727379
Components	Carbon Dioxide	124389
Models	Methane	74828
BIPS	Ethane	74840
onfig Units	Propane	74986
Settings	isoButane	75285
hemicals	n-Butane	106978
Data	Isopentane	78784
Settings	n-Pentane	109660
Regress	n-Hexane	110543
IPs odels	n-Heptane	142825
cence	n-Octane	111659
	n-Nonane	111842
	n-Decane	124185
	Helium	7440597
	Argon	7440371
	Hydrogen	1333740
	Oxygen	7782447
	Carbon Monoxide	630080
	Hydrogen Sulfide	7783064
	Water	7732185
	neo-Pentane (Dimethyl propane)	463821

# Regress raw data

From this page you can :

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

Property				Vapor heat capacity			
Correlation				v=a+b4+c4^2+d4^3			- U-
				y=a+b*t+c*t*2+d*t*3 y=exp(a+b/t+c*in(t)+			
	1 4	value	low lim	y=a*(1-tr)*b+c*ln(1-t	arrib) r)+d*(1-tr)*3		
a	0		0	y=a*(1-tr)*b+c*(1-tr)	r2+d*(1-tr)*3		
b	0		0	y=a+b*(1-tr)+c*ln(1- y=exp(a+b/t+c*ln(1)+			
c	0		0	y=a+b*t+c*t*2+d*t*3	+6***4		
d	0		0	y=exp(a+b/t+c*in(t)+ y=a*trb/tt+c/t+d/r2			
e	0		0	y=a+b*exp(-c.t.^d)			
t.	0		0	y=a+bA+cA*3+dA*8+ y=a/b*(1+(1-t/c)*6)	e#^9		
				y=a*(1-tr)*(b+c*tr+d	"tr"2+e"tr"3)		
1.0				y=a+b*(c/t/sin(c/t))*			
		perature		y=a*2/(1-tr)+b-2*a*c	*(1-tr)-a*d*(1	-tr)*2-c*2*(1-tr)*3/3	-c*d*(1-tr)
Point 1	210	perature K	1.2	y=a*2/(1-b)+b-2*a*c y=exp(a+bA+c*log(1) y=a+b*(1-b)*0.35+c	^(1-tr)-s*d*(1 +d*1*2+e#*2) *(1-tr)*(2/3)+	-tr)*2-c*2*(1-tr)*3/3	2000.000000 
Point 1 Point 2		11 342	1.3	y=a*2/(1-tr)+b-2*a*c y=exp(a+bA+c*log(t)	*(1-tr)-s*d*(1 +d*t*2+e/t*2)	4r)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/3 0	2000.000
	210 230 250	ĸ	1.3	y=a*2/(1-b)+b-2*a*c y=exp(a+bA+c*log(1) y=a+b*(1-b)*0.35+c	^(1-tr)-s*d*(1 +d*1*2+e#*2) *(1-tr)*(2/3)+	-tr)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/5	2000/12/2005 []
Point 2	210 238	к к	1.3	y=a*2/(1-br)+b-2*a*c y=exp(a+bit+c*log(t) y=a+b*(1-br)*0.35+c &J(kmaPK)	*(1-tr)-=*d*(1 +d*1*2+e.t*2) *(1-tr)*(2/3)+ 0	4r)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/3 0	2000/12/2005 []
Point 2 Point 3	210 230 250	к к	1.3	yea/2/(1-tr)+b-2*a*c yeexp(a+b4+c*log() yea+b*(1-tr)*0.35+c &J/(kmol*K) kJ/(kmol*K)	*(1-tr)-s*d*(1 )+d*t*2+e.t*2) *(1-tr)*(2/3)+ 0 0	-tr)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/5 0 0	2000/12/2005 []
Point 2 Point 3 Point 4	210 230 250 265	K K K	1.3 1.4 1.45	y=a*2/(1-br)+b-2*a*c y=exp(a+bA+c*log(l) y=a+b*(1-br)*0.35+c kJ/(kmol*K) kJ/(kmol*K) kJ/(kmol*K)	*(1-tr)-s*d*(1 ++d*1*2+e#*2) *(1-tr)*(2/3)+ 0 0 0	-tr)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/5 0 0 0 0	2000/12/2005 []
Point 2 Point 3 Point 4 Point 5	210 230 250 265 270	K K K	1.3 1.4 1.45 1.5	y=a*2/(1-b)+b-2*a*c y=exp(a+bd+c*log() y=a+b*(1-b)*0.35+c kd(limol*K) kd(limol*K) kd(limol*K) kd(limol*K)	*(1-tr)-s*d*(1 )+d*t*2+et*2) *(1-tr)*(2/3)+ 0 0 0 0	-tr)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/5 0 0 0 0 0	2007.20080. 
Point 2 Point 3 Point 4 Point 5 Point 6	210 230 250 265 270 290	K K K K	1.3 1.4 1.45 1.5 1.55	y=a*2/(1-b)+b-2*a*c y=exp(a+bA+c*log() y=a+b*(1-b)*0.5+c kJ(limol*K) kJ(limol*K) kJ(limol*K) kJ(limol*K) kJ(limol*K)	*(1-tr)-s*d*(1 ++d*1*2+et*2) *(1-tr)*(2/3)+ 0 0 0 0	-tr)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/5 0 0 0 0 0 0 0	2007.22483. 
Point 2 Point 3 Point 4 Point 5 Point 6 Point 7	210 230 250 265 270 290 310	K K K K K	1.3 1.4 1.45 1.5 1.55 1.59	y=a*2/(1-b)+b-2*a*c y=exp(a+bA+c*log() y=a+b*(1-b)*0.5+c kJ(kmoPK) kJ(kmoPK) kJ(kmoPK) kJ(kmoPK) kJ(kmoPK) kJ(kmoPK)	^(1-tr)-=^d^(1 ++d^*t^2+et^2) (1-tr)^(2(3)+ 0 0 0 0 0 0 0 0 0	-tr)*2-c*2*(1-tr)*3/3 d*(1-tr)+e*(1-tr)*(4/5 0 0 0 0 0 0 0 0 0 0 0	2007.20080. 

## 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 <u>flexible datbase format</u> 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

<ul> <li>Operating</li> <li>Components</li> <li>Models</li> </ul>	WATER		ACETONE								
						-					
BIPs	Sort by first name					*					
Config	PRX-NTRL-HV					1.1					
Units	VLE BIPs					•					
Settings											
Chemicals											
Data Settings		200.41	(			1.41					
Regress	Min temp.in data set	329.42			К	14					
BIPs	Max temp in data set										
Data	Min press in data set		1434		bar a	-					
Regress	Max press in data set	30 47			bar a	-					
Models	X Y data fitting error %	2.043	S			_					
Licence	A12	-1310.	2.5								
	A21	-3547.									
	G12		-0.329054								
	K12	0									
						•					
	1					-					
		Save	File								
			1								

# 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)

E Stream				
Operating	METHANOL			
Components	WATER		M	
Models	Sort by first name			
EIPs E Config	PRX-NRTL(P-HV)			
	VLE BIPs			
BIPs Data Regress				
Models	Min temp.in data set	338.86	К	
Licence	Max temp.in data set	443.15	К	

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

Stream	Model for vapor pl	hana	DDY					
Operating				PRX-NRTL(P-HV)				
Components	Model for liquid pl			NRTL(P-HV)		_		
Models	Model for solid ph	lase	SPR	(-NRTL(P-HV)				
BIPS	Regress		meas	ured VI F-I I F-SI F data points		•		
Config	Bips data set to s	solve	Stan	dard set, more accurate but slow	(	+		
Chemicals	Minimization mod	le	Minir	Minimum set, quick but less accurate				
- Data			Stan	dard set, more accurate but slow				
- Data Regress Models		value			-			
- Data Regress Models	1112	1000	low limit	high limit	Calculate			
- Data Regress Models	U12	1000	-10000	high limit	Calculate			
- Data Regress Models	U21	1000 1000	-10000 -10000	high limit 10000 10000	-			
- Data Regress Models		1000 1000 0.33045	low limit -10000 -10000 0.001	high limit	Calculate			
and the second s	U21	1000 1000	-10000 -10000	high limit 10000 10000	Calculate			

• 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 in liquid phase, and in Y1 column the measured solid 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 VLE VLE VLE VLE VLE VLE	X(1) 0.008400 0.025800 0.068000 0.137000 0.240000 0.480000	Y(1) 0.103000 0.227000 0.391000 0.568000 0.680000 0.790000	Temperature (C) 96.5 92.3 87.5 80.1 75.9 70.6	Pressure (mmHg) 735 735 735 735 735 735 735
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

erating Model	for vapor	phase				PRX-NR	TL(P-HV	PRX-NRTL(P-HV)					
	Model for liquid phase					PRX-NR	PRX-NRTI (P-HV)						
0.51750	for solid	phase				SPRX-N	RTL(P-H	V)					
S Regres	ss						measured VLE-LLE-SLE data points						
Bips d	lata set to	o solve				Standard set, more accurate but slow							
Minimi	ization m	ode				F = xerr	* yerr						
a iress				value	lov	v limit	1	high limit	Calcula	te			
U12			858.72		-10000		10000	1					
U21			860.644		-10000 10000		C						
A12			0.455268		0.001 0.6								
			0		0 0		Save						
			0		0		0						
	Y1		Tempera	ature		Pressure		X calc.	Error %	Y calc.			
0.103		96.5	10	С	735	mn	hHG.a	0.00974637	-16.0283	0.0907218			
0.227		92.3		С	735	1111	iHG.a	0.0253501	1.74365	0.233345			
0.391		87.5		С	735	mn	hHG.a	0.0563708	17.1018	0.384345			
0.568		80.1		С	735	mn	nHG.a	0.122284	10.7418	0.564819			
0.68		75.9		С	735	mn	nHG.a	0.209683	12.632	0.659795			
0.79		70.6		С	735	mn	HG.a	0.426852	11.0725	0.769377			
0.82		68.7		C	735		nHG a	0.528006	7.69124	0.802724			
1 906		66 /		C	1735	mn	HC a	0 7/0181	0 11053	0 85/537			

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

Stream	Model for vapor pl	nase			PRX-NRT	L(P-HV	)		•		
- Operating Components							PRX-NRTL(P-HV)				
- Models							SPRX-NRTL(P-HV)				
BIPs	Regress				VLE points calculated with UNIFAC						
Config	Bips data set to s	olve		Standard set, more accurate but slow							
Chemicals	Minimization mod	le			F = xerr *	verr					
BIPs Data Regress											
Begress Begress Begress Begress Begress			value	low li	mit	1	high limit	Calcula	ate		
Licence	U12	-7.96	6507	-10000		10000	li				
	U21	U21 995.52		-10000	-10000 1000		6	Clea	r		
	A12	0.39	4105	0.001		0.6					
		0		0		0		Save	•		
		0		0		0					
	Temp	erature		Pressure	Xo	alc.	Error %	Y calc.	Error %		
	64.2473	С	760.02	mml IG.a	1.02038		-2.14	0.99954	0.00586564		
	65.0575	C	760.02	mmHG.a	0.96715	4	-2.18533	0.975589	0.295526		
	65.8841	С	760.02	mmHG.a	0.91345	3	-2.18274	0.952004	0.548013		
	66.7291	С	760.02	mmHG.a	0.859304	4	-2.12645	0.928728	0.761984		
	67.595	C	760.02	mmHG.a	0.804742	2	-2.01044	0.905691	0.936305		
	68.4851	C	/60.02	mmHG.a	0.74981	В	-1.82863	0.882797	1.0/014		
	69.4033	С	760.02	mmHG a	0.69459		-1.57521	0.85992	1.16304		
	70 3654	~	1 760 02	mmHC a	10 63915	1	-1 2//508	0.836895	1 21506		

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).

Stream							
Operating	METHANOL						
Components	WATER						
- Models BIPs	Sort by first name						
Config	PRX-NRTL(P-HV)						
Chemicals	VLE BIPs			-			
BIPs Data Regress							
Models	Min temp in data set	337.397		к 🖾			
Licence	Max temp.in data set	372.715		K			
	Min press.in data set	101327		Pa.a			
	Max press.in data set	101327		Pa.a			
	X-Y data fitting error %	2.38934					
	K12	0					
	U12	-7.96507					
	U21	995.528	995.528				
	U12-T	0	0				
	U21-T	0	0				
	A12	0.394105	0.394105				
		Save	File				

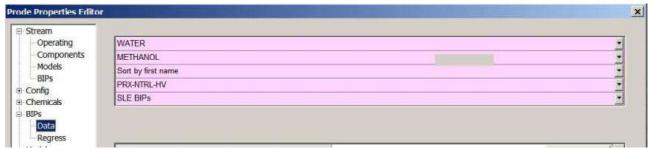
#### Regress SLE (Water-Methanol) and test the calculated freezing point depression

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

Stream	Model for vapo	r phase			PRX-NRT	L(P-HV	î.			•	Ī.	
- Operating - Components							PRX-NRTL(P-HV)					
Models							SPRX-NRTL(P-HV)					
BIPs	Regress				measured VLE-LLE-SLE data points							
Config	Bips data set	-					Standard set, more accurate but slow					
Chemicals BIPs	Minimization r	node			F = xerr	* yerr				•		
Regress Models Licence			value	1	limit		high limit	Calcul	ate			
	U12		317.196	-10000		10000						
	U21		-485.756	-10000				Clear				
	A12		0.6	100 Sec. 10		0.6						
		0		0 0		Save						
			0	0		0						
	¥1		Temperature		Pressure		X calc.	Error %		Y calc. 🔺	]	
	1	266.85	ĸ	0	ba	ir.g	0.94654	-0.354061	1			
	1	259.65	ĸ	0	ba	r.g	0.89154	0.93996	1			
	1	257.65	K	0	ba	ır.g	0.88209	-0.607897	1			
	1	246.85	ĸ	0	ba	r.g	0.817349	-1.4294	1			
	1	242.95	K	0	ba	ır.g	0.784832	1.89598	1			
	1	233.45	K	0	ba	ir.g	0.742629	-2.00941	1			
	1	217.95	K	0	ba	ır g	0.662321	-3.32619	1		1	
	1	21/ 95	L.	10	ha	ur a	10 639367	-0 527807	1		1	

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) 58

Prode Properties Edito	or and a second s							
⊡-Stream								
- Operating	WATER					-		
Components	METHANOL							
Models	Sort by first name							
BIPS	PRX-NRTL(P-HV)					•		
Config     Chemicals	SLE BIPs					-		
BIPs Data Regress								
⊕ Models	Min temp in data set		208.15		К			
± Licence	Max temp.in data set		266.85	K				
	Min press.in data set		101327		Pa.a			
	Max press in data set		101327		Pa.a			
	X-Y data fitting error %		0.656665		2	-		
	K12		0					
	U12		317.196					
	U21		-485.756					
	U12-T		0			-		
	U21-T		0					
	A12		0.6					
						<b>_</b>		
		Save		File				
				ОК	Cancel	Apply		

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

Stream					
Operating	ACETALDEHYDE			-	
Components	Sort by first name				
Models BIPs Config Chemicals	Add	Remove	e C	ear	
BIPs	Units	Molar flov	w	-	
Data	Flow (stream)	0.008263	kmol/s		
Regress Models		20 			
Licence	Component		Molar fraction	i l	
	METHANE		0		
	ETHANE		0		
	WATER		0.636		
			0.364		

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

Operating	Predefined packages		5 PRX-NRTL-HV (incl. hydrate)	PRX-N	IRTL-HV (incl. hydrate)	Save	
Components Models BIPs			1 SRK standard 2 SRKX (SRK Extended) 3 PR Standard				
Config			4 PRX (PR Extended) 5 PRX-NRTL-HV (incl. hydrate)	uid	Solid	Hydrate	
Chemicals	Fugacity	PRX-N	6 SRKX-NRTL-HV		SP-PR-NRIL-HV	HYD-PR-NRIL-HV	-
BIPs	Enthalpy	PRX-N	7 CPA-PRX (incl.hydrate) 8 CPA-SRK		REGULAR	HYD-PR-NRTL-HV	-
Data	Entropy	PRX-N	9 PRX-GMR		REGULAR	HYD-PR-NRTL-HV	¥
Models	Volume	PRX \	10 Lee Kesler Plocker 11 Benedict-Webb-Rubin-Starlin(		REGULAR	HYD-PR-NRTL-HV	<b>_</b>
Licence	Multiphase equilibria		12 PRX-WILSON WS 13 PRX-NRTL WS 14 PRX-UNIQUAC WS 15 PRX-WILSON MHV2 16 PRX-NRTL MHV2 17 PRX-UNIQUAC MHV2	Multip	vhase vapor-liquid		Ŧ

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

Stream									
Operating	Edit BIPs				Use edited BIPS SLE BIPs				•
- Components	Select BIPs Data Set			-					
Models	Get BIPs				Get BIPs from database				
Config	Select the model				PRX-NRTL(P-HV)				-
Chemicals									
BIPs Data	C1	C2	K12	U12	U21	U12-T	U21-T	A12	
BIPs	C1	C2	<mark>К12</mark>	U12 317.196	<b>U21</b> -485.756	U12 T	U21 T	A12	

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

Operating	Selected Stream		9					Save			
Components	Lociccica oucam			<u>.</u>	4			ave			
Models	Operation to solv	8	T-P Flash			•	Compute				
BIPs	Feed(s)		9					*			
hemicals	Spec. (IN)		214.5		к	101327		Pa.a			
IPs	Spec. (OUT)				Pa.a	1		kW			
Data											
Regress	Stream Operating	1	214.5		к	101327		Pa.a			
iodels icence	Flow units Flows (mole)										
	Phase	Feed	Liquid	Solid	Not present	Not present	Not present	Not present			
	Flow ( kmol/s )	0.0082635	0.00825582	7.67843e-06	0	0	0	0			
	Fraction (molar)	1	0.999071	0.000929199	0	0	0	0			
	CH4	0	0	0	0	0	0	0			
	C2H6	0	0	0	0	0	0	0			
	CONTRACTOR CONTRACTOR										
	H2O	0.636	0.635661	1	0	0	0	0			

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)

Stream Operating	Selected Stream		9	1	Save						
Components											
Models	Operation to solve	e	T-P Flash			•					
BIPs Config	Feed(s)		9		•	1 Test Case 1					
hemicals	Spec. (IN)		214.5		к	10		bar.a			
IPs	Spec. (OUT)				Pa.a	1		kW			
- Data											
Regress	Stream Operating	1	214.5		к	1e+06		Pa.a			
odels cence	Flow units Flows (mole)										
	Phase	Feed	Vapor	Liquid	Solid	Not present	Not present	Not present			
				0.07004-05	2.11599e-08	0	0	0			
	Flow ( kmol/s )	0.0082635	0.00818076	8.27224e-05	2.110000-00						
	Flow ( kmol/s ) Fraction (molar)	0.0082635	0.00818076	0.0100106	2.56065e-06	0	0	0			
		0.0082635 1 0.87	Charles and a state			0	0	0			
	Fraction (molar)	1	0.989987	0.0100106	2.560659-06			0 0 0			
	Fraction (molar) CH4	1 0.87	0.989987 0.878778	0.0100106	2.56065e-06 1.3666e-11	0	0	0			

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.

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

	<u>×</u>
WATER Sort by first name CPA-PR UNIQUAC UNIFAC (REV.5) SRK-EP VDW PR-EP VDW	× • •
PR-NTRL-WS PR-WILSON-WS PR-UNIQUAC-WS PC-SAFT CPA-SRK CPA-PR E Associaton Energy	2 166.55
Associaton Volume	0.0692
Save	File OK Cancel Apply
	Sort by first name CPA-PR UNIQUAC UNIFAC (REV.5) SRK-EP VDW PR-EP VDW PR-NTRL-WS PR-WLSON-WS PR-UNIQUAC-WS PC-SAFT CPA-SRK CPA-PR E Associaton Energy Associaton Volume

## 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)

# <u>C / C++</u>

- 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

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)	CONVT	17	"K"
Temperature(rel)	CONVDT	18	"K"
Calorific Value (weight)	CONVHM	19	"Kj/Kg″
Calorific Value (molar)	CONVHMM	20	"Kj/Kmol"
Power	CONV HS	21	"KW"
Entropy (Streams)	CONV SS	22	"KJ/(K*s)″
Heat Capacity (weight)	CONVCP	23	"kJ/(kg*K)"
Heat Capacity (molar)	CONV CPM	24	"kJ/(kmol*K)"
Flow (mass basis)	CONVW	25	"Kg/s"
Flow (gas, mass basis)	CONV WG	26	"Kg/s"
Density (weight)	CONVD	27	"Kg/m3"
Density (molar)	CONV DM	28	"Kmol/m3"
Specific Volume (weight)	CONVSV	29	"m3/Kq"
Specific Volume (molar)	CONVSVM	30	"m3/Kmol"
Thermal Conductivity	CONVTC	31	₩V/(m*K)″
Viscosity (dynamic)	CONVV	32	"Pa*s"
Surface Tension	CONVST	33	"N/m"
Lenght	CONVL	34	"m"
Area	CONVA	35	"m2"
Volume	CONV VOL	36	"m3"
Mass	CONVM	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

# 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 If = 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 composition 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

## <u>Note : this method utilizes the standard ANSI C convention for exchanging parameters</u> (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				
prod_h	( )	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			
reboile		1		

2 3
1
2
3
4
5
6
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] stgdH[0] 	= 1 = 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 vtype[0] ptype[0] prv[0] vtype[1] ptype[1] piv[1] prv[1]	= 2 = 1 = 4 = 2 = 0.96 = 2 = 3 = 0 = 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 sstr[0] spos[0] sft[0] sflow[0] sstr[1] sft[1] sflow[1]	= 2 = stream4 = 4 = LIQ_PHASE = 0.12 = stream5 = 7 = GAS_PHASE = 0.078

# <u>Note : this method utilizes the standard ANSI C convention for exchanging parameters</u> (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
<b>•</b> • •	1.1.	

1 2

#### **Codes for models** Gibbs Equilibrium Reactor

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

1

#### Codes for models

Beggs & Brill / Hazen-Williams / AGA 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) t (double) method (int)

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

## Methods for solving a Polytropic operation

Polytropic stage (compression and expansion)

#### 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
4h	
the procedure can model the procedure returns	compression and expansion units such as centrifugal compressors, expansion turbines etc.
-calculated temperature	options 1,3
-calculated efficiency	options 2,4
models available (**)	
1	given initial condition, pout and polytropic efficiency calculates outlet condition,
	R.A. Huntington "Evaluation of Polytropic calculation Methods for Turbomachinery Performance", method applicable to gas phase only
2	given initial condition, pout and tout calculates polytropic efficiency,
L	R.A. Huntington "Evaluation of Polytropic calculation Methods for Turbomachinery Performance",
	method applicable to gas phase only
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 streampout (double)outlet pressuremodel (int)model, see below codes 1-4param(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 2 3 4	<ul> <li>HEM Homogeneous Equilibrium (Solution of Mass Flux integral)</li> <li>HNE Homogeneous Non-equilibrium (HEM with Boling Delay and Gas-Liquid Slip Contributes)</li> <li>HNE-DS, Homogeneous Non-equilibrium</li> <li>NHNE Non-homogeneous Non-equilibrium</li> </ul>	
model HEM HNE HNE-DS NHNE	recommended range of values for correction parameter not required 0.7-0.8 for safety valves see the paper 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, vapor-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 Inr = PELnr(integer stream)

Given a stream calculates the phase diagram and returns the number of equilibrium lines available

### integer Inr = 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 Inr = 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 If = 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 several methods : getKO(), putKO() allow to access or define all the settings as a 32 bit integer, each bit in the integer represents a different option,

getKS(), putKS() allow to access or define each option, see below a short list of options available

#### int value = getKO(integer stream)

given a stream returns a code (integer) with the options

#### integer res = putKO (integer stream, integer value)

given a stream define the options

#### int value = getKS(integer stream, integer option)

given a stream and option (see below the codes) returns a boolean (0-1) with stored value

#### integer res = putKS (integer stream, integer option, integer value)

given a stream and option define the option.

### Codes used in Prode library

#### Table of codes to specify the different options

reference : methods getKO(), setKO() ...

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

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 Regular Regular 1 10 Wilson Wilson 11 NRTL NRTL UNIQUAC UNIQUAC 12 Soave-Redlich\_Kwong Std. (VDW) 30 SRK(VDW) Soave-Redlich Kwong Ext. (VDW) 31 SRKX(VDW) SRKX-NRTL(P-HV) Soave-Redlich\_Kwong Ext. + NRTL (Modified Huron Vidal) 40 41 Soave-Redlich\_Kwong Ext. + NRTL (Modified LCVM)
Peng Robinson Std. (VDW) SRKX-NRTL(P-LCVM) 50 PR(VDW) Peng Robinson Ext. (VDW) PRX(VDW) 51 Peng Robinson Ext. + Wilson (Wong Sandler) PRX-Wilson(WS) 55 Peng Robinson Ext. + UNIQUAC (Wong Sandler) PRX-UNIQUAC(WS) 56 Peng Robinson Ext. + NRTL (Wong Sandler) Peng Robinson Ext. + NRTL (Modified Huron Vidal) PRX-NRTL(WS) 57 PRX-NRTL(P-HV) 60 61 Peng Robinson Ext. + Wilson (Modified Huron Vidal) PRX-Wilson(P-HV) Peng Robinson Ext. + UNIQUAC (Modified Huron Vidal) PRX-UNIQUAC(P-HV) 62 PRX-Wilson(MHV2) Peng Robinson Ext. + Wilson (MHV2) 65 Peng Robinson Ext. + UNIQUAC (MHV2) Peng Robinson Ext. + NRTL (MHV2) PRX-UNIQUAC(MHV2) 66 PRX-NRTL(MHV2) 67 Peng Robinson Ext. + NRTL (Modified LCVM) PRX-NRTL(P-LCVM) 70 Peng Robinson Ext. + Wilson (Modified LCVM) 71 PRX-Wilson(P-LCVM) Peng Robinson Ext. + UNIQUAC (Modified LCVM) Peng Robinson Ext. + UNIFAC (Modified LCVM) 72 PRX-UNIQUAC(P-LCVM) 73 PRX-UNIFAC(WS) 80 Benedict-Webb-Rubin (modified) BWR 81 Benedict-Webb-Rubin-Starling BWRS 90 Lee Kesler IK 91 Lee Kesler Ploecker LKP PSAFT 100 P-SAFT 110 Soave-Redlich Kwong Ext. - CPA SRKX-CPA(VDW) PRX-CPA(VDW) Peng Robinson Ext. - CPA 111 Peng Robinson Ext. - CPA + NRTL (Modified Huron Vidal) Peng Robinson Ext. - CPA + NRTL (Modified LCVM) Peng Robinson Ext. - CPA + NRTL (Modified MHV2) Peng Robinson Ext. - CPA + NRTL (Modified Wong Sandler) PRXCPA-NRTL(P-HV) 115 PRXCPA-NRTL(P-LCVM) 116 PRXCPA-NRTL(P-MHV2) 117 PRXCPA-NRTL(P-WS) 118 Soave-Redlich\_Kwong Ext. - CPA + NRTL (Modified Huron Vidal) SRKX-CPA-NRTL(P-HV) 120 130 UNIFAC UNIFAC 150 Solid Pure (derived from) PRX-NRTL(P-HV) SPRX-NRTL(P-HV) Solid Pure (derived from) PRXCPA-NRTL(P-HV) 151 SPRXCPA-NRTL(P-HV) 153 Solid Solution (derived from) PRX-NRTL(P-HV) SSPRX-NRTL(P-HV) Hydrate (derived from) PRXCPA-NRTL(P-HV) Hydrate (derived from) PRX-NRTL(P-HV) HPRXCPA-NRTL(P-HV) 170 171 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) PSAFT-E 210 P-SAFT-(MSA) Electrolyte Steam tables based on IAPWS 1995 formulation IAPWS 95 300 311 **GERG 2008** GERG 2008 ISO 18453 (GERG) 312 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 state, integer model)

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()
<ul> <li>operating temperature,</li> </ul>	getT()	setOp()
• flow,	getW()	setW()
<ul> <li>vector [getMCNr() elements] with components codes</li> </ul>	getCC()	putCC()
<ul> <li>vector [getMCNr() elements] Z vector, mole basis</li> </ul>	getZ()	putZ()
<ul> <li>thermodynamic model (and related options)</li> </ul>	getMP()	setMP()
<ul> <li>vector[getMBPNr() elements] of Ci</li> </ul>	getCi()	putCi()
<ul> <li>vector[getMBPNr() elements] of Cj</li> </ul>	getCj()	putCj()
<ul> <li>vector[getMBPNr() elements] of BIP matrix</li> </ul>	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, ccl)
Call putZ(Stream, 2, z2)
Call putCC(Stream, 2, cc2)
Call setS(Stream)
                              ' validates stream
                              ' flow
Call setW(Stream, W)
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...

#### 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	00	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	ОН	OH	n-Propanol
15	СНЗОН	СНЗОН	Methanol
16	H2O	H2O	Water
17	ACOH	ACOH	Phenol
18	CH2CO	CH3CO	Butanone
19	011200	CH2CO	Pentanone-3
20	СНО	CHO	Propionic aldehyde
21	0000	CH3COO	Butyl acetate
22	0000	CH2COO	Methyl propionate
23	HCOO	HCOO	Ethyl formate
24	CH2O	CH3O	Dimethyl ether
25	01120	CH2O	Diethyl ether
26		CHO	Diisopropyl ether
27		THF	Tetrahydrofuran
28	CNH2	CH3NH2	Methylamine
29	02	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	СООН	COOH	Acetic acid
43		НСООН	Formic acid
44	CCI	CH2CI	Butane-1-chloro
45		CHCI	Propane-2-chloro
46		CCI	2-Methylpropane-2-chloro
47	CCI2	CH2Cl2	Methane-dichloro
48		CHCl2	Ethane-1,1-dichloro
49		CCI2	Propane-2,2-dichloro
50	CCI3	CHCI3	Chloroform
51		CCI3	Ethane-1,1,1-trichloro
52	CCI4	CCl4	Methane-tetrachloro
53	ACCI	ACCI	Benzene-chloro
54	CNO2	CH3NO2	NitroMethane
55	-	CH2NO2	Propane-1-nitro
56		CHNO2	Propane-2-nitro
- •		<b>v</b> =	

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	1	1	lodoethane
64	Br	Br	Bromoethane
65	C-C	CH-C	Hexyne-1
66		C-C	Hexyne-2
67	DMSO	DMSO	Dimethylsulfoxide
68	ACRY	AcryInitril	AcryInitrile
69	CICC	CI-(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	000	CF	Perfluoromethylcyclohexane
77	COO	COO	Methyl acrylate
78	SiH2	SiH3	Methylsilane
79 80		SiH2 SiH	Diethylsilane Heptamethyltrisiloxane
80 81		Si	Heptamethyldisiloxane
82	SiO	SiH2O	1,3-Dimethyldisiloxane
83	510	SiHO	1,1,3,3-Tetramethyldisiloxane
84		SiO	Octamethylcyclotetrasiloxane
85	NMP	NMP	N-methylpyrrolidone
86	CCIF	CCI3F	Trichlorofluoromethane
87	0011	CCI2F	Tetrachloro-1,2-difluoroethane
88		HCCI2F	Dichlorofluoromethane
89		HCCIF	1-Chloro-1,2,2,2,-tetrafluoroethane
90		CCIF2	1,2-Dichlorotetrafluoroethane
91		HCCIF2	Chlorodifluoromethane
92		CCIF3	Chlorotrifluoromethane
93		CCI2F2	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	Maria	CHS	Diisopropylsulfide
105	Morpholine	MORPH	Morpholine
106	Thiophene	C4H4S Thiophe	
107		C4H3S C4H2S	2-Methylthiophene 2,3-Dimethylthiophene
108		04023	