ProcessModelling

User Manual

Version 1.0.0

ADG Energy Services Ltd

www.adg-esl.co.uk

1. About This Tool

ProcessModelling grew out of work on gas pipeline criticality and thermodynamic stability — the question of whether a given gas composition, at given conditions, sits safely clear of the pressure/temperature region where it would separate into two phases inside a pipeline it was never designed to carry liquid in. That original, narrow focus grew over time into a general-purpose steady-state process simulation tool: equations of state, a full set of common unit operations, multiphase pipe flow, and the iterative machinery needed to solve flowsheets containing recycle loops and control targets.

The engineering core of the tool — the thermodynamics, the unit operation models, the convergence logic — was developed and proven first, then rebuilt as a native application for macOS, iPadOS, and iPhone, so it could be carried and used directly in the field rather than only at a desk.

A deliberate design choice

ProcessModelling is deliberately not a drag-and-drop flowsheet editor. A model here is written as a deck — a plain-text description of every stream and piece of equipment, in the language described in Section 3. That is a considered choice, not a limitation waiting to be fixed.

It is easy, in a purely graphical simulator, to stitch blocks together by eye until a flowsheet converges without ever having to state — in so many words — what each stream's conditions actually are, or why a given piece of equipment belongs where it does. Writing FEED=, PRESSURE=, TEMPERATURE= explicitly, in order, forces that thinking to happen. The keyword assistance built into the Model tab (Section 2) is deliberately limited to the same idea: it offers the shape of a valid block — which keywords it takes, and what kind of value belongs in each — and never guesses or fills in an actual value on the user's behalf. The tool is meant to help an engineer think through their own process, not to think through it for them.

2. Getting Started

The app is organised into a small number of tabs:

- Model — the deck itself, as plain text.
- Diagram — an automatically generated block diagram of the current deck, for a visual check of the flowsheet's shape.
- Flowsheet / Equipment — results tables, populated after running a calculation.
- Help / About — in-app reference and version information.

To begin, either open an existing .mod file, or use the “Load Example” button shown when the Model tab is empty — this loads a simple, deliberately uncluttered example deck (a feed stream, a heater, and a three-phase separator) as a starting point to explore the deck language from.

With a deck loaded, “Run Calculation” solves the model and populates the Flowsheet and Equipment tabs. A convergence warning banner appears if any part of the model didn't settle cleanly — worth checking before trusting the results.

Inserting a block

Above the Model tab's text editor is an “Insert Block” menu, listing every block type this deck language has — STREAM, VALVE, HEATER, and so on through REPORT. Choosing one appends that block's full skeleton to the end of the deck: its header line, followed by every keyword it takes, each with a placeholder showing what kind of value is expected (e.g. FEED=<stream name>, PRESSURE=<value>) rather than an actual guessed value. This is a reference aid, not an autocomplete in the conventional sense — see “A deliberate design choice” above for why it stops there.

A few block types have keywords that are genuine alternatives rather than all used together — for example HEATER's TEMPERATURE= versus DUTY=, or PUMP/COMPRESSOR/EXPANDER's flat-efficiency mode

versus a full performance curve. Where this applies, the inserted skeleton includes a plain comment line marking the choice, rather than leaving it to guesswork.

A three-phase feed (containing water) needs PHASES=3 on its STREAM block — forcing PHASES=2 on a wet feed will produce persistent phase-balance errors.

3. The Deck Language

A deck is read from top to bottom. TITLE, COMPONENTS, THERMO, and UNITS (in that order) normally come first and apply to the whole model; a stream referenced by name (in a FEED=, FEEDS=, or similar keyword) must already have been defined earlier in the deck, either as a STREAM block or as an unnamed default outlet from an earlier piece of equipment. REPORT blocks are placed wherever their output is wanted, usually at the end.

Every block below is introduced by its own line (the block type, then a name — e.g. “HEATER H101”), followed by its own indented keyword lines.

TITLE

A single free-text line naming the model. Optional, but a good habit for keeping multiple decks straight.

COMPONENTS

Lists every chemical component the model will use, by name (e.g. N2, CO2, C1, C2, C3, nC4, H2O). Must appear before the first STREAM block. Component names must match entries in the critical properties data file.

THERMO

Chooses the equation of state and the property data files to use for the whole model.

EOS=<PR|SRK>	Which equation of state to use — Peng-Robinson or Soave-Redlich-Kwong. See the Documentary Reference section for the original papers behind each.
CRITCSV=<filename.csv>	Critical-properties data file to use instead of the bundled default. A bare filename is looked for alongside the deck file itself; leave this out entirely to use the bundled default.
KIJCSV=<filename.csv>	Binary interaction parameter file, same rules as CRITCSV= above.

UNITS

Sets which unit system the deck's own numbers are written in, and which unit system results are reported in. These can differ — e.g. type the deck in FIELD units but see results in SI.

INPUT=<SI|FIELD|METRIC>	Unit system the values you type into this deck are in.
OUTPUT=<SI|FIELD|METRIC>	Unit system results are reported in.

UNITSET

Defines a custom, named unit system (an alternative to the built-in SI/FIELD/METRIC presets) for use in a later UNITS block. Give it a name, then set only the fields you want to differ from SI.

PRESSURE=<unit>	e.g. psia, bara, kPa
TEMPERATURE=<unit>	e.g. F, C, K
MOLARFLOW=<unit>	e.g. lbmol/h, kmol/h
MASSFLOW=<unit>	e.g. lb/h, kg/h
VOLUMEFLOW=<unit>	e.g. bbl/d, m3/h
GASSTDFLOW=<unit>	e.g. MMSCFD, Sm3/d
DENSITY=<unit>	e.g. lb/ft3, kg/m3
MOLARDENSITY=<unit>	e.g. lbmol/ft3, kmol/m3
VISCOSITY=<unit>	e.g. cP, Pa.s

THERMALCONDUCTIVITY=<unit>	e.g. Btu/(h.ft.F), W/(m.K)

STREAM

Defines a feed stream — a starting point for the model. Give it a name, a composition (each component name followed by its amount), and its conditions.

PRESSURE=<value>	Stream pressure.
TEMPERATURE=<value>	Stream temperature.
FLOW=<value>	Stream flow rate, interpreted per BASIS= below.
BASIS=<GASSTD|MOLAR|MASS>	What kind of quantity FLOW= represents — a standard gas volume rate, a molar rate, or a mass rate.
PHASES=<2|3>	Whether to allow gas/liquid (2) or gas/liquid/aqueous (3) phase splitting. Use 3 for any feed containing water — see the Getting Started section.
TVP=<YES|NO>	Whether to also compute the stream's true vapor pressure. Off by default — inexpensive to turn on.
RVP=<YES|NO>	Whether to also compute the stream's Reid vapor pressure. Off by default — noticeably more expensive than TVP=, since it needs its own separate calculation.

VALVE

A pressure-reducing device with no separate energy input — an isenthalpic pressure drop.

FEED=<stream name>	The stream entering the valve.
PRODUCT=<stream name>	Name to give the outlet stream.
PRESSURE=<value>	Outlet pressure.
PUNIT=<unit>	Overrides the deck's own pressure unit just for this PRESSURE= value.
TVP=<YES|NO>	See STREAM's own TVP= above — applies the same way to this block's outlet.
RVP=<YES|NO>	See STREAM's own RVP= above.

HEATER

Adds or removes heat from a stream, to either a target outlet temperature or a specified duty.

FEED=<stream name>	The stream entering the heater.
PRODUCT=<stream name>	Name to give the outlet stream.
TEMPERATURE=<value>	Target outlet temperature. Give this OR DUTY= below, not both.
TUNIT=<unit>	Overrides the deck's own temperature unit just for this TEMPERATURE= value.
DUTY=<value>	Target heat duty instead of a target temperature, always in plain watts regardless of the deck's own units.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

PUMP / COMPRESSOR / EXPANDER

Raises (PUMP/COMPRESSOR) or lowers (EXPANDER) a stream's pressure with a real energy input or output, at a given efficiency. Two ways to define one: a target pressure and a flat efficiency, or a full performance curve.

FEED=<stream name>	The stream entering the machine.
PRODUCT=<stream name>	Name to give the outlet stream.
PRESSURE=<value>	Target outlet pressure (flat-efficiency mode).
PUNIT=<unit>	Overrides the deck's own pressure unit just for this PRESSURE= value.
EFFICIENCY=<0-1>	Flat efficiency to apply (flat-efficiency mode).
EFFTYPE=<ADIABATIC|POLYTROPIC>	Which efficiency convention EFFICIENCY= (or an EFFCURVE=) is defined against.
NPOLYSTEPS=<integer>	Number of calculation steps used for a polytropic solution. Defaults to 5.
FLOWCURVE=<v1,v2,...>	Flow points defining a performance curve (curve mode) — give this together with DPCURVE= and exactly one of EFFCURVE= or POWERCURVE=, instead of PRESSURE=/EFFICIENCY=.
FLOWCURVEUNIT=<unit>	Unit for the FLOWCURVE= values.
DPCURVE=<v1,v2,...>	Pressure-rise points matching each FLOWCURVE= point.
DPCURVEUNIT=<unit>	Unit for the DPCURVE= values.
EFFCURVE=<v1,v2,...>	Efficiency at each FLOWCURVE= point (curve mode, efficiency-driven).
POWERCURVE=<v1,v2,...>	Power at each FLOWCURVE= point instead of EFFCURVE= (curve mode, power-driven). Give exactly one of the two.
POWERCURVEUNIT=<unit>	Unit for the POWERCURVE= values.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

MIXER

Combines two or more streams into one, conserving mass and energy.

FEEDS=<stream1,stream2,...>	The streams to combine.
PRODUCT=<stream name>	Name to give the combined outlet stream.
PRESSURE=<value>	Optional outlet pressure override — otherwise the mixer resolves pressure automatically from its feeds.
PUNIT=<unit>	Overrides the deck's own pressure unit just for this PRESSURE= value.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

SPLITTER

Divides one stream into two or more, each carrying the same composition and conditions but a different share of the flow.

FEED=<stream name>	The stream to split.
PRODUCTS=<name1,name2,...>	Names to give each outlet stream.
FRACTIONS=<f1,f2,...>	Fraction of the feed flow going to each product, in the same order as PRODUCTS= — must sum to 1.
PDROPS=<dp1,dp2,...>	Optional pressure drop for each individual product line, same order as PRODUCTS=.
PUNIT=<unit>	Overrides the deck's own pressure unit just for the PDROPS= values.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

PIPE

Models pressure drop (and, for a wet or multiphase feed, phase behaviour) along a length of pipe, given its size, route, and roughness.

FEED=<stream name>	The stream entering the pipe.
PRODUCT=<stream name>	Name to give the outlet stream.
LENGTH=<value>	Pipe length.
LUNIT=<unit>	Overrides the deck's own length unit just for LENGTH= (and ELEVATION=, if used).
DIAMETER=<value>	Internal diameter, if not describing the pipe by nominal size and schedule.
DUNIT=<unit>	Overrides the deck's own length unit just for DIAMETER=.
NOMINALSIZE=<value>	Nominal pipe size (inches), used together with SCHEDULE= instead of DIAMETER=.
SCHEDULE=<schedule>	Pipe schedule (e.g. 40, 80), used together with NOMINALSIZE=.
ELEVATION=<value>	Net elevation change over the pipe's length. Give this OR ANGLE= below, not both.
EUNIT=<unit>	Overrides the deck's own length unit just for ELEVATION=.
ANGLE=<degrees>	Uphill (positive) or downhill (negative) angle from horizontal, instead of ELEVATION=.
ROUGHNESS=<value>	Internal pipe roughness. Has a sensible default if left out.
RUNIT=<unit>	Overrides the deck's own length unit just for ROUGHNESS=.
FITTINGS=<...>	Equivalent-length fittings to include in the pressure-drop calculation.
NSEGMENTS=<integer>	Number of segments to break the pipe into for a multi-step calculation, re-checking phase behaviour partway along the run rather than once at the inlet. Defaults to 1 (a single, lumped calculation).
METHOD=<BASIC|BEGGSBRILL>	Calculation method — a simple single-phase pressure-drop method, or the Beggs & Brill multiphase correlation (see Documentary Reference).
PRESSURE=<value>	Optional fixed outlet pressure, if not letting the pipe calculate its own pressure drop.
PUNIT=<unit>	Overrides the deck's own pressure unit just for this PRESSURE= value.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

SCRUBBER

A simple gas/liquid separator — two outlets, no water phase modelled separately from the liquid.

FEED=<stream name>	The stream entering the scrubber.
GASPRODUCT=<stream name>	Name to give the gas outlet stream.
LIQUIDPRODUCT=<stream name>	Name to give the liquid outlet stream.
CARRYOVER=<0-1>	Fraction of liquid carried over into the gas outlet, modelling imperfect separation.
PDROP=<value>	Overall pressure drop across the vessel, applied to both outlets unless overridden below.
PDROP_GAS=<value>	Pressure drop for the gas outlet specifically, overriding PDROP= for that outlet only.
PDROP_LIQUID=<value>	Pressure drop for the liquid outlet specifically.
PUNIT=<unit>	Overrides the deck's own pressure unit for every PDROP keyword above.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

SEPARATOR

A full three-phase separator — gas, oil, and water outlets, each with its own configurable carryover into the others.

FEED=<stream name>	The stream entering the separator.
GASPRODUCT=<stream name>	Name to give the gas outlet stream.
OILPRODUCT=<stream name>	Name to give the oil outlet stream.
WATERPRODUCT=<stream name>	Name to give the water outlet stream.
CARRYUNDER_OIL_WATER=<0-1>	Fraction of oil carried under into the water outlet.
CARRYOVER_WATER_OIL=<0-1>	Fraction of water carried over into the oil outlet.
CARRYOVER_OIL_GAS=<0-1>	Fraction of oil carried over into the gas outlet.
CARRYOVER_WATER_GAS=<0-1>	Fraction of water carried over into the gas outlet.
PDROP=<value>	Overall pressure drop, applied to all three outlets unless overridden below.
PDROP_GAS=<value>	Pressure drop for the gas outlet specifically.
PDROP_OIL=<value>	Pressure drop for the oil outlet specifically.
PDROP_WATER=<value>	Pressure drop for the water outlet specifically.
PUNIT=<unit>	Overrides the deck's own pressure unit for every PDROP keyword above.
TVP=<YES|NO>	See STREAM's own TVP= above.
RVP=<YES|NO>	See STREAM's own RVP= above.

COMPONENTSPLIT

Removes a fraction of one component from a stream by simple mass balance — stands in for equipment such as a dehydration unit without modelling its internal physics.

FEED=<stream name>	The stream to remove a component from.
REMOVE=<component:fraction>	Which component to remove, and what fraction of it — e.g. H2O:0.95 removes 95% of the water present.
PRODUCT=<stream name>	Name to give the main outlet stream (what's left after removal).
REMOVEDPRODUCT=<stream name>	Name to give the stream carrying whatever was removed.
PDROP=<value>	Optional pressure drop across the unit.
PUNIT=<unit>	Overrides the deck's own pressure unit just for PDROP=.

RECYCLE

Closes a loop in the flowsheet — a downstream stream feeds back into an earlier point in the model, and the model iterates until the guessed and calculated values agree. Needs an initial guess to start from.

FEED=<stream name>	The real, calculated stream at the far end of the loop — what the guess is being checked against each pass.
PRODUCT=<stream name>	Name to give the current guessed stream, used elsewhere in the deck as the actual feed to whatever comes next in the loop.
INITIALGUESS=<stream name>	An existing stream to seed the very first guess from — the closer this is to the eventual answer, the fewer iterations are needed.
MAXITER=<integer>	Maximum number of iterations to attempt before giving up.
DAMPING=<0-1>	Slows how aggressively each new guess moves toward the previous pass's answer. Lower values (try 0.5 first) make a stubborn loop — particularly one with a cooler and scrubber inside it — more likely to settle rather than oscillate.

ADJUST

A control block, not a physical piece of equipment — automatically varies one keyword on another block until a target elsewhere in the model is met. A pure control block has nothing of its own to show in a REPORT.

ADJUSTEDVAR=<BLOCK.KEYWORD>	Which block and keyword to vary — e.g. K101.PRESSURE.
TARGETVAR=<BLOCK.KEYWORD>	Which block and keyword to watch as the target — e.g. K101.POWER, or a downstream stream's own property.
TARGETVALUE=<value>	The value TARGETVAR= should reach. Note that POWER/DUTY targets are always plain watts, regardless of the deck's own units.
MIN=<value>	Lower bound for ADJUSTEDVAR= — must genuinely bracket the answer, since the first two passes probe MIN and MAX directly before any further step is trusted.
MAX=<value>	Upper bound for ADJUSTEDVAR=, same bracketing requirement as MIN= above.
TOLERANCE=<value>	How close TARGETVAR= needs to get to TARGETVALUE= to count as converged.
MAXITER=<integer>	Maximum number of iterations to attempt before giving up.

REPORT

Prints and/or writes to CSV the results for named streams and equipment. A bare name can be either a stream or a piece of equipment. ADJUST blocks are never included — they have nothing of their own to show.

FLOWSHEET=<YES|name1,name2,...>	YES prints one combined table covering every stream in the deck; a specific list restricts it to just those streams.
FLOWSHEET_CSV=<filename.csv>	Also writes the flowsheet table to this file, alongside the deck file itself.
EQUIPMENT=<YES|name1,name2,...>	Same as FLOWSHEET= above, but for equipment rather than streams.
EQUIPMENT_CSV=<filename.csv>	Also writes the equipment table to this file, alongside the deck file itself.

4. Documentary Reference

ProcessModelling's core methods are drawn from established, published chemical and petroleum engineering literature rather than novel or proprietary techniques. The primary sources for each are given below for anyone wanting to check the underlying theory directly.

Peng, D.-Y.; Robinson, D.B.
“A New Two-Constant Equation of State.” Industrial & Engineering Chemistry Fundamentals, 15(1), 59–64, 1976.
Used for: The Peng-Robinson equation of state (EOS=PR).

Soave, G.
“Equilibrium Constants from a Modified Redlich-Kwong Equation of State.” Chemical Engineering Science, 27(6), 1197–1203, 1972.
Used for: The Soave-Redlich-Kwong equation of state (EOS=SRK).

Beggs, H.D.; Brill, J.P.
“A Study of Two-Phase Flow in Inclined Pipes.” Journal of Petroleum Technology, 25(5), 607–617, 1973.
Used for: The multiphase pipe flow correlation (PIPE METHOD=BEGGSBRILL).

Wegstein, J.H.
“Accelerating Convergence of Iterative Processes.” Communications of the ACM, 1(6), 9–13, 1958.
Used for: The convergence-acceleration approach behind RECYCLE.

Kooijman, H.A.; Taylor, R.
The ChemSep Book (component property database and format).
Used for: The critical-properties file format (CriticalProperties.csv).