



# EBSILON®*Professional*

## Release Notes (English)

Release 15.0

[www.steag-systemtechnologies.com](http://www.steag-systemtechnologies.com)

**steag**



# Release Notes

## EBSILON®Professional

### Release 15.0

#### Contents

1	Calculation Kernel .....	8
1.1	New components .....	8
1.1.1	Battery (Component 158) .....	8
1.1.1.1	Overview .....	8
1.1.1.2	Pins .....	8
1.1.1.3	Flags .....	9
1.1.1.3.1	FOP – Operation Mode .....	9
1.1.1.3.2	FUTERM – Method for Calculating the Terminal Voltage .....	9
1.1.1.3.3	FLOSS – Loss Calculation .....	9
1.1.1.3.4	FTIMELIM – Behavior when Reaching the Limits .....	10
1.1.1.3.5	No FMODE .....	10
1.1.1.4	Specification Values .....	10
1.1.1.4.1	Specification Values for the Operation .....	10
1.1.1.4.2	Specification Values for the Geometry .....	11
1.1.1.4.3	Specification Values for Capacity and Voltage .....	11
1.1.1.4.4	Specification Values for the State .....	11
1.1.1.4.5	Specification Values for the Loss Calculation .....	11
1.1.1.5	Result Values .....	12
1.1.1.5.1	Result Values for the Operation .....	12
1.1.1.5.2	Result Values for the State .....	12
1.1.1.5.3	Result Values for the Loss Calculation .....	12
1.1.1.5.4	Other Result Values .....	12
1.1.1.6	Characteristic Lines and Characteristic Fields .....	12
1.1.2	Compressor (Map-Based) (Component 159) .....	12
1.1.3	Storage for Compressible Fluids (Component 160) .....	13
1.1.4	Injection with Temperature Control (Component 161) .....	13
1.1.5	Electric Boiler (Component 162) .....	13
1.2	Expansions of Existing Components .....	14
1.2.1	Total and Static State Variables .....	14
1.2.1.1	Basic Information .....	14
1.2.1.2	Result Values .....	15
1.2.1.3	Specifications .....	16

1.2.2	Shaft Pin for Turbines.....	16
1.2.3	Steam Generator (Components 5 and 139).....	17
1.2.3.1	Characteristic Line for Reheat Pressure Drops .....	17
1.2.3.2	New Result Values .....	17
1.2.4	Steam Turbine (Component 6).....	17
1.2.4.1	External Specification of the Polytropic Efficiency .....	17
1.2.5	Condensers and Preheaters (Components 7, 10, and 107).....	17
1.2.5.1	External Specification of the Pressure of the Auxiliary Condensate .....	17
1.2.6	Feedwater Tank (Component 9).....	18
1.2.6.1	External Specification of the Vapor Losses .....	18
1.2.6.2	External Specification of the Pressure of the Auxiliary Condensate .....	18
1.2.7	Generator (Component 11).....	18
1.2.7.1	X-Values of the Characteristic Lines .....	18
1.2.8	Controllers (Components 12, 39, 69).....	18
1.2.8.1	Hiding the Reference Value Pin in Component 69 .....	18
1.2.8.2	Zero Point Shift .....	18
1.2.8.3	Kernel Expressions for Range Limits .....	19
1.2.9	Piping (Component 13).....	19
1.2.9.1	Specific Volume for Geodesic Height.....	19
1.2.9.2	External Heat Transfer Coefficient According to VDI 2055 .....	19
1.2.10	Splitter with Ratio Specification (Component 18).....	20
1.2.10.1	Limitation M3MAX.....	20
1.2.11	Drum (Component 20).....	21
1.2.11.1	Specification of the Drain Mass Flow.....	21
1.2.11.2	Brine .....	21
1.2.12	Combustion Chamber and Duct Burner (Components 22 and 41).....	21
1.2.12.1	Hiding of the Slag Outlet .....	21
1.2.12.2	Second Fuel Pin.....	21
1.2.12.3	Combustion of Cooling Air .....	21
1.2.13	Gas Expander (Component 23) .....	22
1.2.13.1	Stodola Steam Cone Law .....	22
1.2.13.2	Result Value M1M1N.....	22
1.2.13.3	Result value MCORR .....	22
1.2.14	Compressor (Component 24) .....	22
1.2.14.1	Polytropic Efficiency.....	22
1.2.14.2	Result Value MCORR.....	23
1.2.15	Cycle Efficiency Meter (Component 32) .....	23
1.2.15.1	New Result Values .....	23
1.2.16	Value Transmitter (Component 36).....	23
1.2.16.1	Transmission of the Current.....	23
1.2.17	Water Injection (Component 38).....	23
1.2.17.1	Internal Specification of the Mixing Temperature.....	23
1.2.18	Value Indicator (Component 45).....	24
1.2.18.1	Elementary Analysis .....	24
1.2.18.2	Elementary Mass Flow .....	24
1.2.18.3	Composition of the Phases .....	24
1.2.18.4	Molar Mass Flow.....	24
1.2.18.5	Reactive and Apparent Power.....	24
1.2.18.6	Salinity .....	24
1.2.19	Measured Value Input (Component 46).....	25
1.2.19.1	Molar Mass Flow.....	25

1.2.20 Three-Way Valve (Component 49).....	25
1.2.20.1 Delayed Start of the Check of the Switching Criterion .....	25
1.2.20.2 Prevention of Frequent Switchover Processes.....	25
1.2.21 Gasifier (Components 50 and 96) .....	25
1.2.21.1 New Result Values .....	25
1.2.22 Steam Turbine (Extended) (Component 56).....	25
1.2.22.1 Flag FSPECQ for Specifying Power or Mass Flow .....	25
1.2.23 Governing Stage (Component 58) .....	26
1.2.23.1 New Result Values .....	26
1.2.24 General Mixer (Component 60).....	27
1.2.24.1 Treatment of the Voltage .....	27
1.2.25 Feedwater Tank / De-aerator (Extended) (Component 63).....	27
1.2.25.1 New Mode for Floating Pressure Operation.....	27
1.2.26 Programmable Component 65 / XUI Interface.....	27
1.2.27 Evaporator with Steam Drum (Component 70).....	27
1.2.27.1 New Result Values M2M2N and M3M3N .....	27
1.2.27.2 Pin for Pump Power.....	27
1.2.28 CHP Plant (Component 74) .....	28
1.2.28.1 New Fluids.....	28
1.2.29 Coal Dehumidifier (Component 84) .....	28
1.2.29.1 Logic Pin for Residual Moisture .....	28
1.2.30 Boiler: Bundle Heating Surface (Component 89).....	28
1.2.30.1 New Result Values .....	28
1.2.31 Reformer / Shift Reactor (Component 95).....	28
1.2.31.1 Flexible Treatment of Errors.....	28
1.2.32 Coupled Rectifier Column (Component 104).....	29
1.2.32.1 Kernel Expression for the Extractor Temperature.....	29
1.2.33 Direct Storage / Mass Storage (Component 118).....	29
1.2.33.1 Logic Pin for Mass Flow Request .....	29
1.2.33.2 Mass Flow Reduction when Reaching the Filling Level Limit.....	29
1.2.34 Diesel/Gas Motor (Reciprocating Engine) (Component 125).....	29
1.2.34.1 New Fluids.....	29
1.2.35 Stratified Storage (Component 145).....	30
1.2.35.1 Implementation and Monitoring of the Dimensionless Diffusion Number .....	30
1.2.36 Components 7 (Steam Turbine Condenser), 9 (Feedwater Tank), 10 (Feedwater Preheater), 20 (Drum) .....	30
1.2.36.1 Expansion by Transient Modeling .....	30
1.2.37 Component 124 (Heat Exchanger with Phase Transition).....	30
1.2.37.1 Implementation of a Numerical Algorithm .....	30
1.2.37.2 Pinpoint Violations.....	30
1.2.37.3 Convergence Tuning .....	31
1.2.37.4 Heat Transfer Coefficients .....	31
1.2.37.5 Result value KA.....	31
1.2.38 Component 126 (Transient Heat Exchanger).....	32
1.2.38.1 Implementation of an Effectivity Factor and Identification Mode.....	32
1.2.38.2 Reference of the Heat Exchange Surface Area to Different Pipe Diameters.....	32
1.2.38.3 Specification of the Material Properties of the Pipe Walls.....	32
1.2.38.4 Implementation of the Numerical Steady-State Solution .....	32
1.2.38.5 Option to Calculate the Heat Flow Reversal. Specification Value FREV.....	33
1.2.39 Component 119 (Indirect Storage).....	33
1.2.39.1 Specification of the Material Properties of the Storage Wall .....	33

1.2.40	Components 148, 149, 150 (Header) .....	33
1.2.40.1	Expansion by Further Fluids .....	33
1.2.41	Component 131 (Transient Separator) .....	33
1.2.41.1	New Specification Values .....	33
1.3	Physical Properties .....	34
1.3.1	New Function Calls .....	34
1.3.2	TREND: Support for Sea Water .....	34
2	User Interface .....	35
2.1	H-X Diagram for Humid Air .....	35
2.2	Component Matrices .....	35
2.2.1	3D Representation .....	35
2.3	Command Line Options .....	35
2.4	Multi-Simulation Bar .....	35
2.5	General Option .....	35
2.5.1	Loading the Common Language Runtime .....	35
2.6	Macro Objects .....	35
2.6.1	Grouping Specification and Result Values .....	35
2.7	Value Crosses .....	36
2.7.1	Alternative Data Connection by EbsScript Expression .....	36
2.7.2	Defining Values by EbsScript Expressions .....	36
2.8	Filtering Dropdown Boxes .....	37
2.9	Display of Specification Values .....	37
2.10	Time Series Dialog .....	37
2.10.1	Floating-Point Numbers Are Stored with Full Precision .....	37
2.10.2	Pasting Data via the Clipboard .....	37
2.10.3	New Commands in the Context Menu of the Head of the Line .....	37
2.11	Number of Characters in Text Fields .....	37
3	Additional Modules .....	38
3.1	EbsScript .....	38
3.1.1	New Functions .....	38
3.1.1.1	General Functions .....	38
3.1.1.2	Functions in the Interface Unit System .....	38
3.1.2	Logfile Support for the EbsScript.Python Interface .....	38
3.1.3	New Properties for the Type "ebsComp" .....	39

3.1.4	New Properties for Model/Profile Variables.....	39
3.1.5	Expansion of the Formatting for printToString.....	39
3.1.6	Unit Tests .....	40
3.1.7	Expanded Topological Access to Components and Lines .....	40
4	Changes in the Results .....	41
4.1	Component 11 (Generator).....	41
4.2	Component 119 (Indirect Storage).....	41
4.3	Component 126 (Transient Heat Exchanger).....	41
4.4	Default Value for the Reference Pressure .....	41
4.5	Minimum Mass Flow for the Controller.....	41
5	Known Errors.....	43
5.1	Component 124 – Numerical Solution .....	43
5.2	Documentation.....	43

# 1 Calculation Kernel

## 1.1 New components

### 1.1.1 Battery (Component 158)

#### 1.1.1.1 Overview

This component represents a battery consisting of individual cells with a nominal capacity CCELLN and a nominal voltage UCELLN. The actually available capacity decreases according to a factor SOH (state of health) that is specified to the battery by the aging module via Logic Pin 4. The prevailing state of charge of the cell SOC must remain between the limits SOCMIN and SOCMAX.

The component Battery consists of NBANK banks connected in parallel that consist of NRACK racks connected in parallel in each case. Each rack consists of NMOD modules connected in series. Each module consists of NPAR \* NSER modules (connected per NPAR in parallel and NSER in series).

When discharging the battery, the terminal voltage decreases under its nominal value down to a minimum value UCOCELL (cut-off voltage). Here the current can rise up to the short circuit current ISCELL.

The terminal voltage as a function of the discharge current and of the state of charge (SOC) is defined by a characteristic field CUACTION that consists of 20 characteristic lines with the prevailing state of charge as parameter. Instead of the characteristic field, it is also possible to calculate with a constant internal resistance of the cell, which can optionally be specified directly or can be calculated from the short circuit current.

The component can be applied both in electric circuits, where it calculates the current based on the resistances in the electric circuit, and in control loops that require the current to be specified externally. It is also possible to define the current by specifying the desired power on Logic Inlet 5.

To display the aging of the battery, an external aging module has to be connected to Logic Inlet 4. Via this logic inlet, the battery receives its respective prevailing SOH ("state of health") from the aging module. It is defined as the ratio of the prevailing to the original capacity of the battery. The aging module, in turn, can access the the data of the battery via this logic pin and calculate the aging.

#### 1.1.1.2 Pins

In most Epsilon components, the fluid flows into the component via Pin 1 and out of the component via Pin 2. In the case of the component Battery, however, the same pin (namely **Pin 2**) is used for charging and for discharging.

As, topologically, it is an outlet, positive values are displayed for the power when discharging. The charging process is displayed by means of negative values. On the one hand, this is closer to reality (as in reality the same cables are used for charging and discharging) and on the other hand it facilitates the modeling.

Nevertheless, there is also an electric inlet (**Pin 1**) in this component, which, however, is optional and not used for charging but for connecting another battery. This way, several Components 158 can be connected in series. In doing so, the voltages will add up while the current remains the same through all batteries. With the exception of the last one, all batteries must be operated in the mode FOP=-1 ("Current given externally") as the last battery carries out the calculation for the overall package and writes the current onto the line.



**Pin 3** is a logic outlet for the losses. These are calculated by the battery. Here both the losses occurring in the battery and the power required for the cooling can be considered (see Chapter 1.1.1.3.3).

**Pin 4** is the logic inlet for the connection of the aging module: via this connection, the aging module receives the required data of the battery. The aging module calculates the new SOH (state of health) at the end of the respective time step and communicates it to the battery via the logic line at the beginning of the next time step.

**Pin 5** is the logic inlet for the control. The desired power is specified on this line. When the battery is to be charged, a negative value has to be specified. Whether the desired power can be achieved depends both on the state of charge of the battery and on the current:

- If the level of the battery drops below SOCMIN, the discharge current will be set to 0 in steady state. In transient state, either the power can be reduced or the time interval can be shortened (see 1.1.1.3.4).
- If the level of the battery exceeds SOCMAX, the charging current will be set to 0 in steady state. In transient state, either the power can be reduced or the time interval can be shortened (see 1.1.1.3.4).
- If the charge rate CRATE is exceeded by the required current, the current is reduced so far that a charge rate of CRATE is adhered to (see 1.1.1.4.1).
- If the discharge rate DCRATE is exceeded by the required current, the current is reduced so far that a discharge rate of DCRATE is adhered to (see 1.1.1.4.1).

### 1.1.1.3 Flags

#### 1.1.1.3.1 FOP – Operation Mode

- FOP=-1: Current given externally
- Here the voltage results from the internal resistance of the battery cells and the characteristic field respectively (see 1.1.1.3.2). When discharging ( $I > 0$ ), the voltage decreases with increasing current. When charging ( $I < 0$ ), increasing the voltage is required for increasing the current.
- FOP= 0: Out of order
- Current and voltage are set to 0 in this case.
- FOP= 1: Charging: the voltage is specified externally, the current is calculated via the internal resistance or via the characteristic field (see 1.1.1.3.2). Here the applied voltage must be greater than the nominal voltage of the battery.
- FOP=2: Discharging: voltage and current are calculated from the resistance in the electric circuit.
- FOP=3: Power control according to power demand on Pin 5

#### 1.1.1.3.2 FUTERM – Method for Calculating the Terminal Voltage

This flag serves to define the correlation between terminal voltage and current. By default, the calculation is effected with a constant internal resistance. However, it is also possible to store a characteristic field.

- FUTERM=0: the internal resistance of the cell is specified directly in RINTCELL.
- FUTERM=1: the internal resistance of the cell is calculated from the short circuit current ISCCCELL.
- FUTERM=2: the internal resistance of the cell is calculated from the maximum power QMAX.
- FUTERM=3: the terminal power is determined from the characteristic field FCUTERM as a function of the current and the state of charge of the battery.

#### 1.1.1.3.3 FLOSS – Loss Calculation

Losses are calculated by the component itself and are written as heat on Logic Outlet 3. There are the following variants for calculating the losses that occur during the operation of the battery:

- FLOSS=0: Heat loss calculated from voltage and current:

$$QLOSSOP = (UTOTN - U2) * I2$$

Please note that the loss becomes positive at all times because while charging  $I2 < 0$  and  $U2 > UTOTN$ .

- FLOSS=1: Heat loss proportional to the power:  $QLOSSOP = LOSS * |Q2|$
- FLOSS=2: Heat loss proportional to the current:  $QLOSSOP = LOSS * |I2|$

Here  $U2$ ,  $I2$  and  $Q2$  are voltage, current and power at the outlet and the proportionality factor  $LOSS$  is a specification value.

Moreover, for the losses it is also possible to consider the auxiliary power  $QLOSSC$  of the air conditioner required for cooling the batteries. As it depends on the ambient temperatures, there is a characteristic line  $CQLOSSC$  for this purpose. This characteristic line is used both in the design case and in off-design. Therefore the design temperature of the air conditioner (i.e. the temperature where the characteristic line has the value 1) does not have to be consistent with the design case of the overall model.

It is assumed that the air conditioner can be operated independently of the battery. Therefore the auxiliary power is also calculated when the battery is turned off.

#### 1.1.1.3.4 FTIMELIM – Behavior when Reaching the Limits

This flag allows to set the behavior for a transient calculation, e.g. in the context of a time series.

Within the time step, the component carries out a steady-state calculation as a rule, i.e. no time dependencies are considered. Only when calculating the result values  $SOCNEW$  (new state of charge at the end of the time step) and  $DCHARGE$  (change of charge in the time step) does a time dependency come into play.

As long as  $SOCNEW$  is within the valid range between  $SOCMIN$  and  $SOCMAX$ ,  $FTIMELIM$  has no effect. However, if  $SOCNEW$  exceeds a limit, there are the following options:

- $FTIMELIM=0$ : the power is reduced so far that the limit is only reached at the end of the time step.
- $FTIMELIM=1$ : the power is left unchanged but the time interval is shortened, i.e. in the time series table an additional line is inserted with the point in time when the limit is reached.

In both cases,  $SOCNEW$  is then set to the respective limit value ( $SOCMIN$  or  $SOCMAX$ ).

When carrying out a time series calculation, after calculating a time step the result value  $SOCNEW$  is then copied onto the specification value  $SOC$  for the next time step.

#### 1.1.1.3.5 No FMODE

A flag  $FMODE$  for switching over between design and off-design mode does not exist in this component as there are no nominal values for this component that might be calculated by Epsilon in the context of a design calculation.

### 1.1.1.4 Specification Values

#### 1.1.1.4.1 Specification Values for the Operation

These specification values are only relevant for the operation with power demand ( $FOP=3$ ).

CRATEN: nominal value for the charge rate (CRATE)

The charge rate indicates the proportion of the capacity per time unit that is charged and discharged respectively. For instance: at a CRATE of 0.5 A/Ah, the SOC (state of charge) of the battery is increased and decreased respectively by 0.5 in one hour.

The charge rate results from the nominal value CRATEN and the characteristic line CCRATE as a function of the state of charge. By default, this characteristic line is designed in such a way that the charge rate is already reduced when approaching SOC=1.

DCRATEN: nominal value for the discharge rate (DCRATE) (optional)

Only enter a value here if another rate is to be used for discharging than for charging. Otherwise the discharge rate is equated with the charge rate.

STHR: threshold value for charging

Here you can enter a threshold value that has to be exceeded for the battery to switch over to "Charge" mode.

#### **1.1.1.4.2 Specification Values for the Geometry**

NPAR: number of parallel cells per module

NSER: number of serial cells per module

NMOD: number of modules per rack

NRACK: number of racks per bank

NBANKS: number of banks in the component

#### **1.1.1.4.3 Specification Values for Capacity and Voltage**

UCELLN: nominal voltage of a cell

CCELLN: nominal capacity (maximum charge) of a cell

RINTCELL: internal resistance of a cell

ISCCCELL: short circuit current (max. possible current) of a cell

QMAX: maximum power of the total battery

#### **1.1.1.4.4 Specification Values for the State**

SOC: current state of charge

SOCMIN: min. admissible state of charge

SOCMAX: max. admissible state of charge

The "state of health" (SOH) is not managed in the battery but in the aging module. The battery only shows in the result values which SOH the calculation was carried out with.

#### **1.1.1.4.5 Specification Values for the Loss Calculation**

LOSS: loss factor (proportionality factor at FLOSS=1 or 2)

QLOSSCN: nominal value for the power of the air conditioner

This value is multiplied by the value of the characteristic line CQLOSS which depends on the ambient temperature.

TAMB: ambient temperature (for FSTAMB=0)

ISUN: link to Component 117 (for FSTAMB=1)

### **1.1.1.5 Result Values**

#### **1.1.1.5.1 Result Values for the Operation**

RCRATE: applied charge and discharge rate respectively

UCELL: present voltage of a cell

UTOT: present total voltage

ICELL: present current through a cell

ITOT: present total current

QTOT: total battery power

#### **1.1.1.5.2 Result Values for the State**

SOCNEW: new state of charge at the end of the time step

SOH: state of health used for the calculation

CTOT: total capacity at the current state of health

DCHARGE: change of charge during the time step

DCHARGECTOT: Charge change relative to total capacity

#### **1.1.1.5.3 Result Values for the Loss Calculation**

RTAMB: applied ambient temperature

QLOSSOP: loss due to the operation of the battery

QLOSSC: loss due to the air conditioner

QLOSSTOT: total loss

#### **1.1.1.5.4 Other Result Values**

UTOTN: nominal voltage of the total battery

UUN: ratio of present to nominal voltage

IISC: ratio of present to short circuit current

RRINTCELL: used internal resistance, one cell

RRINTTOT: used internal resistance of the total battery

RQMAX: calculated maximum power of the battery

This result value is calculated from the internal resistance assuming that it is constant. If this is not the case (possible at FUTERM=3), this value is only an approximation.

QQMAX: Ratio actual power / maximum power

#### **1.1.1.6 Characteristic Lines and Characteristic Fields**

CRATE: relative charge rate as a function of the state of charge

CQLOSSC: relative power of the air conditioner as a function of the ambient temperature

FCUTERM: characteristic field, consisting of characteristic lines CUTERM\_n for the relative terminal voltage as a function of the relative cell current (in relation to the short circuit current ISCELL) with the state of charge SOC as parameter

### **1.1.2 Compressor (Map-Based) (Component 159)**

This component serves to model a compressor with compressor map.

### 1.1.3 Storage for Compressible Fluids (Component 160)

This component uses identical physical algorithms as Component 119 (Indirect storage). In contrast to Component 119, Component 160 enables an explicit specification of the modes of operation:

- simultaneous charging and discharging
- charging
- discharging
- cooling down (no inflow or outflow of the fluid)

It is important that the fluid in the storage system has a correlation between the fluid pressure and the specific volume/density of the fluid and that this correlation can be described by the corresponding material table. Hence the name of the component. For further details please refer to the EBSILON Online Help.

### 1.1.4 Injection with Temperature Control (Component 161)

This component serves to model a controlled injection. It represents a combination of Components 38 (Water injection) and 39 (Controller with internal target value).

The special feature of this component is the actual value pin (Pin 4). From there, an actual value line can be dragged to any position in the model where a desired temperature is to be achieved. Thus by means of this component it is possible to set a temperature that is not present at the component itself.

The specification value TSCV serves to define which temperature is to be set. With the flag FTSCV, you can select whether you want to directly enter the temperature (FTSCV = 0) or a superheating (FTSCV = 1) or a supercooling (FTSCV = -1).

Internally, Component 161 works like a controller. The change of the values during the iteration can be observed by means of the controller process diagram, which is available in the context menu (right mouse button). In the event of trouble with the convergence, the corresponding tuning measures are available.

As a mass flow is varied here at all times in order to set a temperature, there are no flags for this. The flag FCHAR for setting the characteristic does exist but is set to 0 by default because the component tries to automatically determine the characteristic:

- If the injected fluid can evaporate, it is assumed that the characteristic is negative, i.e. an increase in the spray flow will always lead to a cooling.
- Otherwise the characteristic will be set to positive if the injected fluid is warmer than the main inlet flow, otherwise negative.

If this does not apply or the situation is more complex, the characteristic can also be specified manually though.

### 1.1.5 Electric Boiler (Component 162)

This component serves to evaporate or to heat up a fluid (water by default) by means of electric energy. Here the initial state is either saturated steam (FOUT=0) or water with a temperature specified in the specification value T2 (FOUT=1).

Here optionally the electric power (FSPEC=0) or the desired steam and hot water production respectively (FSPEC=1) can be specified. It is also possible to specify both variables (FSPEC=-1). In this case, the outlet state will be calculated.

## 1.2 Expansions of Existing Components

### 1.2.1 Total and Static State Variables

#### 1.2.1.1 Basic Information

Previously it was assumed in Epsilon that all fluids flow slowly enough so that the kinetic energy can be neglected and total and static state variables can be equated. Therefore no distinction was made between total and static values.

The only exception was the steam turbine (Component 122). As proportionately high flow velocities occur here, the kinetic fractions were considered in the calculation of the steam turbine. In doing so, it was assumed that the variables stored on the lines are total variables as a rule.

In Release 15, it is now possible to differentiate between total and static variables everywhere. Here the assumption previously made for the steam turbine, namely that the line values are total variables, is applied to all lines.

The conversion between total and static variables requires knowledge of the flow velocity  $vel$  (as in Epsilon the abbreviation  $v$  is already used for the specific volume,  $v$  is not available for the velocity).

In Epsilon default units ( $vel$  in m/s,  $H$  in kJ/kg), the following applies:

$$H_{tot} = H_{stat} + H_{kin}, \text{ with} \\ H_{kin} = 0.0005 * vel^2$$

Here the flow velocity can optionally be specified directly or can be calculated from the volume flow when specifying the flow cross section. If the pipe has a circular cross section, the inner diameter can be specified instead of the cross section as well.

In the case of the boundary input value (Component 1) or the start value (Component 33), the specification is effected by means of one of the specification values  $VEL\_SET$ ,  $A\_SET$  or  $D\_SET$ . As a result, all three variables  $VEL$ ,  $A$ , and  $D$  will be displayed.

Additionally, the potential energy can be considered in Release 15 as well. The following applies to it in Epsilon default units ( $z$  in m,  $H$  in kJ/kg):

$$H_{pot} = 0.001 * g * z,$$

where  $g = 9.81 \text{ m/s}^2$  is the acceleration of gravity and  $z$  the height, relating to a selected zero level. In Component 1 and 33 respectively, the height is specified by a specification value  $Z\_SET$  and displayed as result value  $Z$ .

Altogether, the following results:

$$H_{tot} = H_{stat} + H_{kin} + H_{pot}$$

The distinction between total and static variables only occurs within Component 1 and 33 respectively. It does not make sense to treat the flow velocity as an attribute of the line because the line cross section and thus the flow velocity can change from one position of the line to the next. Therefore the specification and calculation must always be effected together with the specification of the speed (and cross section or diameter respectively).

To allow to observe different flow velocities at various positions of the line, it is possible to set several Components 33 on a line. Specifying static variables, however, is only possible in one component.

### 1.2.1.2 Result Values

If the flow velocity and height respectively is known, the following result values will be displayed in Component 33:

- The total pressure PTOT  
It equals the pressure P shown on the line.
- The static pressure PSTAT  
It results from the condition that the entropy  $s=s(p_{tot}, h_{tot})$  calculated from the total variables equals the entropy calculated from the static variables, so that the static pressure can be calculated from  $p_{stat} = p(h_{stat}, s)$ .
- The kinetic pressure increase DPKIN  
It equals the difference between total and static pressure.
- The total temperature TTOT  
It equals the temperature T shown on the line.
- The static temperature TSTAT  
It results from the condition that the entropy  $s=s(p_{tot}, h_{tot})$  calculated from the total variables equals the entropy calculated from the static variables, so that the static temperature can be calculated from  $t_{stat} = t(h_{stat}, s)$ .
- The kinetic temperature increase DTKIN  
It equals the difference between total and static temperature.
- The total enthalpy HTOT  
It equals the enthalpy H shown on the line.
- The static enthalpy HSTAT  
It results from  $H_{STAT} = H_{TOT} - H_{KIN} - H_{POT}$ .
- The kinetic energy HKIN
- The potential energy HPOT
- The internal energy UIE = HSTAT – HPV  
(The usual abbreviation U for the internal energy cannot be used here as in Epsilon, U is used as the abbreviation for the voltage.)
- The displacement energy HPV  
It results from  $HPV = P_{STAT} * V_{STAT}$ .
- The total density RHOTOT  
It equals the density RHO shown on the line.
- The static density RHOSTAT  
It is the reciprocal value of the static specific volume VSTAT.
- The static specific volume VSTAT  
It is calculated from the property tables as  $V_{STAT} = V(P_{STAT}, H_{STAT})$ .
- The static volume flow VMSTAT  
It results from the mass flow and the static specific volume to  $VM_{STAT} = M * V_{STAT}$ .
- The flow velocity VEL
- The inner diameter D of the pipe
- The flow cross section A
- The Mach number MACH  
It results from the flow velocity and the sonic velocity W to  $MACH = VEL/W$ .

### 1.2.1.3 Specifications

Component 1 and 33 respectively do not only enable the display but also the specification of static state variables. Thus static variables given in thermal flow diagrams or also measured static variables can now be specified this way.

For this – with the exception of the last case – the flow velocity, as described above, must have been specified either directly or via cross section or diameter.

The following options have been implemented:

- Specification of the static pressure (instead of the total pressure) when using the total enthalpy of the line that can be given in the same component (as total temperature or total enthalpy or total heat flow) or otherwise
- Specification of the static temperature (instead of the total temperature or the total enthalpy or the total heat flow) when using the total pressure of the line that can be given in the same component or otherwise
- Specification of the static pressure and the static temperature instead of the total variables
- Specification of the static pressure, the static temperature, and the total pressure  
In this case, the flow velocity is calculated.

If, in this case, the cross section or the diameter is specified, the mass flow is determined from the flow velocity.

This case is particularly interesting when static and total pressures are measured by means of various pressure gauges (Prandtl tube, Pitot-Static tube) in order to determine the throughput.

### 1.2.2 Shaft Pin for Turbines

Previously, the second shaft pin of Component 6 (Steam turbine) and Component 58 (Governing stage) was a shaft inlet, and in Component 23 (Gas expander) a shaft outlet. Thus in the case of Components 6 and 58, it was to connect several turbine disks in series, so the shaft power added up, while in the case of Component 23, a distribution of the shaft power to compressor and generator could be displayed.

The reverse case (connection in series of Component 23 and power distribution for Component 6 and 58) could not be graphically represented so far. However, there was a flag FQ (power flow) for the calculation that allowed to switch it over although it had the imperfection that the graphic representation did not match the calculation then.

In Release 15, the possibility to also graphically represent the reverse power flow has now been created for these three components. An additional shaft pin has been implemented for this purpose:

- in Component 6 Pin 8 as shaft outlet
- in Component 23 Pin 6 as shaft inlet
- in Component 58 Pin 5 as shaft outlet

To enable a corresponding display, the previously existing pin can now be hidden and the new pin has been positioned in the same place. Usually, either the inlet or the outlet will be used and the unused pin should be hidden then. As a rule, however, the software also allows to simultaneously use both pins.

As in the case of the previous second shaft pin, the power must be specified on the new pin too. The turbine can only calculate the power on the main shaft outlet.



The new pin has made the flag FQ obsolete. For reasons of compatibility, however, it is still available but has been marked as “deprecated”. Moreover, a comment may be output that points out the options for using the new pin. For the new shaft pin, the flag produces a reversal of the calculation direction as well.

For all three components and also for Component 56, a result value QSHAFT has been implemented that outputs the shaft power generated in the component, regardless of which pins it is distributed to or which further shaft power is added.

## **1.2.3 Steam Generator (Components 5 and 139)**

### **1.2.3.1 Characteristic Line for Reheat Pressure Drops**

Previously, only the off-design behavior of the pressure drop on the high-pressure side could be specified by the user by means of a characteristic line. The pressure drop of the reheat was calculated by the program on the basis of the mass and volume flows respectively.

Now characteristic lines are available for the pressure drop of the reheat as well (CDP34 for Components 5 and 139, CDP910 for Component 139). For these to be used, the flag FVOL has to be set to 3.

### **1.2.3.2 New Result Values**

To facilitate the generation of the characteristic lines, the following result values have been implemented in Components 5 and 139:

- $M1M1N = M1 / M1N$
- $M3M3N = M3 / M3N$
- $P2P2N = P2 / P2N$
- $DP12DP12N = DP12 / DP12N$
- $DP34DP34N = DP34 / DP34N$

In Component 139 additionally for the second reheat

- $M9M9N = M9 / M9N$
- $DP910DP910N = DP910 / DP910N$

## **1.2.4 Steam Turbine (Component 6)**

### **1.2.4.1 External Specification of the Polytropic Efficiency**

Previously, switching over to external specification of the efficiency on Logic Pin 7 using the flag FVALETAI was only possible when using the isentropic efficiency (FETA=0). Now the external specification is also applied to the polytropic efficiency (at FETA=1).

## **1.2.5 Condensers and Preheaters (Components 7, 10, and 107)**

### **1.2.5.1 External Specification of the Pressure of the Auxiliary Condensate**

Previously, the pressure of the auxiliary condensate was always set by the component as the auxiliary condensate is at the same pressure level as the condensate. When modeling, it was therefore necessary to place a control valve or a condensate valve on the auxiliary condensate line in order to decrease the pressure to the condenser level.

To simplify the modeling, there is now a mode “P5 given externally” that can be set by means of the flag FP5. This mode allows to connect a line with a higher pressure on Pin 5. Within the component, the auxiliary condensate is then reduced to the condenser pressure. The result is the same as with an external control valve.

The new mode is now the default setting for newly inserted components. For existing models, FP5 is set to “P5=P3”.

## **1.2.6 Feedwater Tank (Component 9)**

### **1.2.6.1 External Specification of the Vapor Losses**

The vapor losses can now optionally be specified via specification value M5 (as before) or set on the line externally. Switching over between the two modes of calculation is effected by means of the flag FM5.

### **1.2.6.2 External Specification of the Pressure of the Auxiliary Condensate**

Previously, the pressure of the auxiliary condensate was always set by the feedwater tank as the auxiliary condensate is at the same pressure level as the feedwater at the outlet. When modeling, it was therefore necessary to place a control valve or a condensate valve on the auxiliary condensate line in order to decrease the pressure to the condenser level.

To simplify the modeling, there is now a mode “P4 given externally” that can be set by means of the flag FP4. This mode allows to connect a line with a higher pressure on Pin 4. Within the feedwater tank, the auxiliary condensate is then reduced to the condenser pressure. The result is the same as with an external control valve.

The new mode is now the default setting for newly inserted components. For existing models, FP4 is set to “P4=P2”.

## **1.2.7 Generator (Component 11)**

### **1.2.7.1 X-Values of the Characteristic Lines**

Previously, all characteristic lines always referred to the ratio of prevailing shaft power to nominal shaft power. As, however, many manufacturers have their characteristic lines refer to the electrical power, Epsilon now allows to use the flag FCHRX to switch over from Q1/Q1N (FCHRX=1) to Q2/Q2N (FCHRX=2).

## **1.2.8 Controllers (Components 12, 39, 69)**

### **1.2.8.1 Hiding the Reference Value Pin in Component 69**

As in the case of this controller the reference value can also be specified internally (flag FSCV), it is now possible to hide this pin. This is done in the tab “Ports” of the Component Properties.

### **1.2.8.2 Zero Point Shift**

As in the case of controllers in Epsilon the change of the actuating variable is effected via a change factor, controllers previously could not be operated in such a way that the actuating variable was able to change its algebraic sign. To enable this, there is now a specification value CZP that serves to shift the zero point

of the controller internally. Shifting is effected in positive direction. If e.g. "100" is entered, -100 will be mapped onto 0 and it will also be possible to control beyond 0 in the range >-100.

With great values of CZP, the internal actuating variable will become great accordingly so that this way, in the case of similar relative changes, the absolute change of the actuating variable will become very great too. This may lead to convergence problems. In this case, it is recommended to decrease the maximum change factor (CHL2 and CHL3 respectively), and that from the beginning (ITCHL2 and ITCHL3 = 0 respectively).

### 1.2.8.3 Kernel Expressions for Range Limits

Previously, only fixed values could be entered as range limits for the actuating variable. Now it is possible to use a Kernel expression as the limit. To do so, the corresponding flag (FL2MIN and FL2MAX respectively for Component 39, FL3MIN and FL3MAX respectively for Components 12 and 69) has to be set to "Kernel expression", and an EbsScript that calculates the corresponding limit has to be created in EL2MIN and EL2MAX respectively for Component 39 or EL3MIN and EL3MAX respectively for Components 12 and 69.

This feature was needed for the variation of the steam inlet pressure for a preheater in order to achieve a certain feedwater outlet temperature. Without limitation, the controller decreased the pressure so far that the saturated water temperature dropped below the feedwater inlet temperature and no condensation was possible anymore. A fixed limit, however, was not possible either as the feedwater inlet temperature is not known in advance but only appears in the course of the calculation. For instance, the following Kernel expression allows to set the lower pressure limit to a reasonable value in each iteration step:

```
function evalexpr:REAL;  
  
begin  
  
    evalexpr:=waterSteamTable(1006, Feedwater.T, 0.0);  
  
end;
```

## 1.2.9 Piping (Component 13)

### 1.2.9.1 Specific Volume for Geodesic Height

Previously, the arithmetic mean value of the specific volume V1 at the inlet and V2 at the outlet was used for calculating the pressure difference due to the geodesic height at all times. Now it is possible to optionally use V1 or V2 as well. The switchover is effected by means of the new flag FVOLGH.

If the mean value is used and the specific volume at the outlet differs from that at the inlet by a factor of 2 or more, a warning will now be output.

### 1.2.9.2 External Heat Transfer Coefficient According to VDI 2055

When calculating the heat losses from the geometry (FDN = 8), it is now possible not to use the specification value ALPHO for the external heat transfer coefficient but to have this value calculated as a function of the ambient temperature and the wind speed according to VDI 2055.

It must be pointed out that in this component (other than in Component 118) there is no calculation of the temperature gradient in the pipe wall. The outer wall temperature included in the formulae for VDI 2055 is therefore equated with the fluid temperature here.

The external heat transfer coefficient  $\alpha_k$  results from the heat transfer coefficient  $\alpha_{k,\text{free}}$  of the free convection in still air and the heat transfer coefficient  $\alpha_{k,\text{forc}}$  of the forced convection according to

$$\alpha_k = (\alpha_{k,\text{free}}^4 + \alpha_{k,\text{forc}}^4)^{1/4}$$

For the free convection, the following applies at laminar air flow:

$$\alpha_{k,\text{free}} = 1.22 * (\Delta T / d_{\text{out}})^{1/4} \quad (\text{in W}/(\text{m}^2\text{K}))$$

with the temperature difference

$$\Delta T = T_{\text{av}} - T_{\text{amb}} = 0.5 * (T_2 + T_1) - T_{\text{amb}} \quad (\text{in K})$$

and the outer diameter of the insulation

$$d_{\text{out}} = \text{DINNER} + 2 * \text{THPIPE} + 2 * \text{THISO} \quad (\text{in m})$$

In the case of turbulent air flow, the following applies:

$$\alpha_{k,\text{free}} = 1.21 * (\Delta T)^{1/3} \quad (\text{in W}/(\text{m}^2\text{K}))$$

Turbulent air flow sets in under the following condition

$$\text{crit}_{\text{free}} = d_{\text{out}}^3 * \Delta T > 1 \text{ m}^3\text{K}$$

The forced convection depends on the wind speed  $v_{\text{wind}}$ . The following condition is considered to be the criterion for the change from laminar to turbulent here:

$$\text{crit}_{\text{forc}} = d_{\text{out}}^3 * v_{\text{wind}} > 0.00855 \text{ m}^2/\text{s}$$

In the case of laminar flow, the following applies to the forced convection:

$$\alpha_{k,\text{forc}} = 0.0081 / d_{\text{out}} + 3.14 * (v_{\text{wind}} / d_{\text{out}})^{1/2} \quad (\text{in W}/(\text{m}^2\text{K}))$$

and in the case of turbulent flow:

$$\alpha_{k,\text{forc}} = 2 * v_{\text{wind}} + 3 * (v_{\text{wind}} / d_{\text{out}})^{1/2} \quad (\text{in W}/(\text{m}^2\text{K}))$$

The values for the criteria, the contributions of the free and the forced convection to the external heat transfer coefficient as well as the calculated heat transfer coefficient are displayed as result values.

Here the ambient temperature and wind speed can optionally be specified in the component or read from a sun or wind data component with the corresponding index ISUN and IWDATA respectively. This is set via the flags FSTAMB and FSWWIND respectively.

## 1.2.10 Splitter with Ratio Specification (Component 18)

### 1.2.10.1 Limitation M3MAX

The new specification value M3MAX allows to limit the mass flow of the splitter. If a mass flow greater than M3MAX results from the splitting ratio, the splitting ratio will be reduced so far that the mass flow

of the splitter equals M3MAX. The splitting ratio that is actually used will be displayed in the result value RM3M1.

M3MAX can also be used for mechanical shafts and electric lines. In this case, the desired maximum power of the shaft and electric line respectively has to be entered at M3MAX.

## **1.2.11 Drum (Component 20)**

### **1.2.11.1 Specification of the Drain Mass Flow**

Previously, the drain mass flow M5 had to be specified externally on the line. Now there is a flag FM5 that allows to also have this mass flow set by the drum, optionally in absolute or relative terms.

There are the following variants:

- FM5=-1: M5 is set externally (as before)
- FM5=0: absolute specification in specification value M5S ( $M5=M5S$ )
- FM5=1: specification relative to the feedwater inlet ( $M5=M5S*M1$ )
- FM5=2: specification relative to the steam production ( $M5=M5S*M2$ )

### **1.2.11.2 Brine**

The drum can now be operated with brine too. Here it must be pointed out that no salt is contained in the steam. As in steady state the same amount of salt has to be discharged as is supplied and only the drain is available for the discharge, the drain mass flow must be great enough for the salt to remain dissolved.

It is assumed that the salinity in the drum and also in the recirculation corresponds to that of the drain. In the evaporator, care has to be taken that the steam mass fraction remains small enough so that the salt can still be dissolved in the liquid phase.

## **1.2.12 Combustion Chamber and Duct Burner (Components 22 and 41)**

### **1.2.12.1 Hiding of the Slag Outlet**

As usually no slag accumulates in the operation with gas, the slag outlet (Pin 5) is now hidden by default. If needed, it can be shown again in the tab "Ports" of the Component Properties.

### **1.2.12.2 Second Fuel Pin**

As previously with Components 21 and 90, there is now a second fuel pin in Components 22 and 41 too for e.g. displaying an oil firing. By default, however, it is hidden. If needed, it can be shown again in the tab "Ports" of the Component Properties.

### **1.2.12.3 Combustion of Cooling Air**

Previously, the cooling air (Pin 3) was added to the exhaust gas only downstream of the combustion in order to decrease its temperature. It did not take part in the combustion as a rule. When it was impossible to provide enough oxygen for the combustion via Pin 1, an error message was output.

In Release 15, it is possible to have the cooling air participate in the combustion too. For this purpose, there is a flag FCOOLAIR:

- For FCOOLAIR=0, the components behave as previously, i.e. the cooling air does not take part in the combustion. This is the default setting.
  - For FCOOLAIR=1, the cooling air takes part in the combustion.
- The cooling water mass flow has to be specified externally in either case.

## 1.2.13 Gas Expander (Component 23)

### 1.2.13.1 Stodola Steam Cone Law

Previously, the simplified formula for ideal gases was used for calculating the Stodola pressure in this component. As the flue gas is calculated as ideal gas by default, this approximation is definitely justified.

Meanwhile, however, Epsilon allows to activate a real gas correction for the flue gas. As in this case, the application of the ideal gas Stodola formula is inconsistent, the more detailed formula for real gases is used now.

An option to switch over between ideal gas and real gas formula as in Component 6 is not provided here. In Component 6, it only exists for historical reasons anyway and should not be used in new models.

### 1.2.13.2 Result Value M1M1N

The ratio of the current mass flow M1 to its design value M1N is now output as result value M1M1N. As this is the x-value for the efficiency characteristic line, the creation of the characteristic line is thus facilitated.

### 1.2.13.3 Result value MCORR

There is now a result value MCORR that specifies which mass flow would flow through the expander under standard conditions. The calculation is carried out according to [https://en.wikipedia.org/wiki/Corrected\\_flow](https://en.wikipedia.org/wiki/Corrected_flow):

$$MCORR = M1 * \text{SQRT}(T1[K]/TNORM) / (P1/PNORM)$$

with TNORM = 288.15 K and PNORM = 1.01325 bar.

## 1.2.14 Compressor (Component 24)

### 1.2.14.1 Polytropic Efficiency

As an alternative to the isentropic efficiency, the polytropic efficiency can now be output too in the case of the compressor. This is done in the specification value ETAPN. The flag FETA serves to select whether the isentropic or the polytropic efficiency is to be used.

When using the polytropic efficiency, the isentropic efficiency is calculated as well as the result value ETAI and transferred to the specification value ETAIN. This makes it easier to switch over to the isotropic efficiency, saving computing time as the polytropic efficiency requires a complex numerical integration. This makes itself felt in the case of complex physical properties in particular.

In the reverse case, ETAP is usually not calculated; however, there is a flag FOUTETAP by means of which this calculation can be activated. In this case, the calculated value of ETAP is also transferred to the specification value ETAPN.

### 1.2.14.2 Result Value MCORR

There is now a result value MCORR that specifies which mass flow would flow through the compressor under standard conditions. The calculation is carried out according to [https://en.wikipedia.org/wiki/Corrected\\_flow](https://en.wikipedia.org/wiki/Corrected_flow):

$$MCORR = M1 * \sqrt{T1[K]/TNORM} / (P1/PNORM)$$

with  $TNORM = 288.15 \text{ K}$  and  $PNORM = 1.01325 \text{ bar}$ .

## 1.2.15 Cycle Efficiency Meter (Component 32)

### 1.2.15.1 New Result Values

Apart from the efficiency ETA related to the net calorific value (NCV), the efficiency related to the gross calorific value is now displayed as result value ETA\_GCV too:

$$ETA\_GCV = Q1 / (M2 * (H2 + GCV2))$$

Moreover, there are now result values for the heat rate related to the net and the gross calorific value:

$$HR [kJ/kWh] = 3600 * M2 * (H2 + NCV2) / Q1$$
$$HR\_GCV [kJ/kWh] = 3600 * M2 * (H2 + GCV2) / Q1$$

## 1.2.16 Value Transmitter (Component 36)

### 1.2.16.1 Transmission of the Current

In the case of the value transmitter, a transmission of the current is now possible as well by means of  $FIN=15$  and  $FOUT=15$  respectively. Internally, the enthalpy is transmitted in this case; however, it is provided with appropriate prefactors so that the current remains the same (as long as  $MUL=1$  and  $OFFSET=0$ ), or it is provided with appropriate factors or offset.

A transmission of the current to other variables and vice versa is possible too.

## 1.2.17 Water Injection (Component 38)

### 1.2.17.1 Internal Specification of the Mixing Temperature

The injection now allows to enter the desired mixing temperature as specification value TMIX. To do so, the flag FSPEC has to be set to the value 1.

If the inlet temperature is higher than TMIX, the injection mass flow will be calculated in such a way that the desired mixing temperature is achieved.

In the case of lower inlet temperatures, the injection mass flow is set to 0 and the outlet temperature equals the inlet temperature.

In contrast to  $FSPEC = -1$  ("Mixing temperature given externally"),  $FSPEC = 1$  is the more robust variant as when specifying externally, the set temperature will be adhered to in each case, which leads to an error situation when the inlet temperatures are too low. As this situation can also occur temporarily in the course of the iteration, it is recommended not to use  $FSPEC = -1$ .

## 1.2.18 Value Indicator (Component 45)

### 1.2.18.1 Elementary Analysis

Component 45 can now be used for elementary analysis too. To do so, FTYP has to be set to 85, and the desired element has to be set under FSUBST. You can choose between C (FSUBST=21), H (22), O (23), N (24), S (25), Cl (26), Ar (6), ash (27), Ca (43), and Mg (44). The mass fraction for the respective element will be displayed.

### 1.2.18.2 Elementary Mass Flow

Component 45 can now be used for displaying the elementary mass flow too. To do so, FTYP has to be set to 86, and the desired element has to be set under FSUBST. You can choose between C (FSUBST=21), H (22), O (23), N (24), S (25), Cl (26), Ar (6), ash (27), Ca (43), and Mg (44). The mass flow for the respective element will be displayed.

### 1.2.18.3 Composition of the Phases

In the case of a mixture in the multiphase range, the individual phases will usually have different compositions. With FTYP=53 and FTYP=54 respectively, the mass fractions and molar fractions respectively of the individual phases can be output as a result array. Previously, this was only possible for the gaseous (array RAXV) and the liquid phase (array RAXL). In Release 15, all five phases that exist in the TREND library are now supported. For this, there are in addition: RAXSQL (for the solid phase), RAXHYD (for the hydrate phase), and RAXLIQ2 (for a second liquid phase with lower density).

The fraction of the gaseous phase is displayed in the result value RESULT, as previously. The result value RESULT2 contains the total of solid and hydrate phase. The difference of the total from RESULT and RESULT2 to 1 is then the fraction of the liquid phases.

Additionally, there is a result array RAMPHAS that displays the mass fraction for each of the phases.

### 1.2.18.4 Molar Mass Flow

With FTYP=87, the molar mass flow (number of moles flowing through per time unit) can now be displayed. For the molar mass, the same definition as in the case of line value MOLM is used for this, i.e. the constituents given as elementary analysis are also taken into consideration and ash is identified with SiO<sub>2</sub>.

### 1.2.18.5 Reactive and Apparent Power

On electric lines, not only the active power (FTYP = 5) but also the reactive power (FTYP=88) and the apparent power (FTYP=89) are now displayed.

### 1.2.18.6 Salinity

FTYP=22 (mass fraction) and FTYP=43 (emission burden) are now available on brine lines as well. For this, FSUBST has to be set to -1.



## 1.2.19 Measured Value Input (Component 46)

### 1.2.19.1 Molar Mass Flow

Now the molar mass flow can not only be used for the display (see 1.2.18.4) but also as specification. With FTYP=87 in Component 46, the mass flow is set on a line according to the entered number of moles, at the composition that exists on the line in the respective iteration step.

If the molar mass flow is specified on a line whose composition changes over the course of the iteration, this will lead to changes in the mass flow and corresponding consequences for the convergence.

## 1.2.20 Three-Way Valve (Component 49)

### 1.2.20.1 Delayed Start of the Check of the Switching Criterion

As greater fluctuations may occur at the beginning of the iteration, there may be frequent switchover procedures at the beginning of the iteration. To prevent this, it is now possible to specify in specification value FACT from which iteration step onwards the switching criterion is to be checked. Previously this was fixed to the 11<sup>th</sup> iteration step.

### 1.2.20.2 Prevention of Frequent Switchover Processes

To prevent frequent switchover processes during the iteration, there is now a specification value ITSTEP that specifies for how many iteration steps after a switchover procedure no further switchover will be carried out.

## 1.2.21 Gasifier (Components 50 and 96)

### 1.2.21.1 New Result Values

The following result values are now output for the gasifier components:

- RFA: ratio of fuel mass flow to air mass flow
- RFAST: stoichiometric ratio of fuel mass flow to air mass flow (i.e. the ratio required for complete combustion)
- EQRAT: equivalence ratio = RFA / RFAST  
(according to <https://www.sciencedirect.com/topics/engineering/equivalence-ratio>).

An equivalence ratio EQRAT greater than 1 always indicates a fuel excess in the fuel-oxidant mixture, i. e. more fuel than required for a complete combustion (stoichiometric reaction), irrespective of which fuel and which oxidant are used, whereas ratios smaller than 1 indicate a lack of fuel or an equivalent oxidant excess in the mixture.

Please note that the equivalence ratio relates to the air number  $\lambda$  as follows:  $\lambda = 1 / \text{EQRAT}$ .

## 1.2.22 Steam Turbine (Extended) (Component 56)

### 1.2.22.1 Flag FSPECQ for Specifying Power or Mass Flow

As in the case of the simple steam turbine (Component 6), this component now features a flag FSPECQ as well. It serves to set whether the shaft power is to be calculated from the mass flow (FSPECQ=0, default) or the mass flow is to be calculated from the shaft power (FSPECQ=1).

With the mode FSPECQ=1, a feed water pump turbine can be modeled where the shaft power results from the required pump power.

The calculation of the mass flow, however, is extremely disadvantageous for the convergence behavior and should therefore only be used if necessary. In particular, it cannot be used when connecting several turbine disks in series.

## 1.2.23 Governing Stage (Component 58)

### 1.2.23.1 New Result Values

To make the calculation operation and especially the determination of the efficiency from the characteristic line more comprehensible, 12 new result values have been added (bold in what follows).

If the inlet pressure is specified externally (FP=-1) or via a characteristic line (FP=1), Epsilon will calculate the mass flow that would have to flow through the turbine in order to achieve the desired pressure (result value **M1STOD**) backwards from the Stodola law. Comparing with the actual mass flow (line value M1) yields the fraction of the surface area that would have to be opened in order to achieve the desired pressure without throttling (result value AREQ).

The characteristic line CM1 is now used to determine by means of which combination of opened (AOFF), throttled (ATHFF), and closed (ACFF) surface area fractions this AREQ can be achieved. For the x-value AREQ, the y-value AOFF results from the characteristic line, and ATHFF results from the difference to the following y-value. ACFF then results as  $ACFF = 1 - AOFF - ATHFF$ .

This is used to determine which fraction  $MRO = AOFF/AREQ$  can flow through the opened nozzles. At the inlet, this fraction has the unthrottled external pressure (line value P1) and the externally given enthalpy (line value H1), from which the volume flow **VM1O** m<sup>3</sup>/s results. For the efficiency characteristic line, this must be set in relation to the volume flow that flows through these nozzles in the design case, i. e. **VM1NO** = AOFF \* VM1N. Thus the characteristic line is activated with  $x = \frac{VM1O}{VM1NO} = VM1VM1NO$  and yields  $y = \frac{ETAIO}{ETAINO}$ . Thus the result is  $ETAIO = \frac{ETAINO}{VM1VM1NO}$  for the open fraction. With this, the outlet enthalpy **H2O** is calculated for the open fraction.

Because of the throttling, the throttled fraction (MRTH=1-MRO) has a reduced pressure **P1TH** at the inlet, which results from the Stodola law. The enthalpy does not change by this throttling. From this, the volume flow **VM1TH** is calculated, which, for the characteristic line, has to be set in relation to **VM1NTH** = ATHFF\*VM1N. This yields  $x = \frac{VM1TH}{VM1NTH} = VM1VM1NTH$ . The characteristic line then yields  $y = \frac{ETAITH}{ETAINTH}$ , from which  $ETAITH = \frac{ETAINTH}{VM1VM1NTH}$  and the outlet enthalpy **H2TH** result for the throttled fraction.

Multiplied by the respective fractions, the outlet enthalpy of the entire flow (line value) results to  $H2 = MRO * H2O + MTH * H2TH$ . H2 can be used to calculate an effective total efficiency ETAIEFF. This is the actually relevant quantity. The mean efficiency (ETAI=MRO\*ETAIO+MTHR\*ETAITH) output as result value ETAI is irrelevant for the power of the turbine as it does not consider that the efficiency for the throttled fraction only refers to the range P1TH to P2 of the expansion curve. In ETAIEFF, however, it is taken into consideration that the throttled fraction does not perform any work when throttling from P1 to P1TH.

## 1.2.24 General Mixer (Component 60)

### 1.2.24.1 Treatment of the Voltage

Previously, the voltage on the outlet line (U2) was equated with that on the main inlet line (U1) and it was checked whether the voltage on the auxiliary inlet line (U3) was consistent with it.

Due to the expansion of the options in dealing with electric lines in the context of implementing Component 158 (Battery), it turned out that it makes sense to implement different variants for passing on the voltage. The switchover between the variants is effected by means of a flag FSPECU:

- FSPECU = 0: the component does not output any voltage equations, i.e. all three voltages are set externally but have to match
- FSPECU = 1: the voltage is specified on any one pin; the component then transfers it to all other pins
- FSPECU = 2: the voltage at the outlet (U2) is equated with that at the main inlet (U1). U3 is set externally but has to match U1 and U2 respectively. This corresponds to the previous behavior and is the default setting.
- FSPECU = 3: the voltage at the auxiliary inlet (U3) is equated with that at the main inlet (U1). U2 is set externally but has to match U1 and U3 respectively.

## 1.2.25 Feedwater Tank / De-aerator (Extended) (Component 63)

### 1.2.25.1 New Mode for Floating Pressure Operation

Previously, both mode FSPEC=2 and FSPEC=3 were called “floating pressure“. However, the pressure is taken over from the extraction line (Pin 3) only at FSPEC=2, while at FSPEC=3 it is variable indeed but is calculated by the component itself. Therefore a new mode FSPEC=7 has now been implemented; it has the following properties:

- The tank pressure (and thus also the pressure of the feedwater outlet on Line 2) results from the extraction pressure of Pin 3 in all load cases.
- The heating is effected – insofar as it is possible – by the heating water at Pin 7. If this is not possible (no line connected or temperature too low) or insufficient (after the heating water mass flow has reached its maximum value of M7MAX), heating steam from Pin 3 is used for the (remaining) heating.

## 1.2.26 Programmable Component 65 / XUI Interface

The variable for setting/deleting the nominal value writeback flag for specification values has been added.

## 1.2.27 Evaporator with Steam Drum (Component 70)

### 1.2.27.1 New Result Values M2M2N and M3M3N

In this component, result values have been added that indicate the ratio of the prevailing steam production to the steam production in the design case and the ratio of the prevailing exhaust gas mass flow to the exhaust gas mass flow in the design case.

### 1.2.27.2 Pin for Pump Power

There is now an optional Pin 6 where the pump power is output. Depending on the desired level of detail of the modeling, a mechanical shaft, an electric line or a logic line can be connected.

When connecting a shaft, the isentropic efficiency of the circulating pump can be entered in specification value ETAIN and the motor can be modeled separately. Otherwise, the efficiency of the motor should also be considered in ETAIN.

The component only writes the power on Pin 6. When connecting a shaft and electric line respectively, default values are then used for frequency, voltage, current type, and phase (unless the setting of default values on shafts and electric lines has been disabled by the respective model option). If required, however, the desired values can be set by means of a start value (Component 33).

## **1.2.28 CHP Plant (Component 74)**

### **1.2.28.1 New Fluids**

Previously, Heating Cycle 1 could only be operated with water and Heating Cycle 2 only with water, oil, gas or a user-defined fluid. In Release 15, all substances can be used for both heating cycles.

## **1.2.29 Coal Dehumidifier (Component 84)**

### **1.2.29.1 Logic Pin for Residual Moisture**

Previously, the residual moisture of the coal was calculated from specification value XH2OBN and the characteristic line. Now it is possible to control the residual moisture also externally via Logic Pin 7. For this, a flag FXH2OB has been implemented; it has the following settings:

- FXH2OBN=0: use of specification value XH2OBN (as before, default setting)
- FXH2OBN=1: use of the enthalpy of the logic line on Pin 7 as XH2OBN instead of specification value XH2OBN

## **1.2.30 Boiler: Bundle Heating Surface (Component 89)**

### **1.2.30.1 New Result Values**

To make it easier to determine the characteristic line, result values M1M1N and EFFFFN have been added in this component; they show the ratio of the prevailing mass flow and the prevailing effectivity respectively to the corresponding nominal value.

## **1.2.31 Reformer / Shift Reactor (Component 95)**

### **1.2.31.1 Flexible Treatment of Errors**

In this component, a flag FERR has been implemented that allows the user to select how the component is to react if a desired specification for the material concentration cannot be adhered to:

- 0: no message
- 1: comment
- 2: warning
- 3: error (this is the default setting)

Previously, an error was output at all times and the respective material concentration was set to 0. Now the material concentration is set to the value that is only just possible instead (this can be the minimum or maximum value, depending on the case).

When specifying values of the water gas reaction (CO, CO<sub>2</sub>, H<sub>2</sub>, and H<sub>2</sub>O), however, it may happen that there is no solution, e.g. if the inlet flows contain more C than O. Previously, the material balance was violated in this case. Now the material balance is closed by elemental C and free O<sub>2</sub> respectively.

A message on this is output that lies one level below the message level set in FERR, i.e. by default a warning.

## **1.2.32 Coupled Rectifier Column (Component 104)**

### **1.2.32.1 Kernel Expression for the Extractor Temperature**

Previously, the extractor temperature could only be specified in a fixed way in specification value T3. Now it is possible to have it calculated via a Kernel expression ET3. Flag FT3 has to be used to set whether T3 or ET3 is to be used.

## **1.2.33 Direct Storage / Mass Storage (Component 118)**

### **1.2.33.1 Logic Pin for Mass Flow Request**

Previously, both the charging and the discharging mass flow were specified externally for this component. If these specifications could not be fulfilled due to the prevailing storage condition, an error message was output.

Now it is possible to specify the desired mass flow from the storage system on Logic Pin 5. If this value is negative, it refers (in terms of the amount) to the mass flow into the storage system. The flag FM serves to activate the specification:

- FM = -1: both mass flows are given externally
- FM = 1: both mass flows are set according to the desired value on Pin 5 and the filling level

At FM=-1, the storage system can be charged and discharged simultaneously. At FM=1, the discharging mass flow is set to 0 when charging and the charging mass flow is set to 0 when discharging.

### **1.2.33.2 Mass Flow Reduction when Reaching the Filling Level Limit**

When specifying the desired value for the mass flow (FM=1), the flag FTIMELIM allows to adjust what is to happen with a time series calculation if the storage system reaches the lower or upper filling level limit:

- At FTIMELIM=0, the mass flow is reduced so far that the mass flow can remain constant during the entire time interval.
- At FTIMELIM=1, the desired mass flow is fully supplied and it is calculated how long this kind of operation is possible. The time interval is then split accordingly. Also, this mode is always active when the mass flows are specified externally.

## **1.2.34 Diesel/Gas Motor (Reciprocating Engine) (Component 125)**

### **1.2.34.1 New Fluids**

More fluid types can now be used for the high-pressure cooling (Pins 5 and 6) and the lubrication (Pins 8 and 9).

## **1.2.35 Stratified Storage (Component 145)**

### **1.2.35.1 Implementation and Monitoring of the Dimensionless Diffusion Number**

In Component 145, the precision and the stability of the numerical solution depend on the number NFLOW of grid points in the height as well as on the width of the time steps (e.g. in the time series). For this, a dimensionless number, the diffusion criterion RDIFNUMB, has been implemented in Release 15. For a robust numerical solution, the respective parameters, NFLOW and time step, have to be selected by the user in such a way that the result value RDIFNUMB remains  $< 0.8$ . If the value of RDIFNUMB exceeds the limit of 0.8, the user will be warned with an error message.

## **1.2.36 Components 7 (Steam Turbine Condenser), 9 (Feedwater Tank), 10 (Feedwater Preheater), 20 (Drum)**

### **1.2.36.1 Expansion by Transient Modeling**

Components 7, 9, 10, and 20 have been expanded in Release 15 and enable the modeling of the transient processes. Components 9 and 20 use algorithms already implemented in Component 119 for this. The heat exchangers Component 7 and 10 use the combined physical and numerical calculation model of Component 126. In all mentioned components, a flag FINST is installed that is used for switching over between the steady-state (as before) and the transient calculation. The transient modeling requires knowledge about some geometrical details of the components. For further details please refer to the EBSILON Online Help.

## **1.2.37 Component 124 (Heat Exchanger with Phase Transition)**

### **1.2.37.1 Implementation of a Numerical Algorithm**

Component 124 has been expanded by a new mode of calculation. This mode can be activated by means of the new flag FALG. At FALG=0, Component 124 calculates the heat exchange based on an analytical solution, as before, where the K-number is calculated from the individual alpha numbers. At FALG=1, by contrast, the heat exchanger is numerically solved in the direction of the flow direction of the working fluids. The number of elements is set by means of the parameter NFLOW. Then a numerical solution of the heat exchange is sought, in analogy to Component 126. To do so, the combined analytical and numerical procedure from Component 126 is applied. The difference to Component 126 consists in the fact that Component 124 only determines a steady-state solution. The result of the calculation depends on the number of the numerical elements (NFLOW) and the selection of the numerical scheme (FNUMSC). Compared to the analytical solution, the numerical solution has the advantage that it yields more precise solutions especially in the case of a strong non-linear dependency between the enthalpy and the temperature of the fluid (curvature of the lines in the QT diagram).

### **1.2.37.2 Pinchpoint Violations**

In reality, heat can only be transferred from a warmer to a colder medium. This must be fulfilled not only at the endpoints, but also in each intermediate point of the heat exchanger.

The numeric solution automatically meets these conditions because the intermediate points are calculated one after the other. However, the requirement is that the number of intermediate points is large enough so that the specific heat within a cell can be considered constant. But in an analytical solution that looks only

at the endpoints, there may be places in the heat exchanger where this condition is violated (“pinchpoint violation”), namely when the specific heat is not constant (curved lines in the QT diagram), and especially in phase transitions. Therefore, a separate examination for pinchpoint violations must be carried out in the analytical solution after the calculation.

In a design calculation, however, it can happen that temperature specifications are set in such a way that they are not reachable without a pinchpoint violation. In this case, of course, the numerical method cannot find a solution and a corresponding error message is issued.

### 1.2.37.3 Convergence Tuning

Unfortunately, the convergence of the numerical solution cannot be guaranteed in all cases. In this case, it is recommended to change the number of points NFLOW or the numeric schema FNUMSC. The calculation with central differences (FNUMSC=1) is more accurate and therefore also the default setting. However, it can lead to numerical problems, especially in the case of widely differing mass flows. In this case, a switch to the upwind method (FNUMSC=0) is recommended.

### 1.2.37.4 Heat Transfer Coefficients

Component 124 offers the possibility to specify different alpha numbers and exponents for the individual zones. With the numerical solution, you can either work with constant or range-dependent alpha numbers. For this reason, the FTYPHX switch now distinguishes between

- FTYPHX = 0: “General heat exchanger with constant alpha numbers”  
In all ranges, AL12N and EX12 and AL34N and EX34 are used
- FTYPHX = 4: “General heat exchanger with area-specific alpha numbers:

Depending on the phase AL12ECON, AL12EVAN or AL12SUPN and EX12ECO, EX12EVA and EX12SUP, and AL34ECON, AL34EVAN or AL34SUPN and EX34ECO, EX34EVA or EX34SUP are used.

The other FTYPHX settings (1 (Eco), 2 (evaporator) and 3 (superheater)) already used area-specific alpha numbers in the analytical solution. In the numerical algorithm, the calculation is performed in the same way as for FTYPHX=4.

You can find meaningful values for alpha numbers for the different fluid combinations in the standard datasets of component 61.

### 1.2.37.5 Result value KA

When calculating the result value KA, there are two options for the numerical solution:

- Calculation from the sub-results of the individual zones
- Calculation from the total result

Especially with small terminal temperatures, there may be significant differences between the two values. We have therefore decided to display both values: the first as the “physical” value KAPH, for the second one the existing result value KA is used. This has the advantage that the Fourier law  $QT=KA*DTM$  also applies to the numerical solution. In this case, “K” is the “effective” K-value, which would lead to the corresponding heat

## 1.2.38 Component 126 (Transient Heat Exchanger)

### 1.2.38.1 Implementation of an Effectivity Factor and Identification Mode

Component 126 has a new specification value and result value – EFFN and EFF. These represent an effectivity factor. An effectivity factor allows to additionally modify the heat transmission (e.g. at  $EFF < 1$  the heat transmission is deteriorated) and e.g. adjust it to measured data. The effectivity factor is the ratio of the actual heat transmission coefficient  $K$  to the heat transmission coefficient that results from the alpha and lambda numbers specified in the component.

From Release 15 on, Component 126 can be switched to identification mode. There is a flag FIDENT for this purpose. In this mode, the heat absorption on the fluid side 1-2 (the inner pipe) is specified by specification of the temperatures  $T1$ ,  $T2$  or enthalpies  $H1$ ,  $H2$ . For FIDENT=0 (no identification), the calculation is carried out with the specified effectivity factor EFFN. For FIDENT=1 (identification), however, the user is expected to define the two enthalpies  $H1$ ,  $H2$  outside the component, and Component 126 determines the corresponding effectivity factor EFF.

### 1.2.38.2 Reference of the Heat Exchange Surface Area to Different Pipe Diameters

Up to Release 15, the heat exchange surface area of the bridge wall always referred to the inner pipe diameter. From Release 15 on, Component 126 has a new specification value FRELDIA. At FRELDIA=0, the heat exchange surface area refers to the inner diameter of the pipe, as before. At FRELDIA=1, the heat exchange surface area refers to the outer diameter of the pipe.

### 1.2.38.3 Specification of the Material Properties of the Pipe Walls

From Release 15 on, you can specify the properties (density, thermal conductivity, thermal capacity) for known materials by means of the flags FMATBW (bridge wall) and FMATOW (outer wall) in Component 126. Currently, a library of 68 materials is available. In addition, the user can specify own polynomials for the material data in the Kernel expressions ERHOBW, ELAMBW, ECPBW (density, thermal conductivity, thermal capacity of the bridge wall) and ERHOOW, ELAMOW, ECPOW (density, thermal conductivity, thermal capacity of the outer wall).

### 1.2.38.4 Implementation of the Numerical Steady-State Solution

From Release 15 on, Component 126 can calculate the heat exchange in steady state – e.g. if FINST=1 or in the case of a transient calculation under steady state conditions (that do not change over time – not only based on an analytical solution as before. With the new flag FALG, you can search for a numerical solution also in steady state at FALG=1. At FALG=0, Component 126 calculates the heat exchange based on an analytical solution as before, where the  $K$  number is calculated from the individual alpha and lambda numbers. The result of the numerical solution (at FALG=1 also in steady state!) depends on the number of the numerical elements (NFLOW).

In addition, a flag FNUMSC has been implemented in Component 126. With this flag, you can select the scheme for the numerical solution: FNUMSC=0 corresponds to the upwind scheme (highest stability) and FNUMSC=1 corresponds to the method of central differences (highest precision). It is recommended to calculate with FNUMSC=1 at all times. Only in the event of a lack of convergence can FNUMSC=0 be applied instead in order to stabilize the calculation.



### 1.2.38.5 Option to Calculate the Heat Flow Reversal. Specification Value FREV

From Release 15 on, Component 126 can calculate the heat flow from Flow 12 (previously always the “cold” side) to Flow 34 (previously always the “hot” side). As this calculation involves some requirements to the convergence of the model, the user has to explicitly enable this calculation by means of a flag FREV. If FREV=0, the heat flow remains as before (from Flow 34 to Flow 12). For FREV=1, the fluid in Flow 12 can be hotter than the fluid in Flow 34 and the heat flow direction will change accordingly.

## 1.2.39 Component 119 (Indirect Storage)

### 1.2.39.1 Specification of the Material Properties of the Storage Wall

From Release 15 on, you can specify the properties (density, thermal conductivity, thermal capacity) for known materials for the storage wall by means of the flag FMAT in Component 119. Currently, a library of 68 materials is available. In addition, the user can specify own polynomials for the material data in the Kernel expressions ERHO, ELAM, ECP (density, thermal conductivity, thermal capacity of the storage wall).

## 1.2.40 Components 148, 149, 150 (Header)

### 1.2.40.1 Expansion by Further Fluids

From Release 15 on, Components 148, 149, 150 (Header) can be calculated with further line types (fluids):

- flue gas
- air
- brine
- oil
- gas
- crude gas
- user-defined

## 1.2.41 Component 131 (Transient Separator)

### 1.2.41.1 New Specification Values

From Release 15 on, Component 131 has four new specification values:

- FOUTUS\_M – use of outlet, mass flow
- FOUTUS\_H – use of outlet, enthalpy
- FOUTUS\_P – use of outlet, pressure
- FDT – direction of transfer

The three flags FOUTUS\_X can take two values. At FOUTUS\_X=0, the calculated value (M, H, P) on Pin 2 from the current time step will be used, as before. This enables the highest precision. At FOUTUS\_X=1, in contrast, the respective value from the last time step will be used. This has the advantage that this value does not change during the current time step, which leads to a better convergence of the calculation. However, the calculation will become less precise here.

The flag FDT controls the direction of transfer. For FDT=0, the value is transferred from Pin 1 to Pin 2. For FDT=1, the value is transferred from Pin 2 to Pin 1.

## 1.3 Physical Properties

### 1.3.1 New Function Calls

In analogy to the existing function `FuncXCOMP_OF_IND` (mass fraction for substance), there are now also the functions:

- `FuncXCOMP_MOLAR_OF_IND`: molar fraction for substance
- `FuncXCOMP_NO_ELEM_OF_IND`: mass fraction for substance, related only to the non-elementary fractions
- `FuncXCOMP_MOLAR_NO_ELEM_OF_IND`: molar fraction for substance, related only to the non-elementary fractions
- `FuncXCOMP_NO_ELEM_NO_H2O_OF_IND`: mass fraction for substance, related only to the non-elementary and non-water (for FDBR only) fractions
- `FuncXCOMP_MOLAR_NO_ELEM_NO_H2O_OF_IND`: molar fraction for substance, related only to the non-elementary and non-water (for FDBR only) fractions

Moreover, there is the function `FuncMOLARMASS_OF_IND` that returns the molar mass for a substance. For usage in EbsScript (resp. EbsOpen) the argument for the substance-index has to be taken from the enumeration "UniversalSubstanceEnum" (resp. "EpUniversalSubstance").

PLEASE NOTE: this call is only defined for substances which exist in the library that is used. For other substances, the return value is NOT DEFINED, i.e. no assumptions may be made about it at all.

### 1.3.2 TREND: Support for Sea Water

In the TREND library, there is a new module for calculating sea water. To use the sea water module the substance "water" must be present and the "salinity" must be set in the attributes. Please refer to the TREND-manual for the range of validity of the sea water module.

## 2 User Interface

### 2.1 H-X Diagram for Humid Air

From Release 15 on, an H-X diagram can be created and displayed in connection with the line type "Humid Air". The activation can be found under Extras – Diagrams – h-x diagram (humid air).

### 2.2 Component Matrices

#### 2.2.1 3D Representation

In addition to the previous 2D color representation of the component matrices (specification value matrices, result matrices), from Release 15 on there is now a 3D representation too. You can choose between 2D and 3D representation of the respective matrix in a dropdown box on the matrix page of the component.

### 2.3 Command Line Options

ebs2000.exe can now be started with the options "-shownormal", "-showminimized", and "-showmaximized" for selecting the position of the main window.

### 2.4 Multi-Simulation Bar

A dropdown box "Accept calculated global design values" has been added for displaying message boxes / dialogs for the treatment of the calculated nominal values during the multi-simulation.

Please note: these dropdown boxes change the same global settings as the corresponding dropdown boxes in the General Options.

### 2.5 General Option

#### 2.5.1 Loading the Common Language Runtime

A General Option for loading the Common Language Runtime (CLR) at the start of Epsilon has been added. It allows to prevent crashes if EbsScripts (in the debugger) load the CLR due to user-defined code and DLLs respectively.

### 2.6 Macro Objects

#### 2.6.1 Grouping Specification and Result Values

In the macro interface, a grouping of the values like in the case of the standard components can be created on the pages "Specification values" and "Results" respectively in the column "Grouping".

A group is started with the scheme

```
[Name  
and ended with  
]
```

Each occurrence of the characters "[\];\" in the name must be protected by a leading backslash "\".

Each group can optionally be allocated a category and an initial state too. Then the syntax is:

```
[Name;category:category;default_state:state
```

Possible categories are:

- unused
- specificationvalue
- designonlyspecificationvalue
- offdesignonlyspecificationvalue
- nominalvalue
- ebsscript

Possible initial states are:

- open
- closed

A group begins in the line in which it was started and ends with the line in which it was finished. Groups can be nested, and several groups can begin and end in one line.

## 2.7 Value Crosses

### 2.7.1 Alternative Data Connection by EbsScript Expression

In the properties tab "Basic Properties", three types of data connection can now be defined:

- "by connector": the data object is defined by connection line (behavior as before in Ebsilon)
- "by expression": the data object is defined by an EbsScript expression of the type "ebsobject" (e.g. "::STEAM\_35")
- "automatically": if the value cross is connected to an object via the connection line, this object will be used. Otherwise the latest connected object will be memorized and used.

Among other things, the data connections "automatically" and "by connector" enable value crosses that display data of objects inside of macros.

Moreover, a text that is located at the end of the connection line can be selected and specified respectively by means of "Connector text".

### 2.7.2 Defining Values by EbsScript Expressions

Alternatively to a specification/result value, you can now also specify an EbsScript expression in the properties tab "Values" for "Name" at "Display data to selected identifiers". To do so, the setting "<expression>" must be selected from the list first (first entry). Subsequently, the expression must be entered in the column "Unit/Expression".

In the expression, reference is made to the connected object with "\$". Reference to the value cross itself can be made with "\$\$".

## 2.8 Filtering Dropdown Boxes

The dropdown boxes for selecting e.g. specification values, units, etc. now allow to filter the list by entering a text in the input field of the dropdown box.

When a text is entered, only those entries that contain this text will be displayed in the list. If the entered text contains at least one uppercase letter, case-sensitivity applies, otherwise it does not.

The following has to be considered for dropdown boxes for selecting specification values:

The entries "Like in parent profile" and "Expression..." cannot be selected by filtering. Instead, "Expression..." can be activated by Ctrl+X (for 'expression') and "Like in parent profile" by Ctrl+R (for 'parent').

## 2.9 Display of Specification Values

When a specification value was selected in a sub-profile, it used to be impossible to differentiate whether it had been set in this profile or inherited from a superordinate profile.

Now selected specification values that show inherited values are highlighted in grey.

## 2.10 Time Series Dialog

### 2.10.1 Floating-Point Numbers Are Stored with Full Precision

Previously, floating-point numbers were stored as text. From now on, floating-point numbers are stored with full precision. Displayed in the time series, the numbers may be shown abbreviated. When editing and copying, however, the full precision will be used.

This allows to precisely recalculate individual points in time except the "numerical noise".

### 2.10.2 Pasting Data via the Clipboard

In the time series dialog, there is the new feature "Insert copied cells" in the context menu of the head of the line and column respectively. It inserts the data of the clipboard into the time series at the selected position as new lines and columns respectively.

### 2.10.3 New Commands in the Context Menu of the Head of the Line

Moreover, the context menu of the head of the line now also contains the different commands for starting the calculation.

## 2.11 Number of Characters in Text Fields

The maximum number of characters in text fields has been enlarged:

- Single-line text fields can now contain up to 2147483646 ( $2^{31} - 2$ ) characters.
- Multi-line text fields can now contain up to 4294967295 ( $2^{32} - 1$ ) characters.

## 3 Additional Modules

### 3.1 EbsScript

#### 3.1.1 New Functions

##### 3.1.1.1 General Functions

Functions for checking and taking over nominal values:

- hasNominalValues:Boolean : informs whether new nominal values are available
- takeOverNominalValues:Boolean : takes over nominal values into the components; return value is True if the nominal values have been taken over successfully

##### 3.1.1.2 Functions in the Interface Unit System

New functions have been added in the Interface Unit System. The function `simulate_trans(timeStep:Real; startTime:Real = 0)` allows to carry out a transient simulation with the specified time step in seconds and (optionally) with the specified initial point in time. The function `simulate_trans_init(startTime:Real = 0)` enables an initialization simulation (in analogy to the first step in the time series) with the (optionally) specified time step. Both functions in combination with the previously existing function `tsCopyDependent-Values` can be carried out alternatively to a time series.

The functions:

- function `writeToFile(path:String; text:String; append:Boolean; encoding:textEncodingEnum = textEncodingASCII) : Boolean; internal;`
  - function `readLineFromFile(path:String; var text:String; lineNumber:Integer; encoding:textEncodingEnum = textEncodingASCII) : Boolean; internal;`
- are now defined in the Unit System. Both got a new parameter „encoding“ to control the encoding of the file. The enumeration `textEncodingEnum` has the has following entries:
- `textEncodingASCII`: only ASCII characters are written (old mode); Multi-Byte-characters are discarded; Please note: when reading, Unicode Byte-Order-Marks are not evaluated and do not change the reading mode
  - `textEncodingUTF8`: UTF-8-encoding with Byte-Order-Mark (BOM); when reading, checks for BOM and if present adjusts the reading mode respectively
  - `textEncodingUTF16LE`: UTF-16-little-endian-encoding with Byte-Order-Mark (BOM); when reading, checks for BOM and if present adjusts the reading mode respectively

Additionally, there is a new function to read all lines or a range of lines in a single call

- function `readLinesFromFile(path:String; var lines:array of String; first_line:Integer = 1; last_line:Integer = -1; encoding:textEncodingEnum = textEncodingASCII) : Boolean; internal;`

All lines starting from `first_line` to `last_line` (inclusive) will be read into lines. If `last_line` is -1 then all lines starting from `first_line` will be read into lines.

#### 3.1.2 Logfile Support for the EbsScript.Python Interface

In the interface unit Python, new functions for logging the Python calls have been added (see EbsScript.Python-Unit).

To activate the logfile support, `ebsscript_python_interpreter.py` must be registered again (To do so, please execute the `ebsscript_python_interpreter.py` script in Python again)!

### 3.1.3 New Properties for the Type “`ebsComp`”

The type “`ebsComp`” has the following new properties:

- `calculationmode`
- `calculationkernelscript`
- `calculationkernelscriptauxiliarycalls`
- `calculationkernelscriptgeneratedefaultequations`
- `calculationfprog`
- `calculationsequence`

These correspond to the settings in the properties tab “Experts” of the component.

### 3.1.4 New Properties for Model/Profile Variables

For scalars and arrays:

- `name`, `description`, `description2`
- `datastringified`: content represented as text; read/write

Only for arrays:

- `low`, `high`, `size`: all read-only
- `toarray`, `fromarray`: converting to/from dynamic EbsScript array of the respective type
- `resize`: change the size of the array; minimal index (lower-bound) is retained, data are retained or standard-initialized in the case of new elements.
- `rebase`: minimal index (lower-bound) is changed; data are retained; `array_before_rebase[old_lower_bound + index] == array_after_rebase[new_lower_bound + index]` (for `index = 0 .. size-1`)
- `redim`: minimal and maximal index (lower-bound and upper-bound) are changed  
`array_before_redim[index] == array_after_redim[index]` (for all index that are valid before AND after `redim`)  
New elements are standard-initialized.

### 3.1.5 Expansion of the Formatting for `printToString`

In the case of the function `printToString`, the output can now be formatted with a String argument:

After the argument, a colon “:” follows and then an EbsScript String that contains the formatting. It must be in “printf” format (see e.g. <https://docs.microsoft.com/en-us/cpp/c-runtime-library/format-specification-syntax-printf-and-wprintf-functions>), where the leading “%” must be omitted. Furthermore, printf-“\*” arguments are not supported; however, these can be replaced by a corresponding pre-editing of the format string.

Example:

```
var s,f:string;
    r:real;
    i:integer;
begin
    i:=42;
```

```

s:=printToString(i:"08d");// print 8-digits with leading '0's
println(s);

r:=12.345678e90;
f:=".10g";
s:=printToString(r:f);
// print r with at most 10 significant digits in f or e format,
// whichever is more compact for the given value and precision
println(s);

end.

```

Output:

```

00000042
1.2345678e+91

```

Please note: Previously, the formatting with a String argument could only be used for the functions print and println.

### 3.1.6 Unit Tests

EbsScript units can now be furnished with a test routine. To do so, the key word "unittest" must be located before the final "end". Then comes the analogue to a program block. The unit test is carried out by starting the unit.

A sample unit with included unit test can be inserted via the window "Keywords" "Controls → unit-sample".

### 3.1.7 Expanded Topological Access to Components and Lines

In addition to the previously existing accesses by means of `._[pin number]`, `__[pin number]`, `._in` and `._out` there are the following expansions:

Access from a component to a component/line that is connected via a line.

- `._[pin number]` : Access to the next component (NO line) that is connected to the pin "pin number" via a line. For instance, `turbine.__6` will return the component connected to the shaft outlet of the turbine.
- `._[pin number]d` : Access to the next component or line that is connected to the pin "pin number" via a line. For instance, `transmitter.__1d` will return the object connected to the inlet of the value transmitter.
- `._[pin number]p` : Access to the next line (NO component) that is connected to the pin "pin number" via a line. For instance, `controller.__3p` will return the line which is connected to line at the "corrected value" outlet of the controller.

In analogy, there are for the line:

- `._in` and `._out` : access to the next component (NO line) that is connected to the inlet, respectively outlet, of the line. For instance, `water._in` will return the component connected to the inlet of the water line.
- `._ind` and `._outd` : access to the next component or line that is connected to the inlet, respectively outlet, of the line. For instance, `logic._ind` will return the object connected to the inlet of the logic line.
- `._inp` and `._outp` : access to the next line (NO component) that is connected to the inlet, respectively outlet, of the line. For instance, `scheduled._inp` will return the line which is connected to the scheduled value line.



## 4 Changes in the Results

### 4.1 Component 11 (Generator)

In Component 11 (Generator), the result value RGENFREQ was unitless. From Release 15 on, this value has the unit 'frequency' – EBSILON default unit 1/min. For having the frequency displayed in Hz, the user can switch over the display unit.

### 4.2 Component 119 (Indirect Storage)

In Component 119 (Indirect storage), the non-constant (temperature-dependent) values of the thermal conductivity, density, and thermal capacity of the storage wall material remained unconsidered in the case of the setting FALGINST=4 (model using combined numerical and analytical methods) up to Release 15. This concerns the settings FDATA<>1. Therefore the calculated results change in models where the following applied: FALGINST=4 and FDATA<>1.

### 4.3 Component 126 (Transient Heat Exchanger)

In Component 126 (Transient heat exchanger), the results were calculated without considering the parameter LAMADJ (multiplication factor to 1/LAMBDA – the walls' heat conductivity resistance) in the case of the setting FALGINST=4 (model using combined numerical and analytical methods) up to Release 15. In the case LAMADJ=1, this is irrelevant. For the other cases, however, (for instance, up to Patch 3 Release 14 the default value of LAMADJ equalled 0; this corresponds to a total neglect of the resistance due to the thermal conductivity of the walls) the value of LAMADJ remained unconsidered in the result values. From Release 15 on, the parameter LAMADJ is consistently considered. Therefore the calculated results change in models where the following applied: FALGINST=4 and LAMADJ <> 1.

Moreover, the non-constant (temperature-dependent) values of the thermal conductivity, density, and thermal capacity of the material of the wall remained unconsidered in the case of the setting FALGINST=4 (model using combined numerical and analytical methods) up to Release 15. This concerns the settings FDATA BW<>1, FDATA OW<>1. Therefore the calculated results change in models where the following applied: FALGINST=4 and FDATA BW<>1, FDATA OW<>1.

### 4.4 Default Value for the Reference Pressure

The default value for the reference pressure has been made more precise, from 1.013 to 1.01325 bar. This default value is used if no measuring point for the reference pressure exists in a model, which should actually not be the case.

In such models, the pressures are now increased by 0.00025 bar on lines with a relative pressure specification. This has effects on the results for the exergy as well.

To make the model yield the same values as before, you can easily insert a measuring point with a reference pressure of 1.013 bar.

### 4.5 Minimum Mass Flow for the Controller

The controllers in Ebsilon work multiplicatively, i.e. the actuating variable is provided with a certain factor from one step of the iteration to the next. At a start value of 0, the actuating variable will therefore remain at 0. If it happened during the iteration that the actuating value had decreased to very small values, it could move away from there only very slowly. Then it could happen that the global convergence criterion

was achieved while the controller was still stuck in this range and could not reach its target value. To prevent this, a suitable minimum mass flow can be entered at L2MIN and L3MIN respectively.

To make it easier for the controller to move away from extremely small values, the upper limit CHL2 and CHL3 respectively for the change factor is now no longer applied in the case of an increase in the actuating variable if the actuating variable is less than or equal to  $10^{-5}$ . Therefore smaller values can now also increase by more than e.g. 15 percent from one step of the iteration to the next. If the actuating variable decreases, the limitation will be used as previously in any case in order to delay a decrease into deep ranges.

In the case of models where the control is effected in the range of actuating variables below  $10^{-5}$ , however, the control behavior may change due to this. This specifically applies to models where a start value of 0 was entered for switching off the controller. It would be better to use the flag FFU for the purpose of switching off.

## 5 Known Errors

### 5.1 Component 124 – Numerical Solution

There is a large number of variants in this component, both in terms of the choice of materials on both sides and the temperature and mass flow ratios. Unfortunately, we cannot rule out the possibility that certain combinations still have numerical problems and you get corresponding error messages. It would be good if you would contact the hotline in this case so that we can investigate the case and, if necessary, implement appropriate tuning measures in the next patch to get the case up and running.

For time constraints, we have not yet managed to implement access to the result fields of the numerical calculation in the QT diagrams. We'll do this in the next patch. The gradient within the part is still analytical in the QT diagram, so there may be discrepancies between the displayed and the calculated gradient.

As a workaround, you can also use Excel to create the QT chart from the component result arrays. As an example, you can use the file `Data/Examples/Components/Component_124_QT_for_numerical_solution.xlsx`.

### 5.2 Documentation

Unfortunately, it has not been possible to include all of the new features of Release 15 into the Help. Please refer to these Release Notes until a patch for the Help is available.

It is also possible to access the latest status of the Online Help available on the Internet by shifting the Help Preferences to "Launch in Browser".

## **STEAG Energy Services GmbH**

Wetzbach 35

64673 Zwingenberg

Germany

Telefon: +49 6251 1059-0

Telefax: +49 6251 1059-29

[info@ebsilon.com](mailto:info@ebsilon.com)

[www.ebsilon.com](http://www.ebsilon.com)

[www.steag-systemtechnologies.com](http://www.steag-systemtechnologies.com)

**steag**