

# Configuration Dialog Box (D2000/Mathematical Data Processing/Eval Tags)

## Eval tags - configuration dialog box

Editing of all objects in the process [D2000 CNF](#) is being performed in the [configuration dialog box](#), a specific part of which is common for all editable objects and another part depends on the type of edited object.

The configuration dialog box of eval tags consists of several parts (tabs) that contain similar parameters.

[General properties](#)  
[Groups](#)  
[Process alarms](#)  
[Calculation](#)

### General properties

---

#### Description

A text string describing the eval tag. Maximum: 128 characters.  
Possibility to use the [Dictionary](#) (to open press **CTRL+L**).

#### Status Text

Defines a [status text](#) for the eval tag. The status text allows redefining values' identifications of the eval tag.

#### Transformation palette

Selection of an index to transformation palette. See the topic [Transformation palette](#).

#### Value type

Value type selection of the eval tag. Admissible value types are listed in the following table.

Identification	Eval tag value type
De	Logical value
Ce	Integer value
Ae	Analog value
TmA	Absolute Time
TmR	Time interval - relative time

#### Technical units

Technical units of the eval tag. Maximum: 12 characters. Possibility to use the [Dictionary](#) (to open press **CTRL+L**).

#### Limits

Technological limits may be defined for eval tags of the Ce - integer, Ae - analog and TmR - time interval type. There are four limits: VHL, HL, LL and VLL. The limit may be specified either directly - by entering a value into the input field, or its value may be defined by a system object (dynamic limit) - the button placed right from the input field.

VHL	Very High Limit - the highest limit
HL	High Limit - high limit
LL	Low Limit - low limit
VLL	Very Low Limit - the lowest limit

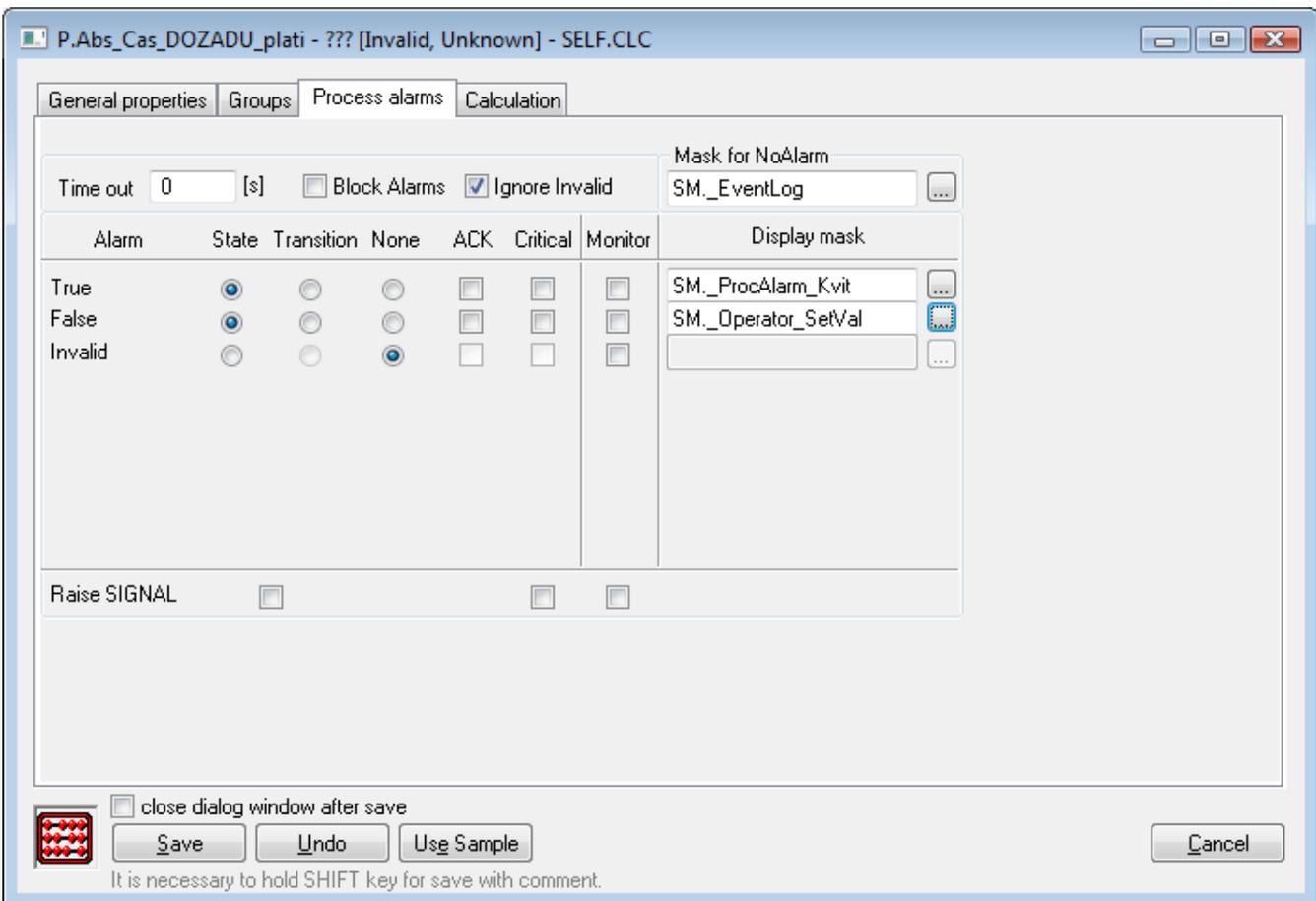
Values of individual limits determine the state of the eval tag according to the value. The relation eval tag value - limits gives 6 possible states.

Limit	Object state according to relation <i>Value- Limit</i>
	<b>Above VHL</b> (object value > VHL)

VHL	
	<b>Above HL</b> (HL < object value < VHL)
HL	
	<b>Normal</b> (LL < object value < HL)
LL	
	<b>Bellow LL</b> (VLL < object value < LL)
VLL	
	<b>Bellow VLL</b> (object value < VLL)

As the limits can be dynamic (determined by the object value), a situation may occur, when the condition  $VLL < LL < HL < VHL$  is not met (the limits crossing). The eval tag is then in the state **Limit Problem**.

## Process alarms



P.Abs\_Cas\_DOZADU\_plati - ??? [Invalid, Unknown] - SELF.CLC

General properties Groups **Process alarms** Calculation

Time out 0 [s]  Block Alarms  Ignore Invalid Mask for NoAlarm SM\_EventLog

Alarm	State	Transition	None	ACK	Critical	Monitor	Display mask
True	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SM_ProcAlarm_Kvit
False	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SM_Operator_SetVal
Invalid	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Raise SIGNAL

close dialog window after save

Save Undo Use Sample Cancel

It is necessary to hold SHIFT key for save with comment.

### Timeout

Time delay (in seconds) for the evaluation of process alarm. Process alarm is raised, if a cause of raising the alarm is valid at least within this period.  
**Note:** the time delay is applied only to status alarms - transition alarms occur immediately when the value is changed.

### Block alarms

Enabled **Block Alarm** option blocks the evaluation of process alarms.

### Ignore invalid

If the option is checked, it will remove invalid values when evaluating one of the defined process alarms.

Example: value change from S1 to S2 is executed as follows: S1 --> invalid --> S2.

If the parameter is checked, the alarm evaluation will be: S1--> S2.

If the parameter is not checked, the alarm evaluation will be: S1 --> invalid --> S2.

## Mask for NoAlarm

Selection of a [display mask](#), that will be used in process [D2000 HI](#) (the [Alarm list](#) - the **Event description** column) after changing the status of one of the defined process alarms into the Normal state.

Individual items allowing the settings of parameters of process alarms are ordered in a table. The table contains the following columns:

- **Alarm** - the column contains all possible [causes to raise process alarm](#) for the object of Eval tag type.
- **State** - the radio buttons allow selecting just one of the **State**, **Transition** or **None** options. Enabled **State** option means, that process alarm starts when the value of the I/O tag is, at least within the period defined by the **Timeout** parameter, in the particular state (the name of a particular state is placed in the particular row, in the **Alarm** column - *pa\_ValueStateName*).
- **Transition** - checked **Transition** option means, that the process alarm starts when the I/O tag value will pass to a particular state (the name of a particular state is placed in the particular row, in **Alarm** column - *pa\_ValueStateName*).
- **None** - if checked, process alarm is not evaluated.
- **ACK** - checked **ACK** option means, that the process alarm must be acknowledged by the operator in the process [D2000 HI](#). "Transition" process alarm must be always acknowledged.
- **Critical** - flag of critical process alarm.
- **Monitor** - the option will write a particular change of an I/O tag value state into the log database as a spontaneous value change.
- **Display mask** - selection of a [display mask](#), that will be used in process [D2000 HI](#) (the [Alarm list](#) window - the **Event description** column) after changing the status of one of the defined process alarms into a particular state.

## Raise SIGNAL

**Raise signal** parameter contains three checkboxes with the following function:

- **the first checkbox** - is placed below the **State** and **Transition** columns. Enabling the option, the system generates the signal, if any of configured process alarms occur.
- **the second checkbox** - is placed below the **Critical** column. Enabling the option, the system generates the signal, if any of configured critical process alarms occur.
- **the third checkbox** - is placed below the **Monitor** column. If checked, the system will generate the signal, if any logged changes of defined process alarms are written into the log database.

Generating the signal means, that the system tag **Signal\_Trigger** is set to a TRUE value.

## Calculation

---

### Expression

The arithmetical expression determines the eval tag value. The expression may contain [extended syntax](#), could be written in several lines, and may contain a comment. The comment begins with the character ';' and finishes at the line end. For example:

```
M.1 + ; description
2
```

**Note:** Typing the character '^' after the name of an object of the [Structured variable](#) shows the list of all structure columns.

For more information on expressions see the chapter [Mathematical expressions in eval tags and events](#).

### Replace Invalid values with 0

If checked, all invalid values of the objects defined in the expression will be replaced with the value of 0. The feature can be used to prevent the expression from getting invalid values.

There are converted just the values of input objects, invalid values of intermediate data are not converted. Values of inputs objects are converted as follows:

- Integer --> 0
- Real --> 0.0
- Relative time --> 0.0
- Boolean --> False

Other value types are not converted.

## Objects

Buttons to select an object of the D2000 system. The selected object is inserted into the expression on the current cursor position.

## Constants

Buttons to select a constant. Clicking it opens the [dialog box containing the list of predefined constants](#). The selected constant is inserted into the expression in the current cursor position.

## Functions

Buttons to select a function. Clicking it opens the [List of the functions](#) dialog box. The selected function is to be inserted into the expression at the current cursor position.

## Attributes

Buttons to select an attribute. Clicking it opens the [dialog box containing the list of attributes](#). The selected attribute is inserted into the expression at the current cursor position.

## Start value

Value of the object before beginning the calculation. Maximum: 20 characters.



### Start value checkout

Start value is saved in a text form. During the system start, the **Server** assigns values to individual objects according to the start value definition. In case of the successful conversion of a start value (text) to the required type, a value of the eval tag will be valid. If the conversion is not successful, then this value

will be invalid. It is possible to save a start value that cannot be converted to the required type. The button **Start value checkout** () checks the start value and, in case of unsuccessful conversion, reports it.

## Target column

If no target column is defined, the value of the expression (the input field [Expression](#)) will become the value of an eval tag. If it is defined, it must contain the reference to a column of an object of [Structured variable](#) type in form: *ObjectName[0]^ColumnName*. This notation defines a structured variable column, which will contain the results of the expression evaluated.

### Note:

If a constant expression of value 1 is defined, the process D2000 Calc (administrator of objects of Eval tag) type ) ensures that all fields of a defined column will have the same value of 1.

If there is an expression  $Sec+1$ , the process D2000 Calc (administrator of objects of Eval tag) type ) ensures that all fields of a defined column will have the same value, which corresponds to the value of system variable *Sec* increased by 1.

A reasonable use of this option is the possibility to use a reference to a column of a structured variable in an expression, where in brackets (row index) is a value of 0. Such reference will be referred to as a reference to a column of a structured variable. This ensures that when evaluating an expression, the value of which is inserted into e.g. 3rd row, all the indices with the constant value of 0 will be replaced with the value of 3.

### Example:

Structured variable SV.Production consists of 10 rows and 3 columns.

Column	Type	Meaning
ProductsALL	Int	Number of products
Rejects	Int	Number of rejects
ProductsOK	Int	Number of good-quality products

Obviously, the following equation can be formed:

$ProductsOK = ProductsALL - Rejects$

Provided that the columns ProductsALL and Rejects are populated fully automatically and the value of the column ProductsOK is to be determined, it can be done with the configuration of an eval tag as described below:

### Expression:

$SV.Production[0]^ProductsALL - SV.Production[0]^Rejects$

Calculation of the difference will be in the column  $SV.Production[0]^ProductsOK$ .

**Note:** The expression may contain references to columns of other structured variables.

## **Evaluation method**

Selection of a method used to calculate the expression determining the eval tag value. Available evaluation methods:

- **Periodically** - expression is calculated periodically according to the given period (**Period** option) and offset (**Offset** option).
- **On change** - expression is calculated when a value of at most one object placed in the expression notation, has changed.
- **Trigger** - definition of a trigger (condition) that causes the expression to be re-evaluated. The object representing the trigger may be either of a simple type (unstructured) or a column of a [Structured variable](#) object. The object may be defined in several ways:
  - entering the object name into the input field,
  - selection of the object from the list of objects - the list is opened by clicking the button placed right to the input field,
  - creating a new object - the button **Create a new object**.

It is also necessary to determine a given object state when the trigger is valid. The list box placed below the input field to enter the object name contains a list of admissible [object value states](#). This list is different for individual object types. Trigger of expression's re-evaluation will become valid when the object passes into the selected state. If the option **Inverse function** is checked, the trigger is valid, when the object is in another state than the selected one.

## **Evaluation period**

### **Period**

Expression evaluation period (hour:min:sec). It is defined only for the periodical evaluation method of an eval tag.

### **Offset**

The time offset of the basic time interval (hour:min:sec). It is defined only for the periodical evaluation method of the eval tag.

### **Use monotonic time**

## **Maximum density of evaluation [s]**

The parameter can be defined for the [evaluation method On change](#). It allows restricting the evaluation of a given expression so that the expression will be evaluated once within the defined time. It is used especially in cases when the values of the objects defined in the expression are often changed and immediate calculation of the expression is not required.



**Related pages:**

[Eval tags](#)