Configuration Dialog Box (D2000/Data Archiving in D2000 System/Historical Values)

Historical values - 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.

Configuration dialog box of processes consists of several parts (tabs) containing the similar parameters.

General properties

Description

A text string describing the historical value. Maximum: 128 characters.
Possibility to use the Dictionary (to open click the **CTRL+L**).

Status Text

Defines a status text of the historical value. The status text allows to redefine individual values' identification for the historical value.

Transformation palette

Selection of an index to transformation palette. See the topic Transformation palette.

Value type

Selection of a value type of the historical value. The possible types are shown in the following table.

Label	Value type
Int-Integer	Integer
Re-Real	Real
Bo-Boolean	Boolean

Note: Value type may be defined for the archive purpose **Calculate archived values by defined statement** only.

Technical units

Technical units of the historical value. Maximum: 12 symbols. Possibility to use the Dictionary (to open click the **CTRL+L**).

Limits

Technological limits are effective only for historical values which are calculated by D2000 Archiv (evaluated and statistical historical values). There are 4 limits defined: VHL, HL, LL and VLL. Limit can be defined either directly – by the value entry into the input field, or its value can be determined by a system object (dynamic limit) – the button right from the input field.

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

Values of the individual limits determine the state of the historical value according to its value. The relation value- limits gives six possible states.

Limit	Object state according to relation *Value - Limit*
	Above VHL (object value > VHL)
VHL	
	Above HL (HL < object value < VHL)
HL	
	Normal (LL < object value < HL)
LL	
	Bellow LL (VLL < object value < LL)
VLL	
	Bellow VLL (object value < VLL)

As the limits can be dynamic (determined by the object value), a situation when the relation VLL<LL<HL<VHL is not valid (the limits crossing) can occur. The historical value is then in **Limit Problem** state.

Note: Changing the value of dynamic limit will not cause a new evaluation of limits and possibly a new value with a changed Limits attribute. New value value of dynamic limit is taken into account only when archiving a new value.

Archive

Archive purpose

There are the following options:

- Archive object values - allows to archive values of D2000 system objects. Archiving can be either periodical or on value change.
- Ignore identical - optimization for processing of old values.
- Calculate archived values by statistical function - allows to calculate values of defined object of *Historical value* by a statistical function.
- Calculate archived values by defined statement - calculates defined object of *Historical value* type that are defined in the tab Statement. Calculation gives values, that are to be archived.
- Fill archive from script (Value storage) - values can be filled either from an ESL script or manually from the process D2000 HI.

Archiving parameters

The part of the configuration dialog box contains the following options:

- **Archive** - if the option **Archive** is checked, then defined object to archive will be archived. If it is not checked, the object will not be archived.
- **Write Start/Stop** - enables / disables automatic writing of START or STOP values.
- **Depository** - the option allows to enable / disable storing values of the historical value into the depository database.
- **Depository segment** - allows to specify the depository database segment the values to be stored in if the parameter **Depository** is checked. The parameter can be used for Oracle platform with depository database segments enabled or for PostgreSQL platform with depository database segments enabled.

ARCHIVE OBJECT VALUES

Ignore identical

Optimisation of old values processing coming from communication (automatically or as a result of TELL command GETOLDVAL) or values of remote tags (as a result of TELL command GETOLDVAL).
If the checkbox is checked, during processing of old value the archive database is queried whether the value is already present there. If it exists, the value is discarded (and recalcs of statistical or calculated archived values which use this primary archive value are not performed either).
Note: Optimisation is useful e.g. for archiving of I/O tags from communication using the protocol IEC62056-21:2002 File I/O. Its communication files contain several historical values (which KOM process sends as old values) and one new value for every I/O tag.

Object to archive

Definition of a D2000 system object, values of which are to be archived. The object may be defined either by typing its name into the input edit field or selecting from the list of object. To open the list of object click the button right from the input field.

Object to archive or value source may be:

1. An object, value of which is a simple type (Integer, Boolean, Real,...) - just one value is archived, therefore such archived object is called ***simple historical value***.

2. An item of object of Structured variable type - there is also just one value archived, therefore such archived object is called ***simple historical value***, too.
3. A column of object of Structured variable type (e.g. *SV.Strct[0]^ColName*) - there are archived all values of the column. Such archived object is called ***one-column historical value***.
4. An entire object of Structured variable type (e.g. *SV.Struct*) - there are archived all values of the object. Such archived object is called ***structured historical value***.

Note: The above described fact implies, that the object to archive directly specifies the type of historical value - simple, one-column or structured historical one. So all changes in the configuration of the historical value has directly effect on its functionality, mainly if the object is used in other historical values.

Archiving method

When archiving the object values into the primary archive, it is possible to use the following method of archiving:

- **Periodical** - writing values into the archive is periodical. The archiving process in defined time moments stores the archive object value into the archive. Timestamp (the value time) is not given by the occurrence time of the archive object value, but by the time of the value storing into the archive.

  Reading the values stored periodically by means of D2000 system (ESL: GETARCHARR, GETARCHVAL, D2000 ObjApi: GetArchivData, D2000 VBApi: VBApiGetArchData, D2000 WorkBook) follows the rule that the archive object value out of time moments given by the period, is unknown (invalid). The result of the data reading is therefore given by the **oversampling** and the begin (BT) and end (ET) time as follows:
  - **oversampling (step) = 0**
    The reading results are all the values, time of which belong into the interval <BT, ET>.
  - **oversampling (step) <> 0**
    The reading result is the value array with timestamps continuously:
    BT+0*step, BT+1*step, BT+2*step, ..., BT+N*step.
    The number of values is given by the end ET of the time interval. The array value without a record with the same timestamps in the archive is invalid. The value with such a record is filled according to it. The above facts imply, that when reading periodical data, it is necessary (advisable):
    - to adjust BT exactly for some of the object archiving moments given by the period and time offset of the archiving.
    - oversampling value (step) must be an integer multiple of the archiving period.
    - ET = BT + (N-1)*step, where N is an integer number: the number of values in the final selection.

    Note: The statistical archive, from the reading point of view, acts periodically.
- **On value change** - just value changes of the archive object undetected by the value filter are stored into the archive.

  Reading of values stored by using a filter by means of D2000 system (ESL: GETARCHARR, GETARCHVAL, D2000 ObjApi: GetArchivData, D2000 VBApi: VBApiGetArchData, D2000 WorkBook) follows the rule that the archive object value at any time (**t**) is given (equal) by the last historical value before the given time (**t**). The data reading result is therefore given by the oversampling and the begin (BT) and end (ET) time as follows:
  - **oversampling (step) = 0**
    The reading result are all the values, time of which belongs to the interval <BT, ET> and 1 value before BT time, in case there is no value exactly equal with BT time in the archive.
  - **oversampling (step) <> 0**
    The reading result is a array of values with timestamps continuously:
    BT+0*step, BT+1*step, BT+2*step, ..., BT+N*step.
    The number of values is given by the end ET of the time interval. The array value without any record in the archive with the same timestamp will be given by the last value before the time required.

Publish values

If the option **Publish values** is checked, then the historical value publishes its last archived value in the way, that depends on the object to archive as follows:

- For a simple historical value - the object of *Historical value* type gets the last value.
- When archiving an one-column historical value - last archived values of individual items are filled into the relevant items of the column of a Structured variable type object defined by the parameter **Target object**.
- When archiving a structured historical value - last archived values of individual items are filled into the relevant items of a Structured variable type object defined by the parameter **Target object**.

Note: To ensure the correct functionality of the feature **Publish values** for an **one-column historical value** (**structured historical value**) the number of rows (columns) of the structured variable defined in the parameter **Target object** must be the same as the number of rows (columns) of the object defined by the parameter Object to archive.

Target object

The input field is enabled if the option **Publish values** is checked. It allows you to define an object, that will contains values of the historical value. When archiving a simple historical value, target object must not be defined, but **must** be defined for one-column historical value and structured historical value - the size of target object must be the same as the size of the object defined by the parameter Object to archive.

CALCULATE ARCHIVED VALUES BY STATISTICAL FUNCTION

Historical value

Definition of an object of *Historical value* type, values of which are to be calculated. It can be:

- **simple HV** - simple historical value, item of one-column historical value (e.g. *H.ColArchiv[2]*) or item of structured historical value (e.g. *H.Struct[2]^ColName*),
- **one-column HV** - **one-column historical value** (e.g. *H.ColArchiv*) or column* of **structured historical value** (e.g. *H.Struct[0]^ColName*),
- **structured HV** - **structured historical value.**

* The list of columns is given by the respective structure definition, that defines the structure of the historical value (e.g. *H.Struct[2]^ColName*).

Publish values

If the option **Publish values** is checked, then the historical value publishes its last archived value in the way, that depends on the object defined by the parameter Historical value as follows:

- **For a simple HV** - the object of *Historical value* type you are configuring gets the last value (if the parameter Historical value is not defined).
- **When archiving an one-column HV** - last archived values of individual items are filled into the relevant items of the column of a Structured variable type object defined by the parameter **Target object**.
- **When archiving a structured HV** - last archived values of individual items are filled into the relevant items of a Structured variable type object defined by the parameter **Target object**.

Note: To ensure the correct functionality of the feature **Publish values** for an **one-column HV** (**structured HV**) the number of rows (columns) of the structured variable defined in the parameter **Target object** must be the same as the number of rows (columns) of the object defined by the parameter Historical value.

Target object

The input field is enabled if the option **Publish values** is checked. It allows you to define an object, that will contains values of the historical value. When archiving a **simple HV**, target object may not be defined, but **must** be defined for **one-column HV** and **structured HV** (see the parameter Historical value) - the size of target object must be the same as the size of the object defined by the parameter Historical value.

CALCULATE ARCHIVED VALUES BY DEFINED STATEMENT

The option allows to define a mathematic expression among objects of *Historical value* type (the tab Statement). The expression can not contain the references to any current values of D2000 system objects. Calculation of the expression provides values to be archived.

For example:
We have got two measurement points - two I/O tags which are archived as the historical values *H.Flow1* and *H.Flow2*. You need to archive the sum of both the flows. You can use two methods:

1. Create an object of Eval tag type, that adds together the values of the I/O tags and then archive it.
2. Create an object of *Historical value* type with the expression of „H.Flow1 + H.Flow2“.

Both the methods gives the same result. A problem may occur, when you need to modify e.g. the value of the object *H.Flow1* already archived. If you have used the first method, you must also manually change the value of the historical value. The second method automatically recalculates the expression and corrects the sum.

Calculation methods

Definition of a methods to calculate given expression - periodically or on value change. If the option **On value change** is selected, the expression will be recalculated when a value of at least one objects defined in the expression is changed.

Object of *Historical value* type defined in this way can be either simple or one-column historical value. If the object is **one-column historical value**, the expression may also contain the references to other structured or **one-column historical values** with the row index of 0. During the calculation, the index is dynamically replaced by current row number of the one-column historical value to recalculate. That allows to define the same expression for all column items.

Publish values

If the option **Publish values** is checked, than the historical value will publish its last value:

- If the option is not checked, the parameter **Structure size** defines whether the historical value is **simple** or **one-column** one.
  If **Structure size** is not defined, the historical value is a **simple** one. If it is defined, the object is **one-column historical value** - possible to define a column of a Structured variable type object, column of a structured historical value or a one-column historical value. The number of rows of the calculated one-column historical value is given by the number of rows of the objects defined in the parameter **Structure size**.
- If the option is checked, the parameter **Target column** specifies whether the calculated historical value is simple or one-column.
  If **Target column** is not defined, the historical value will be a **simple** one that gets the last value.
  If it is defined, the object will be a **one-column historical value** - possible to define a column of a Structured variable type object. So, the number of rows of the historical value is given by the number of rows of the target column.

Archive size

Definition of archive size - see the parameter Publish values. The parameter will appear, if the parameter **Publish values** is not checked.

Target column

Definition of an object, that will contain values of the historical value - see the parameter Publish values. The parameter will appear, if the option **Publish values** is checked.

FILL ARCHIVE FROM SCRIPT (VALUE STORAGE)

Object of *Historical value* type filled from a script can be used as a storage of values which are not generated by archiving the values of other D2000 system object, statistical calculation or other expression calculation. Values can be only filled from an ESL script or manually from the process D2000 HI.

Value storage can be simple, one-column or structured one. This is given by the parameter **Archive structure** or **Target structure** (it depends on the parameter **Publish values**)

Inserted values are periodical

If the option is checked, you must define a period and offset (the tab Time parameters).

Publish values

Checking the option **Publish values** "renames" the parameter **Archive structure** to **Target structure**. The functionality of the parameter to define the archive is not changed, but it is not possible to define an object of *Historical value* type.
If the option **Publish values** is checked, the parameter **Target structure** defines whether the historical value is **simple** or **structured** one. If **Target structure** is not defined, the historical value is a simple one and gets the last value. If it is defined, the historical value is a structured one and will fill last value into the respective item of defined structured variable.

Archive playback

The parameter is enabled when the parameters **Inserted values are periodical** and **Publish values** are checked. If it is checked, the historical value will not publish its last value but the value that is valid according to the current time (e.g. from a ESL script, the feature allows to fill an object of *Historical value* type with values containing the future timestamps and the object "plays" them in real time).

Archive structure

The parameter appears when the parameter **Publish values** is not checked.

- If it is not defined, the historical value is a simple one.
- The historical value is a one-column one, if the parameter contains:
  - column of a structured historical value (e.g. *H.Struct[0]^ColName*)
  - **one-column historical value** (e.g. *H.ColArchiv*)
  - column of an object of Structured variable type (e.g.. *SV.Struct[0]^ColName*)
- The historical value is a **structured** one, if the parameter contains:
  - **structured historical value**
  - object of Structured variable type

Target structure

The parameter will appears if the parameter **Publish values** is checked. Allows to define an object, that will contains values of the historical value - see the parameter Publish values. If it is not defined, the historical value itself will contain the values.

Statement

In the top part of the tab, there is placed the input edit filed for entering the expression, that determines the value of the historical value. The expression can contains functions, constants, attributes but objects of *Historical value* type only. The expression may also contain the extended syntax.

Objects

The button allows to select an object of D2000 system. Selected object is to be inserted into the expression on the current cursor position.
Note: Expression can contain objects of *Historical value* type only.

Constants

The button allows to select a constant. Clicking it open the dialog box containing the list of predefined constant. Selected constant is to be inserted into the expression on the current cursor position.

Functions

The button allows to select a function. Clicking it opens the List of functions dialog box. Selected function is to be inserted into the expression on the current cursor position.

Attributes

The button allows to select an attribute. Clicking it open the dialog box containing the list of attributes. Selected attribute is to be inserted into the expression on the current cursor position.

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 value. 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.

Maximum density of evaluation [s]

The parameter can be defined for the archive method **Filter**. It allows to restrict the number of evaluations of given expression so that the expression will be evaluated once within defined time. It is used especially in cases, when the values of the objects defined in the expression are often changed and calculating the expression is not required immediately. It is used especially in cases, when the values of the objects defined in the expression are often changed and calculating the expression is not required immediately.

Calculation

Method of the statement calculation:

- *Continuous* - continuous (on the fly) calculation. Result values are calculated on the fly and they are automatically available (in dependence on the system load). A disadvantage of the method is a higher demand on the computing power (especially for frequent changes of primary historical values).
- *On demand* - calculation is executed and result is stored to the archive on demand. The demand can be generated by the action CALCONDEMANDSTAT or the Tell command RECALC).
  Note: historical value calculated *on demand* should not have any depending historical values calculated *continuously*, because the result would be wrong.
- *On read* - calculation is executed as a result of a read request. An advantage of this method is that values are not stored in the database so they don't occupy any disk space. There is no possibility of re-calculation in case of writing delayed data into the archive. A disadvantage is a necessity to read source data and calculate it for each read demand.
  Note: historical value calculated *on read* should not have any depending historical values calculated *continuously* or *on demand*, because the result could be wrong (due to delayed data) or the calculation could be ineffective (multiple calculations of a single *on read* object if it is used by several other historical values).

Time parameters

Archiving period

The parameters define the period (Hours : Minutes : Seconds) and the time offset within the period for the periodical primary and statistical archiving method (Hours : Minutes : Seconds).

History depth

Archiving time (Months : Days : Hours). The parameter determines the archiving time depth. It is the minimal time period, during which the data will be stored in the on-line archive. The older data are being erased from the archive.

The maximum history depth is 800 months (approximately 66 years).

Stored time

For periodical data archiving, it is defined, what time data with the value for the given period will be stored into the archive. The time data can present the begin time - **Period begin** option or the end time of the interval (period) - **Period end** option.

Note: For archives filled from script, this setting does not directly affect anything - the data has a time stamp with which it was stored from the script. It however affects the calculation of statistics and whether statistics are calculated from the value at the edge of the interval.

Conditions

Defining the conditions for the start and interruption of the archiving is provided by the mechanism of the object archiving dynamical control depending on values or states of others objects in the system. Both conditions need not have to be defined. If not defined, the archiving will be started immediately after the start and initialisation of the process Archive.

Start condition

Defining the condition causing the start of the given historical value.

The object representing the archiving start condition may be defined by several ways:

- writing the object in to the input field,
- selecting the object from the list of objects - the list opened after clicking the button right from the input field,
- creating a new object - **Create new object** button.

Also, it is necessary to define, for what state of the given object is the condition valid. In the list under the object entry field, there are displayed possible object value states. This list is different for various kinds of types of objects. The archiving start condition will be valid, if the object is in the chosen state. If the option Inverse function is enabled, the condition is valid, if the object is in status other than the chosen state.

Stop condition

Defining the condition that causes stopping the archiving of the given historical value.

The object representing the archiving stop condition may be defined by several ways:

- writing the object name into the input field,
- selecting the object from the list of objects - the list is opened by clicking the button right from the input field,
- creating a new object - **Create new object** button.

Also, it is necessary to define, for what state of the given object is the condition valid. In the list under the object entry field, there are displayed possible object value states. This list is different for various kinds of types of objects. The archiving stop condition will be valid, if the object is in the chosen state. If the option Inverse function is enabled, the condition is valid, if the object is in status other than the chosen state.

Statistics

Statistical function

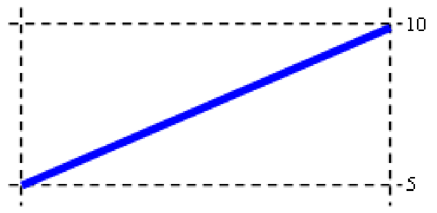
When archiving into the statistical archive it is possible to use these implemented functions.

Function	Meaning
None	No function.
Average *	Arithmetical average of all archive object values.
W-Average *	Weighted arithmetical average of all archive object values.
Integral	Time integral of historical values.
Sum	Sum of archive object values.
Maximum	Maximum of archive object values.
Minimum	Minimum of archive object values.
Count	Number of archive object values.
Filter	Applying a filter for value storing into the statistical archive.
Increment	If the newer value is greater than the older one, then the difference between the values, otherwise the newer value (the function is useful to process counter values that oveflow and start from zero again). Parameter **(Compare value)** – weight of impulse. The result will be the impulse multiplied by its weight. Weight of 1 will ensure standard behaviour.
Delta	Delta between values. Parameter **(Compare value)** – weight of impulse. The result will be the impulse multiplied by its weight. Weight of 1 will ensure standard behaviour.
EcoAvg	Average of the object values within the elapsed time period (**Period** parameter in **Time parameters** tab) according the methodology based on flags of individual values entering the statistic. The same purpose is fulfilled by the function %EcoAveR, that is implemented for eval tags.
GT Time (>)	The function calculates the time, during which the value of the historical value was greater than the entered constant (**Compare value**).
GE Time (>=)	The function calculates the time, during which the value of the historical value was greater or equal to the entered constant (**Compare value**).
LT Time (<)	The function calculates the time, during which the value of the historical value was lower then the entered constant (**Compare value**).
LE Time (<=)	The function calculates the time, during which the value of the historical value was lower or equal to the entered constant (**Compare value**).
Maximum in time interval	Obsolete - do not use!
Minimum in time interval	Obsolete - do not use!

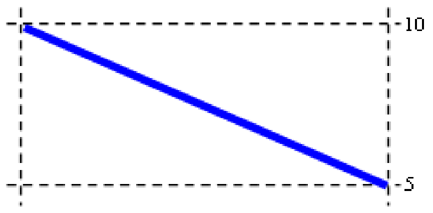
Number of local maximums	
Number of local minimums	
Sum of positive values	Sum of positive values of the historical value.
Sum of negative values	Sum of negative values of the historical value.
Average of positive values	Arithmetical average of positive values of the historical value.
Average of negative values	Arithmetical average of negative values of the historical value.
Sum of increments	Sum of increments for given time interval. If the new value is less than the old value, the increment is 0. Parameter **(Compare value)** – weight of impulse. The result will be the impulse multiplied by its weight. Weight of 1 will ensure standard behaviour.
Time slice**	Object value in given time moments.
Sample standard deviation	The function calculates the sample standard deviation of all values of the archive object.

* For non-periodical values we recommend you to use the **W-Average** (weighted average) function and the **Average** function for periodical values.
** The function allows recalculation of the historical value if historical values archived primarily have been changed.

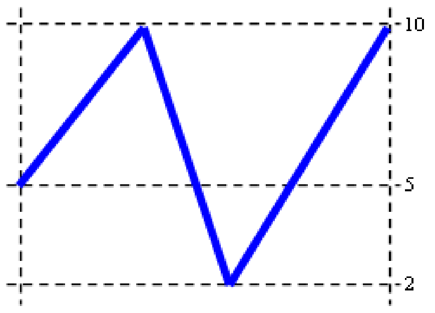
The difference among the functions **INCREMENT**, **DELTA** and **SUM OF INCREMENTS** is shown in the following figures.



DELTA = INCREMENT = SUM OF INCREMENTS = 5



DELTA = -5
INCREMENT = 5
SUM OF INCREMENTS = 0



DELTA = 5
INCREMENT = 15
SUM OF INCREMENTS = 13

In the first case, all three functions are equal to 5 (10-5)

In the second case

- DELTA = 5-10 = -5
- INCREMENT = 5 *(because 5 <10)*
- ADDITIONAL AMOUNT = 0 *(because 5 <10)*

In the third case

- DELTA = (10 - 5) + (2 - 10) + (10 - 2) = 5
- INCREMENT = (10-5) + 2 *(because 5 <10)* + (10-2) = 15
- ADDITIONAL AMOUNT = (10-5) + 0 *(because 5 <10)* + (10 - 2) = 13

Calculation

Statistics calculation method:

- *Continuous* - continuous (on the fly) calculation. Result values are calculated on the fly and they are automatically available (in dependence on the system load). A disadvantage of the method is a higher demand on the computing power (especially for frequent changes of primary historical values).
- *On demand* - calculation is executed and result is stored to the archive on demand. The demand can be generated by the action CALCONDEMANDSTAT or the Tell command RECALC). **Note:** historical value calculated *on demand* should not have any depending historical values calculated *continuously*, because the result would be wrong.
- *On read* - calculation is executed as a result of a read request. An advantage of the method is that values are not stored in the database so they don't occupy any disk space. There is no chance of wrong calculation in case of writing delayed data into the archive. A disadvantage is a necessity to read source data and calculate it for each read demand.
  Note: historical value calculated *on read* should not have any depending historical values calculated *continuously* or *on demand*, because the result could be wrong (due to delayed data) or the calculation could be ineffective (multiple calculations of a single *on read* object if it is used by several other historical values).

Validation criteria

The value of the parameter Validation criteria defines the value percentage in the primary archive (used fro the calculation of values stored into the statistical archive) that has to be valid, in order to acquire a valid result. If there were less valid values than stated in **Validity criteria**, the result will be Weak_Value.

Time interval for statistical calculation

Statistical time period defines the time interval, the set of historical values, that will be processed by the particular statistical function. By default, the interval is equal with the archiving period. If it is necessary to enter different period, check the option **Time interval different from archiving period** and enter required period into the parameter **Interval**. The time period must be greater than 0 [s].

Compare value

A parameter for the functions ***GT Time (>)***, ***GE Time (>=)***, ***LT Time (<)***, ***LE Time (<=)***.

Integral time units

A parameter for the function ***INTEGRAL***:

- **Hours** - hour integral
- **Minutes** - minute integral
- **Seconds** - second integral

Filter

The system allows to archive considerable value changes of the archive object. This archiving method represents defining of three deadband levels, that are possible to set various values of significant changes.

Values of filtering:

- **High limit** - defining the high limit for filtering.
- **Low limit** - defining the low limit for filtering.
- **Above limit** - defines a significant change of the archive object above the high limit.
- **In limit** - defines a significant change of the archive object within the the lower and high limits.
- **Below limit** - defines a significant change of the archive object below the low limit.

Related pages:

Historical values