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

Historical values - configuration dialog box

Editing of all objects in the D2000 CNF process is 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 processes consists of several parts (tabs) containing related 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 historical value. The status text allows for redefining 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 purpose of a historical value of "**Calculate archived values by defined statement"** type only.

Technical units

Technical units of 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. The limit can be defined either directly – by the value entered 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 give 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 the **Limit Problem** state.

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

Archive

Archive purpose

There are the following options:

- Archive object values - (primary archives) implement archiving of values of D2000 objects. Archiving can be either periodical or on value change.
- Calculate archived values by statistical function (statistical archives) - implement applying of a statistical function to a defined object of *Historical value.*
- Calculate archived values by defined statement (evaluated archives) - support defining a custom formula on top of existing objects of *Historical value* type. The formula is defined in the Statement tab. The results of the calculation are again values that will be archived.
- Fill archive from the script (Value storage) - values can be written either from an ESL script or manually from the process D2000 HI.
  Note: all types of *Historical values* (also primary, statistical, and evaluated) can be modified 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 the defined object will be archived. If it is not checked, the object will not be archived (the archiving is disabled).
- **Write Start/Stop** - enables/disables automatic writing of START or STOP values.
- **Depository** - the option allows enabling/disabling of storing values of the historical value into the depository database.
- **Depository segment** - allows specifying the depository database segment the values to be stored in if the parameter **Depository** is checked. The parameter can be used for the Oracle platform with depository database segments enabled or for the PostgreSQL platform with depository database segments enabled.

ARCHIVE OBJECT VALUES

Ignore identical

Optimization of processing of old values of primary archives coming from communication (automatically or as a result of the GETOLDVAL command sent to the D2000 KOM process) or values of remote tags (as a result of TELL command GETOLDVAL sent to the D2000 Gateway Client process).
If the checkbox is checked, during the processing of the old value, the cache or archive database is queried whether the same value is already present there. If it exists, the value is discarded (and recalcs of statistical or calculated archived values that 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 IEC62056-21:2002 File I/O protocol. Its communication files contain several historical values (which the D2000 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 by selecting it from the list of objects. To open the list of objects, click the button right from the input field.

Object to archive or value source may be:

1. An object, the value of which is a simple type (Integer, Boolean, Real,...) - just one value is archived, therefore such archived object is called a ***simple historical value***.
2. An item of an object of Structured variable type - there is also just one value archived, therefore such archived object is called a ***simple historical value***, too.
3. A column of an object of Structured variable type (e.g. *SV.Strct[0]^ColName*) - there are archived all values of the column. Such an archived object is called a ***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 an archived object is called a ***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 ones. So all changes in the configuration of the historical value have a direct 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 in the archive are periodical. The archiving process stores the archive object value in the archive in defined time moments. Timestamp (the time of value) is not determined by the timestamp of the archived object value (e.g. an I/O tag), but by the time when the value is periodically written into the archive.

  Reading the values stored periodically by means of the D2000 system (ESL: GETARCHARR, GETARCHVAL, GETARCHROW, GETARCHCOL, GETARCHSTRUCT, D2000 ObjApi: GetArchivData, D2000 VBApi: VBApiGetArchData, D2000 WorkBook) follows the rule that the archive object's value out of time moments given by the period, is unknown (invalid). The result of the data reading is therefore given by the **resampling** and the begin (BT) and end (ET) time as follows:
  - **resampling (step) = 0**
    The reading results are all the values, the time of which belongs to the interval <BT, ET>.
  - **resampling (step) <> 0**
    The reading result is an array of values with timestamps contiguously:
    BT+0*step, BT+1*step, BT+2*step, ..., BT+N*step.
    The number of values is given by the end of the time interval (ET). The value of the array item is invalid if a record with the same timestamps does not exist in the archive database. If such a record exists, it is returned as a value of the array item. The above facts imply that when reading periodical data, it is necessary (advisable):
    - to adjust BT exactly to some of the object archiving moments, given by the period and time offset of the archiving.
    - resampling 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, as far as reading is concerned, behaves as a periodical archive.
- **On value change** - all the value changes of the archive object, which are not filtered out by the value filter, are stored in the archive.

  Reading of values stored by using a filter by means of the D2000 system (ESL: GETARCHARR, GETARCHVAL, GETARCHROW, GETARCHCOL, GETARCHSTRUCT, D2000 ObjApi: GetArchivData, D2000 VBApi: VBApiGetArchData, D2000 WorkBook) follows the rule that the archive object's value at any time (**t**) is given by (equals to) the last historical value at or before the given time (**t**). The data reading result is therefore given by the **resampling** and the begin (BT) and end (ET) time as follows:
  - **resampling (step) = 0**
    The reading results are all the values, the time of which belongs to the interval <BT, ET> and one value before BT time, in case there is no value with a time exactly equal to BT time in the archive.
  - **resampling (step) <> 0**
    The reading result is an 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. If a record with the same timestamps does not exist in the archive database, the value of the array item will be equal to the last value before the specific required (however, the timestamp will be set accordingly).

Publish values

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

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

Note: To ensure correct functionality of the **Publish values** feature for a **one-column historical value** (resp. **structured historical value**), the number of rows (in case of a **structured historical value**, also the number of columns) of the structured variable defined in the **Target object** parameter 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 **Publish values** option is checked. It allows you to define an object, that will contains values of historical value. When archiving a **simple historical value**, the target object must not be defined. However, it **must** be defined for **one-column historical value,** and for **structured historical value** - the size of the 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 historical value** - 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 historical value** - **one-column historical value** (e.g. *H.ColArchiv*) or column* of **structured historical value** (e.g. *H.Struct[0] ^ColName*),
- **structured historical value** - **structured historical value.**

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

Publish values

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

- **For a simple historical value** - the object of the *Historical value* type you are configuring gets the last value (unless the parameter Historical value is defined).
- **When archiving a one-column historical value** - the last archived values of individual items are published in the relevant items of the column of a Structured variable type object defined by the **Target object** parameter.
- **When archiving a structured historical value** - the last archived values of individual items are published in the relevant items of a Structured variable type object defined by the **Target object** parameter.

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

Target object

The input field is enabled if the **Publish values** option is checked. It allows you to define an object, that will contains values of historical value. When archiving a **simple historical value**, the target object must not be defined. However, it **must** be defined for **one-column historical value** and for **structured historical value** (see the Historical value parameter) - the size of the 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 defining a mathematic expression containing objects of *Historical value* type (the Statement tab). The expression can not contain references to any other D2000 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 historical values *H.Flow1* and *H.Flow2*. If you need to archive the sum of both 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 archives it.
2. Create an object of *Historical value* type with the expression of „H.Flow1 + H.Flow2".

Both methods give the same result. A problem may occur when you need to modify e.g. the value of the object *H.Flow1*, which is already archived. If you 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. So the D2000 Archive keeps the calculated (and also statistical) archives in synchronization with their source archives.

Calculation methods

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

An object of *Historical value* type defined in this way can be either simple or one-column historical value. If the object is a **one-column historical value**, the expression may also contain references to other structured or **one-column historical values** with a row index of 0. During the calculation, the index is dynamically replaced by the current row number of the one-column historical value to recalculate. That allows defining the same expression for all column items.
Example of a structured expression: H.SourceColumn[0] + H.SourceStruct[0]^ColA

Publish values

If the option **Publish values** is checked, then 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 a **one-column** one.
  If **Structure size** is not defined, the historical value is a **simple** one. If it is defined, the object is a **one-column historical value** - it is possible to define a column of a Structured variable type object, a 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 the **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** - it is possible to define a column of an object of Structured variable type. So, the number of rows of the historical value is given by the number of rows of the target structured variable.

Archive size

Definition of archive size - see the Publish values parameter. The parameter will be visible only if the **Publish values** parameter is not checked.

Target column

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

FILL ARCHIVE FROM SCRIPT (VALUE STORAGE)

An object of *Historical value* type filled from a script can be used as a storage of values that are not generated by archiving the values of other D2000 system objects, statistical calculations, or by other expression calculations. Values can be only filled from an ESL script or manually from the D2000 HI proc ess.

Value storage can be simple, one-column, or structured. This is defined 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 Time parameters tab).

Publish values

Checking the option **Publish values** "renames" the parameter **Archive structure** to **Target structure**. The functionality of the parameter to define the structure of the archive is not changed, however, it is not possible to define an object of *Historical value* type.
If the **Publish values** option is checked, the parameter **Target structure** defines whether the historical value is a **simple** or **structured** one. If the **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 publish the last value into the respective item of a 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. (This feature supports storing data with future timestamps in an object of *Hi storical value* type - e.g. from an ESL script - and then the object "plays" these values in real-time).

Archive structure

The parameter appears when the **Publish values** parameter 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**
  - an object of Structured variable type

Target structure

The parameter will appears if the **Publish values** parameter is checked. It allows defining an object, that will contain values of the historical value - see the Publish values parameter. If it is not defined, the historical value itself will contain the values.

Statement

In the top part of the tab, an input edit field is placed. It serves for entering the expression, that determines the value of the historical value. The expression can contain functions, constants, and attributes - but objects must be of *Historical value* type only. The expression may also contain extended syntax.

Objects

The button allows selecting an object of the D2000 system. The 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 selecting a constant. Clicking it opens the dialog box containing the list of predefined constants. The selected constant will be inserted into the expression on the current cursor position.

Functions

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

Attributes

The button allows selecting an attribute. Clicking it opens the dialog box containing the list of attributes. The 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 values. Only the values of input objects are replaced, and invalid values of intermediate data are not replaced. Values of input 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 restricting the number of evaluations of a given expression so that the expression will be evaluated once within a defined time to reduce CPU and disk I/O usage. 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 (depending on the system load). A disadvantage of the method is a higher demand for computing power (especially for frequent changes of primary historical values that enter into the expression).
- *On demand* - the calculation is executed and the result is stored in the archive on demand. The demand can be generated by the CALCONDEMANDSTAT action or by the RECALC command).
  Note: historical values calculated "*on-demand*" should not have any depending on historical values calculated *continuously* because their results would be wrong.
- *On read* - the 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 request. This method should be used for objects whose values are rarely needed.
  Note: historical value calculated "*on-read*" should not have any depending historical values calculated *continuously* or "*on-demand*", because the result could be wrong in some cases (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 online 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 defines what time data with the value for the given period will be stored in the archive. The time data can present either the beginning time - the "**Period begin**" option or the end time of the interval (period) - the "**Period end"** option.

Note 1: For archives filled from the script, this setting does not directly affect anything - the data has a timestamp 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.

Note 2: when displaying data in graphs, the parameter "Stored time" is taken into account:

- for primary periodic archives, the value is plotted for the entire period (regardless of whether the start or end time is stored)

- for statistical archives, the value is plotted for the entire period (regardless of whether the start or end time is stored).
  The exception is the "time slice", which depends on the value of the parameter "Stored time":
  - if "*Period begin*" is set, the value for period <T, T+Period> has timestamp T and is plotted for period <T, T+Period>
  - if "*Period end*" is set, the value for period <T, T+Period> has timestamp T+Period and is plotted for period <T+Period, T+2*Period>
- for script-filled archives, it depends on the value of the parameter "Stored time":
  - if "*Period begin*" is set, the value with time T is plotted for period <T, T+Period>
  - if "*Period end*" is set, the value with time T is plotted for period <T-Period, T>

Conditions

Defining the conditions for the start and interruption of the archiving provides the mechanism of dynamical control of the object archiving, depending on the values or states of other 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 initialization of the process Archive.

Start condition

Definition of the condition causing the start archiving of the given historical value.

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

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

Also, it is necessary to define, for what state of the given object the condition is valid. In the list under the object entry field, the possible object value states are displayed. This list is different for specific types of objects. The archiving start condition will be valid if the object is in the selected state. If the option Inverse function is enabled, the condition is valid, if the object is in a 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 in several ways:

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

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

Statistics

Statistical function

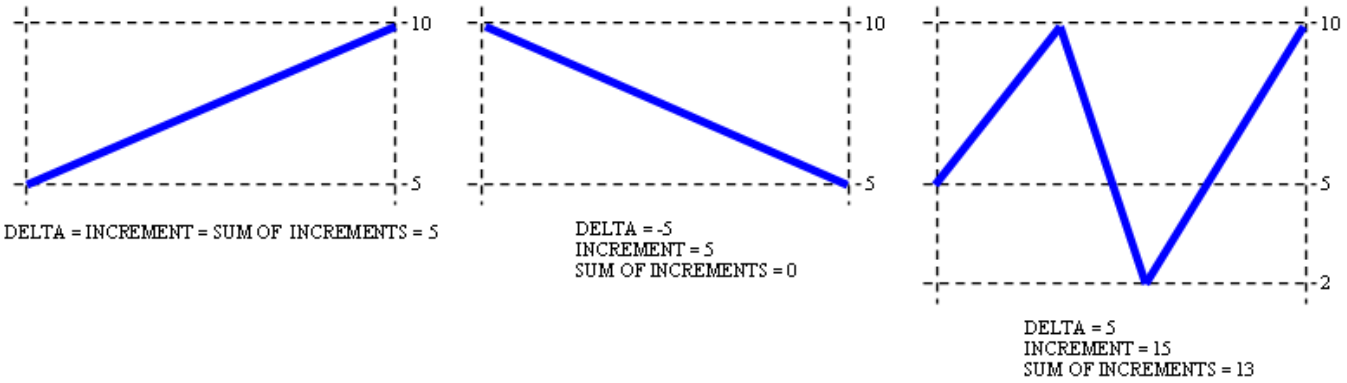
When archiving into the statistical archive, it is possible to use these implemented functions.

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 is integral to historical values.
Sum	Sum of archive object values.
Maximum	Maximum of archive object values.
Minimum	Minimum of archive object values.
Count	The number of archive object values.
Filter	Applying a filter for value storing in 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 overflow and start from zero again). Parameter **(Compare value)** – the weight of impulse. The result will be the impulse multiplied by its weight. A weight of 1 will ensures standard behavior.

Delta	Delta between values. Parameter **(Compare value)** – the weight of impulse. The result will be the impulse multiplied by its weight. A weight of 1 will ensure standard behavior.
EcoAvg	Average of the object values within the elapsed time period (**Period** parameter in **Time parameters** tab) according to the methodology based on flags of individual values entering the statistic. The same purpose is fulfilled by the function %EcoAveR, which 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 than 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 the time interval	Obsolete - do not use!
Minimum in the time interval	Obsolete - do not use!
Number of local maximums	The number of local maximums in a given time interval.
Number of local minimums	The number of local minimums in a given time interval.
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 a given time interval. If the new value is less than the old value, the increment is 0. Parameter **(Compare value)** – the weight of impulse. The result will be the impulse multiplied by its weight. A weight of 1 will ensure standard behavior.
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 using the **W-Average** (weighted average) function, for periodical values the **Average** function is appropriate.
** The function allows recalculation of the historical value if historical values archived primarily have been changed. In addition, the time of the end of the interval is always stored (this 'statistics' basically only resamples the source archive with the selected period).

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



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)*
- SUM OF INCREMENTS = 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
- SUM OF INCREMENTS = (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 (depending on the system load). A disadvantage of the method is a higher demand for computing power (especially for frequent changes of primary historical values that enter into the expression).
- *On demand* - the calculation is executed and the result is stored in the archive on demand. The demand can be generated by the CALCONDEMANDSTAT action or by the RECALC command).
  Note: historical values calculated "*on-demand"* should not have any depending on historical values calculated *continuously* because their results would be wrong.
- *On read* - the 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 request. This method should be used for objects whose values are rarely needed.
  Note: historical value calculated "*on-read"* should not have any depending historical values calculated *continuously* or "*on-demand"*, because the result could be wrong in some cases (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 for the calculation of values stored in the statistical archive) that has to be valid, in order to acquire a valid result. If there were fewer valid values than stated in the **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 to the archiving period. If it is necessary to enter a different period, check the option **Time interval different from archiving period** and enter the 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

Include edge values

This parameter (available from D2000 version 22) determines whether a value with an interval start/end time enters the calculation. The parameter is configurable for functions:

- Average
- Sum
- Maximum
- Minimum
- Count
- EcoAvg
- Maximum in the time interval
- Minimum in the time interval
- Sum of positive values
- Sum of negative values
- Average of positive values
- Average of negative values
- Sample standard deviation

It cannot be configured for other functions:

- W-Average
- Integral
- Filter
- Increment
- Delta
- GT Time (>)
- GE Time (>=)
- LT Time (<)
- LE Time (<=)
- Number of local maximums
- Number of local minimums
- Sum of increments
- Time slice

Filter

The system allows archiving of significant changes in the values of the archived object. This method of archiving represents the definition of three sensitivity bands in which it is possible to enter different values of a significant change.

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 for value above the high limit.
- **In limit** - defines a significant change of the archive object for value within the lower and high limits.
- **Below limit** - defines a significant change of the archive object for value below the low limit.

Related pages:

Historical values