

# D2\_CalcStatFunc

## Accessing historical values - D2\_CalcStatFunc function

The function allows you to calculate the specified statistical archive function on demand.

### Declaration

D2\_CalcStatFunc

```
(archIdent, bt, et, statFuncID, validPerc, paramIdent, refresh)
```

### Parameters

archIdent	TEXT type	<ul style="list-style-type: none"><li>Reference to one value of historical value.</li><li>Reference to simple value of object (not structured variable).</li><li>In case of structured variable it is one item (SV.Struct[2]^Item), otherwise name of object (e.g. I/O tag). In this case the system will automatically look for a "suitable" object of Historical value type. If the column "Item" is of the Object type it is possible to specify a string ",ALL" right after the name of column (e.g. SV.Struct[4]^Item,ALL). This causes the historical values to be obtained for the connected object, not for the item of a structured variable.</li></ul>
bt	ABS. TIME type	Interval begin time.
et	ABS. TIME type	Interval end time.
statFuncID	INT type	Statistical function type. The type is represented by a numerical value (see the <a href="#">table</a> ).
validPerc	INT type	Validation criteria.
paramIdent	REAL or INT types	Parameter for some types of functions.
refresh		Optional parameter of optional type.

### Description

The function calculates the statistical function specified by the parameter *statFuncID* on the values represented by the historical value *archIdent* within the time interval given by the parameters *bt* and *et*. After successful execution, the function returns a result value.

The parameter *validPercIdent\_Int* represents [Validation criteria](#) used during evaluation of the statistical function. It can get the values within 0 .. 100 (the error ERR\_RANGE\_ERROR occurs if the condition is broken).

For some statistical function, it is required to enter the parameter *paramIdent* according to the table:

Statistical function	Parameter description								
_STAT_F_INTEGRAL	<p><a href="#">Integral time units</a>. For individual types, there are also established predefined local variables according to the table:</p> <table><tr><th>Constant</th><th>Description</th></tr><tr><td>_INTEGRAL_HOURUNIT</td><td>Hour integral</td></tr><tr><td>_INTEGRAL_MINUNIT</td><td>Minute integral</td></tr><tr><td>_INTEGRAL_SECUNIT</td><td>Second integral</td></tr></table>	Constant	Description	_INTEGRAL_HOURUNIT	Hour integral	_INTEGRAL_MINUNIT	Minute integral	_INTEGRAL_SECUNIT	Second integral
Constant	Description								
_INTEGRAL_HOURUNIT	Hour integral								
_INTEGRAL_MINUNIT	Minute integral								
_INTEGRAL_SECUNIT	Second integral								

