CALCSTATFUNCARR

CALCSTATFUNCARR action

Function

Calculation of statistical archive function for all specified item intervals.

Declaration

```
CALCSTATFUNCARR archIdent, bt_TmA, et_TmA, step, deepTime,
statFuncIdent_Int, validPercIdent_Int, paramIdent,
locVarColValueIdent_Rec, statusIdent_Int
```

```
[,archivInstance_Int]
```

Parameters

archIdent	in	Reference to one value of historical value or reference to an object, values of which have been archived. **Warning:** If the parameter is the reference to an object archived several times, there is not specified which one of the historical objects is to be used.
bt_TmA	in	Identifier of *AbsTime* type - interval beginning.
et_TmA	in	Identifier of *AbsTime* type - interval end.
step	in	Identifier of *Int* type - time step [s]. *Note:* the parameter must be a positive integer.
deepTime	in	Identifier of *Int* type - time-depth [s] for calculating the statistics. *Note:* The parameter must be a positive integer.
statFunctIdent_Int	in	Identifier of *Int* type - statistical function type.
validPercIdent_Int	in	Identifier of *Int* type - validation criteria.
paramIdent	in	Identifier of *Int* or *Real* types - parameter for some types of functions.
locVarColValueIdent_Rec	out	Reference to a column of a structured variable of *Record* type - result values.
statusIdent_Int	out	Calculation (action) success.
archivInstance_Int	in	Optional identifier of *Int* type - identification of archive instance. If the parameter is not defined, the value 0 will replace it.

Description

The action executes the calculation of the statistical function specified by the *statFuncIdent_Int* parameter on the values represented by the *archIdent* historical value for the following time intervals specified by the *bt_TmA*, *et_TmA*, *step* and *deepTime* parameters:

- first time interval - *<bt_TmA - deepTime; bt_TmA>*
- second time interval - *<bt_TmA + step - deepTime; bt_TmA + step>*
- .....
- n-th time interval - *<bt + (N-1)*step - deepTime; bt + (N-1)*step>*

The timestamp of the result for a certain time interval is identical to the end time of the time interval. After the action is executed, calculation results are to be filled into the given column of the local variable specified by the *locVarColValueIdent_Rec* parameter.

Note: When calculating **CALCSTATFUNCARR** action, a time limit that is active in the computations of statistical archives and does not allow for calculation for future times, is not applied. With **CALCSTATFUNCARR**, it is, therefore, possible, for example, to calculate statistics over a script-filled archive that has data in the future.

The *statusIdent_Int* parameter gets one of the following values:

- _ERR_TRANS_ABORT
- _ERR_TRANS_ERROR
- _ERR_TRANS_IGNORED
- _ERR_NO_ERROR

The set of implemented functions is identical with the functions, which may be configured for statistical archive (except the *FILTER* and *TIMESLICE* functions). For individual functions, there are predefined loca l constants in ESL.

The *validPerctIdent_Int* parameter represents the Validation criteria used during the evaluation of the statistical function. It accepts values within 0 .. 100 (the error *ERR_RANGE_ERROR* occurs if the condition is not met).

For some of the statistical archive functions, the *paramIdent* parameter must be specified according to the table:

Statistical function	Parameter description
_STAT_F_INTEGRAL	Integral time units. For individual types, there are also established predefined local variables according to the table: **Constant** / **Description** _INTEGRAL_HOURUNIT / Hour integral _INTEGRAL_MINUNIT / Minute integral _INTEGRAL_SECUNIT / Second integral
_STAT_F_GE_TIME _STAT_F_GT_TIME _STAT_F_LE_TIME _STAT_F_LT_TIME _STAT_F_ADDITION_PARAM _STAT_F_INCREMENT_PARAM _STAT_F_DELTA_PARAM	Compare value

For other statistical archive functions, the *paramIdent* parameter is not evaluated.

Value of parameter *archivInstance_Int* defines the instance of the archive which executes the request. If the parameter is not defined (or the value is 0), the active archive instance will execute the request.

Example

Calculation of the weighted floating average of the values within the range of 2 hours.

```
 TIME _bt
 TIME _et
 INT _step
 INT _deepTime
 REAL _retCode
 INT  _idx
 RECORD NOALIAS (SD.Data) _statData

 _bt := %StrToTime("0:00:00 16-11-2006")
 _et := %StrToTime("0:00:00 17-11-2006")
 _step := 3600         ; value for each hour
 _deepTime := 2*_step  ; floating average, time depth of 2 hours

 CALCSTATFUNCARR H.Col[1], _bt, _et, _step, _deepTime, _STAT_F_WGAVG, 100,
0, _statData^Value, _retCode
 IF _retCode = _ERR_NO_ERROR THEN
   ; calculation done, browsing result values
      FOR _idx=1 TO _statData\DIM DO_LOOP
     ; _statData[_idx]^Value - value
 ELSE
   ; an error occurred
 ENDIF
```

Note

Using the **CALCSTATFUNC** action does not allow evaluating of the *FILTER* and *TIMESLICE* statistical functions.

Related pages:

Script actions