CALCARCHEXPR

CALCARCHEXPR action

Function

The action performs a calculation of specific archival expression over archival values. Archival values are presented through the names of archival objects in the expression.

Declaration

```
CALCARCHEXPR archExprIdent_Txt, btIdent_TimA, etIdent_TimA, stepIdent_Int,
_ignoreInvalid_Bool, _locVarColValueIdent_Rec, statusIdent_Int[,
archivInstance_Int]
```

Parameters

archExprIdent_Txt	in	Value identifier of Text type containing the expression.
btIdent_TimA	in	Identifier of AbsTime type - interval beginning.
etIdent_TimA	in	Identifier of AbsTime type - interval end.
stepIdent_Int	in	Identifier of Int type - time step [s].
_ignoreInvalid_Bool	in	Identifier of Bool type - invalid values in expression will be replaced by 0.
_locVarColValueIdent_Rec	out	Reference to a local variable column of RECORD type - final values.
statusIdent_Int	out	Identifier of Int type - the success of calculation.
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 performs a calculation of specific archival expression over archival values. Archival values are presented through the names of archival objects in the expression.

The rules of this calculation are identical to the calculation rules mentioned in definition of calculated archival object.

If a structured archive is used in an expression, a zero [0] number of row cannot be used.

The calculation is performed for entered time period <*btIdent_TimA*, *etIdent_TimA*>.

If parameter *stepIdent_Int*
= 0 - it means the calculation for change => each change some of the source data (archived) causes the entered expression recount and relevant value will be in result. **0** - it means the periodic calculation => the entered expression is evaluated in periods *btIdent_TimA*, *btIdent_TimA*+1**stepIdent_Int*, *btIdent_TimA*+2**stepIdent_Int* .... Resultant value array is written into entered column of local structured variable *_locVarColValueIdent_Rec* (variable can be dimensioned again).

Note

1. The expression must contain the name at least one of archive value.
2. If the expression contains several archive values, they all must have one parent in terms of DODM.

Example

```
 TIME _bt, _et
 INT _step
 INT _retCode
 TEXT  _expr
 TEXT _err
 BOOL _bIgnoreInvalid
 RECORD NOALIAS (SD.Data) _data

 _bt := %StrToTime("00:00:00 01-01-2008")
 _et := %StrToTime("00:00:00 02-01-2008")
 _step := 1
 _bIgnoreInvalid :=  @TRUE
 _expr :=  "H.a + H.GetArchRowCol_Ref[2]^I2 + H.GetArchRowCol_Ref_Col_I1
[2]"

 CALCARCHEXPR _expr, _bt, _et, _step, _bIgnoreInvalid, _data^Value,
_retCode

 IF _retCode =  _ERR_NO_ERROR THEN
   ; processing of result
 ELSE
   _err :=  _ERR_MSG ; extended error description
 ENDIF
```

Owing to the expression parsing works during system running (not during configuration) and if any errors occur, it is possible to obtain their detailed description as value of predefined local variable of Text type *_ERR_MSG*.

Related pages:

Script actions