GETARCHROW

GETARCHROW action

Function

Bulk read of archive values.

Declaration

```
 GETARCHROW archIdent, locRecDstIdent, timeFromIdent_TmA, timeToIdent_TmA,
stepIdent_Int, maxValsIdent_Int, statusIdent_Int [,archivInstance_Int]
```

Parameters

archIdent	in	Reference to row of: • object of Structured variable type, • local variable of *Record* type, • structured historical value.
locRecDstIdent	out	Local variable of *Record* type - read result.
timeFromIdent_TmA	in	Identifier of *Absolute time* type - interval beginning.
timeToIdent_TmA	in	Identifier of *Absolute time* type - interval end.
stepIdent_Int	in	Identifier of *Int* type - time step for archive values oversampling.
maxValsIdent_Int	in	Maximum number of values. If given interval contains more data, the action will trim off the data and return ERR_MORE_DATA warning in the parameter *statusIdent_Int*.
statusIdent_Int	out	Read 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 reads the values of several historical values. These are specified by the row given in the parameter *archIdent*. For each item in given row, the action internally generates **archive data read requests** and also waits for read data. By generating all read requests at once (next request doesn't wait until previous request is done) the action provides the increase of read speed by 50% to 200% than reading of the same number of values by the action GETARCHARR. This speed up depends on the system load and the number of columns in the row. The speed is higher when the system load is higher and the action reads more items.

Archive data read request contains reference to data by means of:

1. Object of Historical value type (or item of structured historical value) - in this case, the object that is to be read is uniquely determined.
2. Other system object (I/O tag, user variable, ...) or item of local variable variable of *Record* type - in this case, the system automatically search for the historical value that archives the object determined by the request (e.g. if the request contains the reference to I/O tag that has been archived, the system automatically redirects the request to the historical value). If such historical value doesn't exist, the action will be terminated and return the _ERR_ARCHIV_NOT_RUNNING error.

The parameter *archIdent* can be defined by one of the following ways:

1. **Row of local variable of „typed ALIAS" type**
   Example:
   ALIAS (structure definition) _lAT
   SET _lAT AS SV.Struct
   GETARCHROW _lAT[4] , ...

   Individual read requests are ALWAYS addressed to items in given column of the structure *SV.Struct* (even if the column contains items of *Object* type).

2. **Row of object of *Structured variable* type**
   Example:
   GETARCHROW SV.Struct[4] , ...

   Request addressing is the same as described in the article 1.

3. **Row of local variable of *Record* type**
   Example:
   RECORD (structure definition) _lRec
   REDIM _lRec[10]

   SET _lRec[4]^Item1 AS Sec
   SET _lRec[4]^Item2 AS SysTime
   SET _lRec[4]^Item3 AS M.MeranyBod
   .....

   GETARCHROW _lRec[4], ...

   All the columns of the structure definition must be of *Object* type. Individual read requests are to be addressed to the objects the row items refer to.
   If any item does not refer to object, the action is to be terminated and returns the ERR_NO_ASSIGNED_ALIAS error.

4. **Row of structured historical value**
   Example:
   GETARCHROW H.Struct[4], ...

   Individual read requests are to be gradually addressed to all items in given row. The object *H. Struct* cannot be a one-column historical value.

Read result is stored in the local variable *_locRecDstIdent*. Result of every request is either one value or a sequence of values. Every sequence of values is arranged by time in ascending order and stored row by row in the respective column of the local variable *_locRecDstIdent*. The structure type of the structured variables *archIdent* and *_locRecDstIdent* must be the same (it ensures the same number of columns). The action automatically resizes the variable *_locRecDstIdent* to required number of rows (the number is given by the largest number of values in the read result).

The parameters *timeFromIdent_TmA* and *timeToIdent_TmA* specify the time interval for reading values. The parameter *stepIdent_Int* defines the oversampling (in seconds) of read values. If it is equal to 0, reading is not to be oversampled. Therefore, if the value is differing from 0, the times of all values in the result structure is the same and every column contains read values up to the last row. If the value is 0, the size of the result structure is given by the maximum number of values read by one request. Therefore, not all columns will contains values read from the archive up to the last row.
If the parameter *timeToIdent_TmA* is higher than actual time and the parameter *stepIdent_Int* is differ from zero, oversampled values with future timestamps will be invalid.

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

The variable *statusIdent_Int* indicates the success of reading the action. If this variable acquires the value _ERR_NO_DATA(22), it means that all archives, which were used when reading, do not contain any data. If at least one of the archives contains data, the action returns _ERR_NO_ERR(0).

Example	GETARCH* actions - example.

Related pages:

Script actions