

GETARCHCOL

GETARCHCOL action

Function

Bulk read of archive values.

Declaration

```
GETARCHCOL archIdent, locRecDstIdent, timeFromIdent_TmA, timeToIdent_TmA,
stepIdent_Int, maxValsIdent_Int, statusIdent_Int [,archivInstance_Int]
[TIMEBYROW | TIMEBYCOL]
```

Parameters

archIdent	in	Reference to an item of: <ul style="list-style-type: none">the object of Structured variable type,the local variable of <i>Record</i> type,the structured historical value, or reference to one-column historical value .
locRecDstIdent	output	The local structured variable for a read result.
timeFromIdent_TmA	in	An Identifier of the <i>Absolute time</i> type - interval beginning.
timeToIdent_TmA	in	An Identifier of the <i>Absolute time</i> type - interval end.
stepIdent_Int	in	An Identifier of the <i>Int</i> type - time step for archive values resampling. For resampling details, see Resampling note .
maxValsIdent_Int	in	The maximum number of values. If the given interval contains more data, the action will trim off the data and return ERR_MORE_DATA warning in the parameter <i>statusIdent_Int</i> .
statusIdent_Int	output	Read success.
TIMEBYROW		Places result values by time into rows.
TIMEBYCOL		Places result values by time into columns.
archivInstance_Int	in	Optional identifier of <i>Int</i> type - identification of archive instance . If the parameter is not defined, the value 0 will replace it.

Description

According to the parameter *archIdent*, the action (analogous to the action [GETARCHROW](#)) generates more requests for archive reading and synchronously waits for their completion. The item defined by *archIdent* specifies the row that is to be used for archive data requests.

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

1. A column of a local variable of the „[typed ALIAS](#)“ type

Example:

```
ALIAS (structure definition) _IAT
SET _IAT AS SV.Struct
GETARCHCOL _IAT[0]^Item1 , ...
```

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

2. A column of an object of the *Structured variable* type

Example:

```
GETARCHCOL SV.Struct[0]^Item2 , ...
```

Request addressing is the same as described in article 1.

3. A column of a local variable of the *Record* type

Example:

```
RECORD (structure definition) _IRec
REDIM _IRec[3]
```

```
SET _IRec[1]^Item2 AS Sec
```

```
SET _IRec[2]^Item2 AS SysTime
SET _IRec[3]^Item2 AS M.MeranyBod
.....

GETARCHCOL _IRec[0]^Item2, ...
```

The column must be of the *Object* type. Individual read requests are to be addressed to the objects the column items refer to.
If any item doesn't refer to object, the action is to be terminated and returns the `ERR_NO_ASSIGNED_ALIAS` error.

4. A column of structured historical value

Example:

```
GETARCHCOL H.Struct[0]^Item2, ...
```

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

5. One-column historical value

Example:

```
GETARCHCOL H.Struct, ...
```

Individual read requests are to be gradually addressed to all items.

The read result is stored in the local structured variable defined by the parameter `_locRecDstIdent`. The result of every request is either one value or a sequence of values. Every sequence of values is stored in the local variable `_locRecDstIdent`:

- **in rows** (analogous to the action [GETARCHROW](#)) - the first column contains the values obtained by the read request of the first row (respective column), the second column contains the values obtained by the read request of the first row, Structured variable `_locRecDstIdent` must be of correct size. If not, the action automatically resizes it to the required size,
- **in columns** - every sequence of values is written into one row of the local variable `_locRecDstIdent`. If some of the sequences contain more values than the number of columns, the variable `statusIdent_Int` gets the value of `_ERR_MORE_DATA`, but the values that could be written will be inserted into the local variable `_locRecDstIdent`.

The 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 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](#)