

GETOLDVAL

GETOLDVAL action

Function

Declaration

Reading of values from the archive of the substation.

GETOLDVAL statIdent, fromTimeIdent_TmA, endTimeIdent_TmA, retIdent_Int, archIdent [MASK mask_txt]

or

GETOLDVAL statIdent, fromTimeIdent_TmA, endTimeIdent_TmA, retIdent_Int NOWAIT [MASK mask_txt]

Parameters

statIdent	in	Reference to a Station type object.
fromTimeIdent_TmA	in	Identifier of the AbsTime type - begin time for reading from the archive.
endTimeIdent_TmA	in	Identifier of the AbsTime type - end time for reading from the archive.
retIdent_Int	output	Return code identifier of the Int type.
archIdent	in	Reference to a Process type object (archive).
NOWAIT	in	Keyword. If it is not entered, the action waits synchronously for the termination of data reading.
MASK	in	Keyword.
mask_txt	in	Identifier of Text type - text string that consists of a mask formed from the names of I/O tags for which the values are to be read.

Description

The action will start the backward reading of values from the archive of a station. Time interval is given by the identifiers fromTimeIdent_TmA and endTimeIdent_TmA. Return code (retIdent_Int) gains one of the following values:

- _ERR_TRANS_ABORT
- _ERR_TRANS_ERROR
- _ERR_TRANS_IGNORED
- _ERR_NO_ERROR

In case of the first declaration, the action waits for the termination of the reading and archiving of values, if these are archived. In this case it is necessary to archive all I/O tags in one archive. An archive is represented by the identifier archIdent. The second declaration only starts the backward reading of values and doesn't wait for its termination. Return values is to be always _ERR_NO_ERROR.

If the keyword MASK and identifier mask_txt are defined, the action reads values only for those I/O tags that match the definition, which is entered in the identifier mask_txt. For example mask_txt := "M. RIADENIE*".

Note

If backward reading of communication station values is being performed, the flag A is set for the communication station value.

 Related pages:

[Script actions](#)