SET WITH

SET WITH action

Function	Changes several values at the same time (always in connection with structures).
Declaration	`SET dstIdent_Rec WITH srcIdent_Rec`
Parameters	dstIdent_Rec — in — Values' destination (row identifier or whole structure identifier). srctIdent_Rec — in — Values' source (row identifier, or whole structure identifier).
Description	The structure types of the parameters `dstIdent_Rec` and `srcIdent_Rec` must be always identical (otherwise the action generates the error *_ERR_RECORD_NO_COMP*). The same applies to their size. Their type may be: • RECORD, • ALIAS (typed), • An object of the *Structured variable* type. The action sets the values of items from `srcIdent_Rec` to the values of items from `dstIdent_Rec`. If `dstIdent_Rec` is the reference to a row, then `srcIdent_Rec` must be the reference to a row, too. Accordingly if `dstIdent_Rec` is a reference to a whole value, then `srcIdent_Rec` must be a reference to a whole value, too. Therefore, the combination: RECORD (SD.RecordDef) _lArr1 SET _lArr1[1] WITH SV.Structure is not permitted and the script editor reports an error during the compilation. When setting a whole value, the number of rows must be identical. For example: RECORD (SD.RecordDef) _lArr1 SET _lArr1 WITH SV.Structure The action **SET WITH** will generate the error `_ERR_RANGE_ERROR` (the size of `SV.Structure` is 10 rows and the size of the local variable `_lArr1` after the declaration is just 1 row). Correct copying the whole value into the local variable is as follows: RECORD (SD.RecordDef) _lArr1 REDIM _lArr1[SV.Structure\DIM] SET _lArr1 WITH SV.Structure When copying values to a local variable of RECORD type, which contains items of Object type, the action sets values of the objects, the individual adjusted items are pointed to. This effect is not always desirable. Therefore it is possible to use the NOALIAS modifier when you declare a local variable of RECORD type as follows: RECORD NOALIAS (SD.RecordDef) _lArr1 For a local variable declared in this way, the pointing feature for all items of Object type is disabled. Their type is not predefined. A value of any type may be assigned to them. For example: _lArr1[1]^Object := 1 _lArr1[1]^Object := "Text" See also: • Script local variables • Permissible combinations of parameters • **SET AS** action

Example	RECORD (SD.RecordDef) _lArr1 RECORD (SD.RecordDef) _lArr2 INT _index REDIM _lArr1[5] REDIM _lArr2[5] ;... initialization of values to the local variable lArr1 ; copying values of the 3rd row from _lArr1 into the 2nd row of _lArr2 SET _lArr2[2] WITH _lArr1[3] ; copying the whole value using a loop ; it assumes the same array sizes _index := 1 NextRow: IF _index <= _lArr1\DIM THEN SET _lArr2[_index] WITH _lArr1[_index] _index := _index + 1 GOTO NextRow ENDIF ; The same result may be achieved by using SET WITH action SET _lArr2 WITH _lArr1

Related pages:

Script actions