

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). |
| | srcIdent_Rec | in | Values' source (row identifier, or whole structure identifier). |
| Description | <p>The structure types of the parameters <code>dstIdent_Rec</code> and <code>srcIdent_Rec</code> must be always identical (otherwise the action generates the error <code>_ERR_RECORD_NO_COMP</code>). The same applies to their size. Their type may be:</p> <ul style="list-style-type: none">RECORD,ALIAS (typed),Object of <i>Structured variable</i> type. <p>The action sets the values of items from <code>srcIdent_Rec</code> to the values of items from <code>dstIdent_Rec</code>. If <code>dstIdent_Rec</code> is the reference to a row, then <code>srcIdent_Rec</code> must be the reference to a row, too. Accordingly, if <code>dstIdent_Rec</code> is the reference to a whole value, then <code>srcIdent_Rec</code> must be a reference to a whole value, too.</p> <p>Therefore, the combination:</p> <pre>RECORD (SD.RecordDef) _IArr1 SET _IArr1[1] WITH SV.Structure</pre> <p>is not permitted and the script editor reports an error during the compilation.</p> <p>When setting a whole value, the number of rows must be identical. For example:</p> <pre>RECORD (SD.RecordDef) _IArr1 SET _IArr1 WITH SV.Structure</pre> <p>The action SET WITH will generate the error <code>_ERR_RANGE_ERROR</code> (the size of <code>SV.Structure</code> is 10 rows and the size of the local variable <code>_IArr1</code> after the declaration is just 1 row). Correct copying the whole value into the local variable is as follows:</p> <pre>RECORD (SD.RecordDef) _IArr1 REDIM _IArr1[SV.Structure\DIM] SET _IArr1 WITH SV.Structure</pre> <p>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:</p> <pre>RECORD NOALIAS (SD.RecordDef) _IArr1</pre> <p>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:</p> <pre>_IArr1[1]^Object := 1 _IArr1[1]^Object := "Text"</pre> <p>See also:</p> <ul style="list-style-type: none">Script local variablesPermissible combinations of parametersSET AS action | | |
| Example | <pre>RECORD (SD.RecordDef) _IArr1 RECORD (SD.RecordDef) _IArr2 INT _index REDIM _IArr1[5] REDIM _IArr2[5] ;... initialization of values to the local variable IArr1 ; copying values of the 3rd row from _IArr1 into the 2nd row of _IArr2 SET _IArr2[2] WITH _IArr1[3] ; copying the whole value using a loop ; it assumes the same array sizes _index := 1 NextRow: IF _index <= _IArr1\DIM THEN SET _IArr2[_index] WITH _IArr1[_index] _index := _index + 1 GOTO NextRow ENDIF ; The same result may be achieved by using SET WITH action SET _IArr2 WITH _IArr1</pre> | | |



Related pages:

[Script actions](#)