Assignment

Function

Assignment (or conditional assignment) of one value.

Declaration

```
dstIdent  := expression [TIME  timeExpression]
dstIdent  :?= expression
```

Parameters

dstIdent	in	Values destination (row identifier or whole structure identifier).
expression	in	Expression defining a value.
TIME timeExpression	in	Expression (AbsTime) defining the time of the value occurrence (optional parameter).

Description

The time of a value created by an assignment is the current time. If the parameter

```
TIME
```

is defined, then the value of the parameter `timeExpression` will be used for the time of when the value is generated.

`dstIdent` is the identifier of one value of:

- object or local variable of *INT*, *BOOL*, *REAL*, *TIME,* and *TEXT* types.
  The parameter *`expression`* must be given type. If not, the script will attempt to convert the value into the correct type. If it is not successful, then the result value will be invalid.

- a local variable of *RECORD* type or object of Structured variable type.
  You can carry out an assignment just to one item of the particular structure row. Other assignment rules are compliant with *simple type*.

```
RECORD (SD.RecordDef) _lArr
 REDIM _lArr[10]
 _lArr[2]^Int := 1
```

- local variable of *ALIAS* type (untyped).
  If it is not assigned to a D2000 system object, the error _ERR_NO_ASSIGNED_ALIAS will occur. Otherwise, the expression is evaluated and the result value is assigned to the object with the local variable assigned.

```
ALIAS _a
SET _a AS U.Int
_a := 1
WAIT
```

- local variable of ALIAS type (typed)
  You can carry out an assignment just to one item of the particular structure row. If the local variable is not linked to an D2000 system object, the error _ERR_NO_ASSIGNED_ALIAS will occur. Otherwise, the expression is evaluated and the result value is assigned to the object item.

```
ALIAS (SD.RecordDef) _aArr
SET _aArr AS SV.Structure
_aArr[2]^Int := 1
WAIT
```

Conditional assignment :?=

In a conditional assignment, the expression is evaluated first, and the resulting value is compared to the current *dstIden* value.
If the values are different, the assignment is performed. If the values are not different, the assignment will not be performed.

When comparing values (if both values are valid), only the value without other attributes, such as the time of value creation is taken into account,
The comparison considers two invalid values to be the same.

Conditional assignment makes sense especially when changing the values of objects within DODM, which prevents unnecessary activities that follow the assignment.
For example:

- when writing to the output I/O tag, communication takes place at the protocol level
- when writing to a user variable or to an item of a structured variable that has an active entry of a value change in the configuration database

A conditional assignment is the equivalent of the following entry

```
IF M.Output # _newValue THEN
  M.Output := _newValue
ENDIF
```

Note

If an assignment changes the value of a D2000 system object, in principle it is just a "request" for changing the value. After the assignment is executed, the script continues in the execution of the next actions and there is no warranty, that the object value has been changed (it depends on the current situation - the system load).
Example:

```
INT _prevValue

_prevValue := U.Int             ; assign previous value
U.Int := U.Int + 1              ; add the value of 1 to object value
IF U.Int = _prevValue THEN      ; check the value
  ; the assignment has not been executed
ELSE
  ; the assignment has been executed
ENDIF
```

The case that assignment is not still executed, is more probably. WAIT action allows waiting for the assignment execution (its completion).

Indexed variable in an active picture

Assignment to an indexed variable in the active picture script environment.

Related pages:

Script actions