

# EXPORT\_CSV\_TEXT

## EXPORT\_CSV\_TEXT action

### Function

Export of a row, column or whole [structured local variable](#) (i.e. local variable of *Record* type) into a string in CSV format.

### Declaration

```
EXPORT_CSV_TEXT toExportStruct, delimiter, outputIdent_Text, timeZone,
retCodeIdent_Int[, transPalIdx_Int] [TIME]
```

### Parameters

<b>toExportStruct</b>	in	<a href="#">Row identifier</a> , <a href="#">column identifier</a> or <a href="#">identifier</a> of local variable of <i>Record</i> type.
<b>delimiter</b>	in	Identifier of <i>Text</i> type - delimiter to be used in the CSV file.
<b>timeZone</b>	in	Parameter of <i>Text</i> , <i>Int</i> or <i>Bool</i> type - defines interpretation of timestamps.
<b>retCodeIdent_Int</b>	out	<a href="#">Return value</a> of <i>Int</i> type - action success.
<b>transPalIdx</b>	in	Optional parameter of <i>Int</i> type.
<b>outputIdent_Text</b>	out	Export result of <i>Text</i> type
<b>TIME</b>	in	Keyword.

### Description

The action generates a string in CSV format. Item delimiter in the CSV string is specified by the parameter *delimiter*.

The action success is indicated by the output parameter *retCodeIdent\_Int*. The value of 0 means a successful export, values other than 0 are an error.

Depending on the parameter *toExportStruct*, the action exports:

- the entire local variable
- the row of local variable
- the column of local variable

#### CSV format

The local variable is saved to the specified CSV row-by-row (row in a structure presents a row in a CSV). Because the CSV format is not unique defined, the action awaits a delimiter as its second parameter. Delimiter is one character, that is to be used to separate items in the CSV string. Real numbers use the decimal separator '.'.

If the keyword

```
TIME
```

is used, then the CSV string also contains the value time for each item. Time is presented before the item value. CSV string therefore contains the double number of columns.

Time can be formatted by the mask defined in the [transformation palette](#) - the index is given by parameter *transPalIdx\_Int*.

If the parameter is not defined or its value is 0, there is used the mask *dd.mm.rrrr hh:mi:ss*.

If its value is -1, then values of *Absolute time* and *Real* types are to be converted according to the settings specified in the **Regional and Language Options** of the current user on Windows systems. For different operating systems the value of -1 will be automatically replaced with the value of 0.

The parameter *timeZone* defines export interpretation of timestamps. In case of empty text, timestamps are exported as local times. In case of defined time zone name (e.g. "Europe/London"), timestamps are exported as local times for given time zone. In case of integer value, timestamps are exported as times with fixed offset from UTC, where given value defines offset in seconds. For historical reasons boolean values are accepted: @FALSE - timestamps are exported as local times, @TRUE - timestamps are exported as times with fixed offset of 3600 seconds (1 hour) from UTC. Usage of boolean value @TRUE is not recommended and generates warning when ESL script is being saved. It is recommended to replace it with integer value.

The CSV delimiter (the parameter *delimiter*) may be obtained by calling the function [%GetCSVDelimiter](#).

### Example

Assumes:

The object SD.ArchVal of *Structure definition* is defined with the columns:

Name	Type
------	------

Value	Real
Flags	Integer

There is defined the object SV.ArchVal of *Structured variable*. The structure type is SD.ArchVal. The number of rows: 3. Start values are:

Value	Flags
1.2	3
2	2
3	1

The example does not test the action execution success.

```
RECORD (SD.ArchVal) _rec

REDIM _rec[SV.ArchVal\DIM]
SET _rec WITH SV.ArchVal

EXPORT_CSV _rec, ';', _output1, @TRUE, _retCode
EXPORT_CSV _rec[1], ';', _output2, @TRUE, _retCode
EXPORT_CSV _rec^Value, ';', _output3, @TRUE, _retCode
```

The actions exports the local variable `_rec`, that contains a copy of `SV.ArchVal`.  
Content of variables:

`_output1`

```
1,2000000000000000;3
2,0000000000000000;2
3,0000000000000000;1
```

`_output2`

```
1,2000000000000000;3
```

`_output3`

```
1,2000000000000000
2,0000000000000000
3,0000000000000000
```



**Related pages:**

[Script actions](#)