

# MACRO

MACRO runs a script (or "macro") in scripting enabled software like AutoCAD, Rhinoceros, MatLab, Word, Excel, etc.

## Syntax

MACRO(ProgramName\$, (Script)FileName\$, Macro\$, OutputFile\$/LocationLabel\$, [ContinueCaption\$])

### Arguments

- ProgramName\$ is a string expression containing the name of the software to be used. Present programs tested:
  - MSWord (also see [WINWORDS\(\)](#) function)
  - MSEXCEL (also see [EXCEL#\(\)](#) function)
  - MSVISIO
  - Rhinoceros (use Rhino3, Rhino4)
  - AutoCAD (use AutoCAD16, AutoCAD17)
  - MatLab
- (Script)FileName\$ is a string expression containing the name of (drawing) file or the file in which the macro script has to be saved.
- Macro\$ is a string expression containing the macro script, either VBA or any other supported script.
- OutputFile\$ is a String expression containing the name of an output file created by the script, if any. This output file can e.g. contain a set of values computed by the script in the software used.
- ContinueCaption\$ is an optional string argument containing the caption of the modal continue button that is shown after posting the script to the connected software. In the case of a CAD system or drawing program, this allows you to make changes to the result generated by the script before allowing the system to continue the session.

## Remarks

1. To create a script for the function, usually you can record a series of actions with something like a macro recorder or discribed syntax of the program you want to script to be used in. After creating the script you copy it as [TEXTITEMS\(\)](#) into the data slot of a parameter and you use the [TEMPLATES\(\)](#) function to fill the script with your parameters to integrate and use it dynamically into your knowledge base.
2. If OutputFile\$="NullString", the MACRO() function will return a string containing "ProgramName\$ script completed".
3. In Visio, Word and Excel the posted macro is immediately removed from the document after it has been executed.
4. In addition to the programs mentioned Qnowledge is always willing to investigate implementation of the same functionality for programs not yet mentioned above

## Examples

### Microsoft VISIO

The relation

CompID = MACRO("Visio", Diagram\$, CompMacro\$, "VisioMacro.epo", "Move object to final location")

either creates a drawing with name *Diagram\$* and posts the macro *CompIDMacro\$* to Visio, or simply posts the macro *CompMacro\$* to Visio if Visio already has a drawing with name *Diagram\$* open. In this case, in the macro an index of the newly created component is saved in the file VisioMacro.epo (on the current default solution/project directory). *CompMacro\$* is a VBA macro that is typically created with the [TEMPLATES\(\)](#) function.

As soon as the macro is executed, the modal continue button will appear with the message "Move object to final location" which gives the user the opportunity to move the new object manually to a desired location. If *ContinueCaption\$* is omitted in the MACRO() function, the session continues without further notice.

### RHINO version 3

The relation

VolComp = MACRO("Rhino3", MacroName\$, CompMacro\$, "RhinoMacro.epo")

saves the Rhino script *CompMacro\$* in the file *MacroName\$* and passes the file to Rhino for execution. *CompMacro\$* is a VBA macro that is typically created with the [TEMPLATES\(\)](#) function. In this case, the macro not only places the component in the drawing but also calculates the volume of the newly created component and saves it in the file *RhinoMacro.epo* (on the current default solution/project directory). Since the *ContinueCaption\$* is omitted in this MACRO() function, the session continues without further notice.

### RHINO version 4

The only difference between MACRO("Rhino3",...) and MACRO("Rhino4",...) is that the first uses a single running instance of Rhino and the latter makes a new instance of rhino each time the function is executed (see Rhino Wiki). If you want to execute a script by a single running instance of Rhino, please use the "Rhino4.Interface" argument instead.

## Microsoft WORDe

The relation

WText\$ = MACRO("Word", WordDocument\$, CompMacro\$, LocationLabel\$, "Select location for including item...")

either creates a document with name *WordDocument\$* and posts the macro *CompMacro\$* to word or simply posts the macro *CompMacro\$* to Word if Word already has a document with name *WordDocument\$* open. *CompMacro\$* is a VBA macro that is typically created with the [TEMPLATE\\$\(\)](#) function.

BEFORE executing the macro, the modal continue button will appear with the message "Select location for including item..." which gives the user the opportunity to manually select a location in the document on which to perform the macro (typically the inclusion of a picture or dedicated piece of text. If ContinueCaption\$ is omitted in the MACRO() function or has the value "NullString", the macro will be performed on the document on the location at which *LocationLabel\$* occurs, after which the session continues without further notice.

Please note that you can use this functionality in addition to the [WINWORD\\$\(\)](#) function.

## Microsoft EXCEL

The relation

EText\$ = MACRO("Excel", ExcelSheet\$, ExcelMacro\$, "ExcelMacro.epo", "Press OK to Continue")

either creates a spreadsheet with name *ExcelSheet\$* and posts the macro *ExcelMacro\$* to Excel or simply posts the macro *ExcelMacro\$* to Excel if a sheet with name *ExcelSheet\$* is already open. *ExcelMacro\$* is a VBA macro that is typically created with the [TEMPLATE\\$\(\)](#) function (you can create a macro.

AFTER executing the macro, the modal continue button will appear with the message "Press OK to Continue" which gives the user to check the result of whatever macro has been run. If ContinueCaption\$ is omitted in the MACRO() function or has the value "NullString", the macro will be performed on the sheet, after which the session continues without further notice.

Please note that you can use this functionality in addition to the [EXCEL#](#) function.

## Mathworks MatLab

Documentation will be follow soon... In the mean time please contact Qknowledge for further details.

---

Quick links: [Functions overview](#) | [Attribute overview](#) | [Constants overview](#) | [Dimensions overview](#)