AUTODESK UNIVERSITY

GEN21552

Macro and Cheese: An Automation Feast

Volker Cocco

Autodesk, Inc.

Learning Objectives

- What macros are and when should one use them
- Understand macro syntax
- Learn how to build a macro
- Learn how to add macro commands to a tool palette

Description

AutoCAD software users tend to repeat the same task frequently. Something as simple as placing text may require that one sets a specific layer current, then sets a text style current, and then sets the height of the text. This example requires 3 steps before typing the verbiage. Other repetitive tasks can consume even more time. Creating a simple macro will automate processes such as this into one function. Macros can also assist in ensuring that drafting standards are being used. This session will demonstrate how to easily create macros to automate repetitive tasks and how one can add them to a tool palette for easy access. This session features AutoCAD and AutoCAD LT.

Your AU Expert

Volker Cocco:

Employed at Autodesk, Inc., as a technical support specialist and AutoCAD KDE (Knowledge Domain Expert), Volker Cocco has been working with AutoCAD software since 1991. He has worked for various Autodesk Resellers since 1997 and he has had extensive experience troubleshooting and supporting Autodesk products. With a background in CAD drafting and management, Volker has instructed basic to advanced AutoCAD technical classes including sessions at Autodesk User Group International (AUGI) CAD Camp and Autodesk University.

What macros are and when should one use them

Macros - an instruction that represents a sequence of instructions in abbreviated form.
- Dictionary.com

One of AutoCAD's greatest assets is that one can customize the application to fit ones needs.
Macros provide a method for customizing AutoCAD and AutoCAD LT (as well as AutoCAD platform vertical applications) without having to learn a programming language. Basically…, a macro is nothing more than a series of commands and their options written into a string of text for repeated use. However, these command macros can contain commands, special characters, DIESEL expressions, or AutoLISP programming code. Macros can be as simple as calling a single command, setting a system variable, or contain a complex series of commands and settings.
Note that AutoCAD LT does not support AutoLISP.

Why and when should one use macros? The bottom line is that you don't have to, but if you want to maximize productivity and minimize redundant tasks then you should start thinking about automating your tasks using macros. My thoughts are that if I perform a task three times, it's time to automate that task.
Drafting standards are another biggie. When working with my drawing I don't want to have to try and remember what style of text I use for annotation text, what height, or what layer it's placed on. Instead, I'll create a command macro to do all the heavy lifting (thinking) for me. Making these macros available to everyone else working in my drafting department ensures that consistency is applied to a project.

Understand macro syntax

The key to writing macros is to have an understanding of commands, syntax and the rules for writing the macros.

In a nutshell, a macro is nothing more than the commands one would input as seen on the CLI (Command Line Interface). In addition to everyday commands used in AutoCAD, one needs to be aware of special characters used within Macros.

Character	Description
;	Equivalent to pressing ENTER
[Blank Space]	Equivalent to pressing ENTER or the SPACE BAR
****	Pauses for user input
_	Translates AutoCAD commands and options that follow (international support)
.	(Period) Ensures that if a command has been undefined, the command will work
-	Suppresses dialog - runs the command line version of the command
*****	Repeats macro until canceled by user
^	Equivalent to CTRL key (**^C** *would cancel a command*)
$	Introduces a DIESEL Expression
@	Specifies the use of the last referenced point picked in the drawing
‘	(Apostrophe) Issues a transparent command function

Stuff to remember

Active commands

When running a macro, the intent is to run a series of commands. In some cases one will want any active command to be canceled, in order to process the macro. In order to ensure that any active command is canceled, one would use ^C as the first function of the macro. The ^C cancels most commands, but some commands such as dimensioning, require ^C^C and commands such as –Layer (command line version of the Layer dialog) require ^C^C^C. Typically, ^C^C will do although my preference is ^C^C^C.

There are commands which one will typically not want to cancel; for example, the Zoom command is often used transparently (or interactively) within other commands. For such functions one would use the apostrophe (‘) at the beginning of the macro to allow the macro to run transparently within another command.

Macro Characters

A space is considered a character, and in a macro the space is the same as using the spacebar or ENTER key to end or repeat a command.

Use Standard Commands in Macros

Always! Many users will modify the acad.pgp by assigning different shortcuts for commands. Always use the full command when writing a macro. However, for the options within a command it’s ok to use the one or two characters that are used to invoke the command option.

Some users or CAD managers may undefined certain commands. In order to circumvent this place a period (.) in front of the command in order for the macro to run with as expected.

International support

When writing macros one can also add global support in order to use the macro on any of the different language installs which AutoCAD supports. This is fairly easy to do, precede the command used in the macro with an underscore character (_). This tells AutoCAD that the command should be translated prior to being executed.

Repeating a macro

One may wish to run the same macro several times in a row, to do this precede the command with the asterisk character (*). This will repeat the macro until one presses the ESC key or runs another macro to cancel the current one.

Ending a Macro sequence

As mentioned earlier, a space character is the same as pressing ENTER to end a command. Some functions such as the TEXT command require the use of the ENTER key to end the command, and some commands may require multiple ENTER or Spacebar entries to complete. To resolve this use a semicolon (;) to simulate the ENTER command. That said, if a macro string ends with a control character (^), a

backslash (\) for user input or a semicolon (;), the Space character is not added. In fact, adding a Space character would repeat the command.

Learn how to build a macro

So how does one get started? Well, the easiest way is to cheat! Yup. I said it. Seriously, use AutoCAD to help you out with this. You will have to use a bit of old school AutoCAD by working with the command line and the Text Windows (F2). I'd also recommend turning Dynamic Input off.

The bottom line is that macros (for the most part), consist of the same commands and options which one would use while drafting. If you know the commands, you can easily write a macro. If you don't know the commands, it's a great way to learn about options which you may never have used.

Decide what you want the macro to do by going through the steps in AutoCAD. Once the task is complete use the function key F2 to open the Text Window. There you will see a history of the steps you used. As an example, take a look at the following command process:

The AutoCAD command as shown on the command line

```
Command: REC
RECTANG
Specify first corner point or
[Chamfer/Elevation/Fillet/Thickness/Width]: 0,0
Specify other corner point or [Area/Dimensions/Rotation]:4,2
```

As a macro the input would appear as such:

```
^C^C^C_.rectang;0,0;4,2;
```

Once you have an understanding of the command prompts, writing the macro is fairly easy.

Sample Macros

The following sample macros cover various functions in AutoCAD and can easily be modified to suit different needs.

Start a new drawing and save it as "Start.dwg"

Run the Rectangle Macro from the first example. Save the drawing by running the following macro "**ZQS**" which will Zoom to the extents of the drawing, leaving some breathing room for the entities, followed by a quick save.

^C^C^C_.Zoom;E;;.9x;_QSAVE;

1. Cancels current command
2. Zoom command
3. Enter executes the command
4. Extents Option
5. Enter to Finish
6. Enter to Repeat Zoom
7. .9x Scale Option
8. Enter
9. QSAVE command

Next we'll modify some settings in the drawing file. For a visual as well as demonstration purpose, type OSNAP at the command prompt. Clear all the settings and close the dialog.

Next run the "**Settings**" macro which will disable SNAP, GRID, DYNAMIC INPUT and the NAVBAR:

^C^C^C_.SNAP;Off;_.GRID;Off;_.DYNMODE;0;_.NAVBAR;Off;

1. Transparent mode executed (in case this needs to be done within a command)
2. Snap command
3. Enter
4. Off option
5. Grid command
6. Enter
7. Off option
8. Enter
9. Dynmode (dynamic input) command
10. Enter
11. 0 = Off option
12. Enter
13. Navbar command
14. Enter
15. Off option
16. Enter

I make a point of using my OSNAPS, enabling those which I need at the time and disabling those I don't. I'll have several preset macros for that as well and I always run a macro for my

default OSNAPS. We can modify the OSNAP settings using the OSMODE system variable. Start the line command and select the top left corner of the rectangle. Note that the OSNAPS are off. Run the "**OS1**" macro to set the OSNAPS to ENDPOINT, MIDPOINT, CENTER and NODE. This macro runs transparently within another command.

'Osmode;15;

1. Transparent mode invoked (in case this needs to be done within a command)
2. OSMODE system variable
3. Enter
4. Set values to ENDPOINT, MIDPOINT, CENTER and NODE
5. Enter

Again, pick the top left corner of the rectangle and note that the OSNAPS are working. Finish the line command and type OSNAP at the command prompt. Review the settings and close the dialog.

Next we'll COPY and ROTATE the rectangle using the "**CRO**" Macro. Yes, I'm aware that there are many tools in AutoCAD which do the same function. Remember this is an example and it makes use of several special characters:

^C^C^C_.copy;\;_rotate;L;;@;

1. Cancels current command
2. Copy command
3. Enter executes the command
4. Wait for user input to select object (rectangle in this example)
5. Enter to end selection
6. Wait for user input to select base point
7. Wait for user input to select placement point
8. Rotate command
9. Enter executes the command
10. Selects the last object created
11. Enter to end selection
12. Previous (last referenced point) base point is prompted for selection
13. (last referenced point) is picked
14. Wait for user input for rotation angle

We began with a new drawing file, therefore we only have the defaults found in the "acad.dwt". Take a look to see which layers, linetypes and text styles exist. Also note that there are no blocks currently referenced in the drawing.

Next let's run the macro "LAYERP" which will create a layer called "Phones" with a color set to red and then set that layer current. Note the hyphen (-) which suppresses the layer dialog:

^C^C^C-Layer;m;PHONES;C;RED;PHONES;;

1. Cancel current command
2. Command line Layer command
3. Enter
4. Make a new layer and set current option
5. Enter

6. New Layer name
7. Enter
8. Specify layer color
9. Enter
10. Specify which layer to assign the color to
11. Enter
12. Enter to exit command

Now that we have the layer “Phones” in the drawing we can run a macro which will insert a block called “FNPHONE” on that layer. Had we not run the previous macro first, this macro would fail.

This macro requires that the folder containing the “FNPHONE.dwg” is in your AutoCAD support path or that the block already exists as a reference in the current file. This macro also repeats using the asterisk (*) as a prefix, until the user cancels (ends) the command:

***^C^C^C-Layer;s;PHONES;;-insert;FNPHONE;S;1;R;\;;**

1. Multiple mode
2. Cancel current command
3. Command line layer command
4. Enter
5. Set Layer current option
6. Enter
7. Specify layer to set current
8. Enter
9. Enter to finish command
10. Command line Insert command
11. Enter
12. Block name to insert (must reside in drawing, be in the support path, or have full path specified in quotes”)
13. Enter
14. Scale factor option
15. X value
16. Enter to accept X value as Y value
17. Rotation Option
18. Enter
19. Wait for user input to specify angle
20. Enter to accept angle (RMC to accept last used angle)
21. Pick insertion point
22. Repeat

So the previous two macros are all fine and dandy, but they can be improved by combining the sequences of commands. In the following example we will use the “FLOW” macro to demonstrate a similar sequence of events. In this example, a custom linetype called “Flow” (how appropriate, eh?) will be loaded from the “AU 2016.LIN” file and a new layer “AU 2016” is created with a color red and set current. Finally, the PLINE command is executed.

Note that in this macro we specifically point the linetype command to a linetype file using the full path. It’s important to note that instead of using back slashes (\), forward slashes (/) are used. Remember that back slashes are used to pause for user input in a macro. Also note that quotes are used to enclose a text string due to the spaces:

^C^C^C_-linetype;L;flow;"D:/Users/Coccov/Desktop/AU 2016/AU 2016.LIN";;_-LAYER;M;"AU 2016";C;RED;"AU 2016";L;FLOW;"AU 2016";;_PLINE;

1. Cancels the current command
2. -LINETYPE executes the Linetype command (suppressing the dialog)
3. ENTER
4. L to load linetype
5. ENTER
6. Loads the linetype “Flow” from “AU 2016.LIN”
7. ENTER
8. -LAYER executes Layer command (suppressing the dialog)
9. ENTER
10. Make (creates layer and sets current
11. “AU 2016”
12. ENTER
13. Color
14. ENTER
15. RED
16. ENTER
17. “AU 2016” (Ensures color is assigned to appropriate layer)
18. ENTER
19. L (sets linetype)
20. FLOW
21. ENTER
22. “AU 2016” (Ensures linetype is assigned to appropriate layer)
23. ENTER
24. ENTER to finish command
25. PLINE
26. ENTER

Yes. It is lengthy, but pretty cool, eh?

The final macro is one I use to clean my drawing and check for drawing database errors. I do this typically at the end of a project, but it’s sometimes necessary when receiving a file or after having introduced (inserted) blocks or other references. The macro “**CLEANUP**” will run the “**-PURGE**” command to remove all referenced entities and Registered Applications, followed by an AUDIT of the drawing file.

^C^C^C_-PURGE;A;;N;;R;;N;_AUDIT;N;

1. CANCEL current command
2. PURGE command (suppressing dialog)
3. ENTER
4. ALL
5. ENTER
6. ENTER (all names to purge)
7. N (to verify)
8. ENTER to execute
9. ENTER to repeat the command
10. R for Regapps
11. ENTER
12. ENTER (all names to purge)

13. N (to verify)
14. ENTER to execute
15. AUDIT command
16. ENTER
17. Y (to fix all errors found)
18. ENTER to execute

So now that you've worked up a sweat, you don't want to lose all this work. Hmmm… which macro should we run now?

(Subtle hint: "ZQS")

Learn how to add macro commands to a tool palette (and a bit more)

It's fairly simple to add commands to tool palettes (I can say that because I've done it), obviously the first time is the toughest. The same can be said about creating new commands in the CUI (Customize User Interface) Editor. We'll take a look at how to do both. If you wish to keep things simple and only work with tool palettes, that works for me. In my opinion, just like creating a macro to automate repetitive tasks, I only want to create the command once so I add the macro as a command in the CUI.

That said, I'll cover both the CUI and the tool palettes.

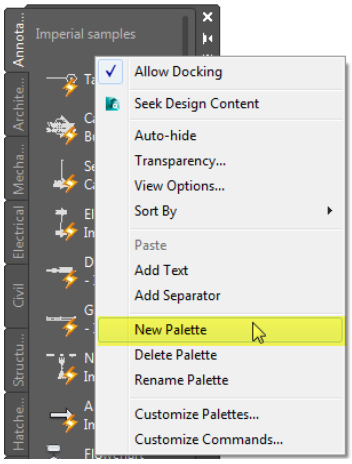
Adding command macros to a palette

Use the keyboard CTRL+3 shortcut to open your tool palettes (if closed).

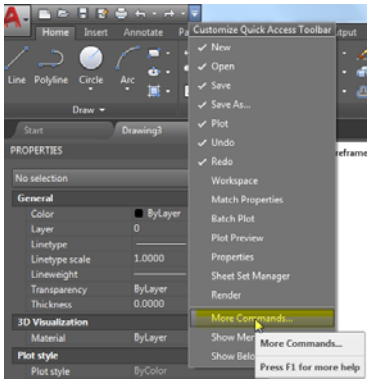
Right-Mouse click over a palette to create a new palette

Call the new palette "Macros"

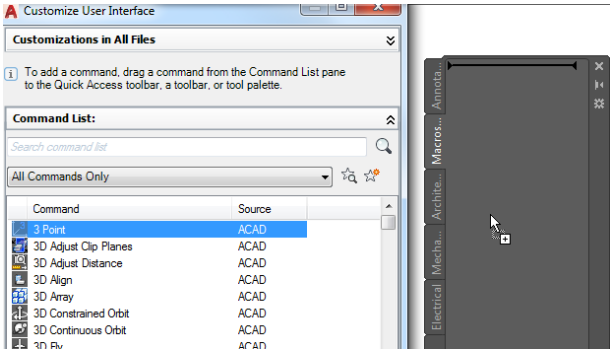
In order to add our macros to the palette it's easiest to create a command using one of our macros in the CUI, but it's even easier to cheat and place an existing draw or modify command on the palette and then modify it with our macro in the command string.



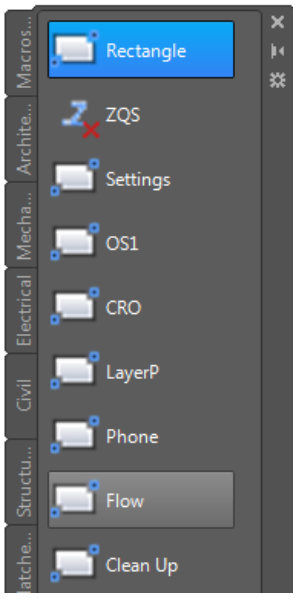
To do that we do need to open the CUI, in this case the Quick CUI will suffice. From the QAT (Quick Access Toolbar) select "More commands…"



Pick the first command in the list, press and drag and drop that command to the blank palette. Then close the CUI editor.



Next, Right-Mouse click over that command and select Properties. In the Properties dialog fill in the information necessary for the “Rectangle” macro. Then close the Properties dialog. One can easily add an image (but that’s another topic) to the tool palette command.

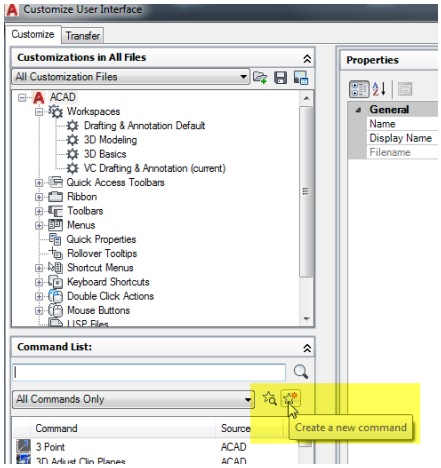


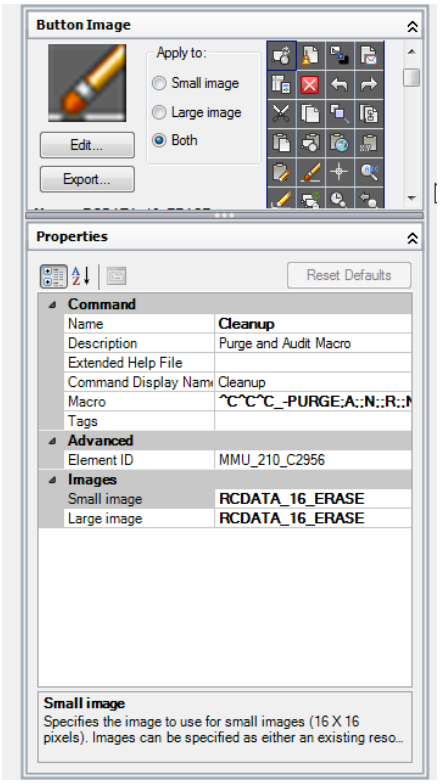
Adding commands to the CUI

My preferred method for tool palette commands is to add them to the CUI. This allows me to not only add the commands to the tool palette, but add them to other elements of the CUI such as the Ribbon as well. The CUI is a topic in itself, and my recommendation would be to create a custom CUI, which I would then add as a partial menu. That way the customization is easily stored and I can share the menu for use on additional settings. Upgrading at a later date is easier as well.

Unfortunately, we just don't have time for that in this class so I am including a supplement from my previous Autodesk University 2013 class, AC3274: CUIX Zen: Customizing the CUI. This will cover the nitty gritty details.

At this time, let's take a look at the quick start glossy overview:



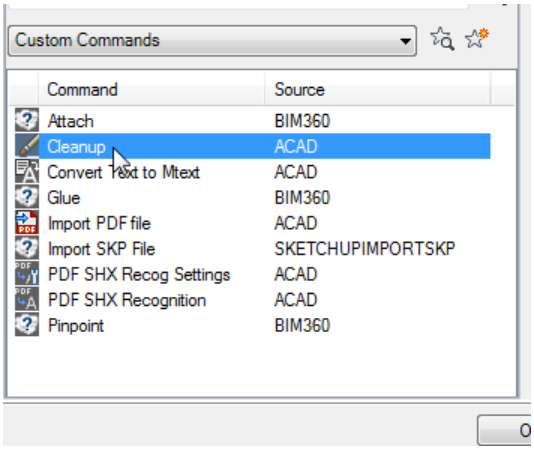


1. Type CUI at the command prompt
2. This opens the Customize User Interface Editor
3. In the "Customize" tab select the "All Customization Files" drop down list and set the acad.cuix current.
4. The "Command List:" pane in the lower left corner of the CUI editor is now empty.
5. Next to the "All Commands Only" drop down select "Create a New Command".
6. In the "Properties" pane on the right, populate the fields as follows:
    - Name: Cleanup
    - Description: Purge and Audit Macro
    - Extended Help File: <blank>
    - Command Display Name: Cleanup
    - Macro: ^C^C^C_-PURGE;A;;N;;R;;N;_AUDIT;Y;
    - Tags: Purge, Audit (optional)
    - Images (Small and Large): Select the erase image
    - Select "Apply"
7. In the "Command List:" select "Custom Commands" from the drop down list to view the new command.
8. Note: The new command and that it resides in the " acad.cuix"

Repeat the procedure for the remaining macros.

Once finished, you can then drag and drop the commands onto the tool palette as was done in our previous example.

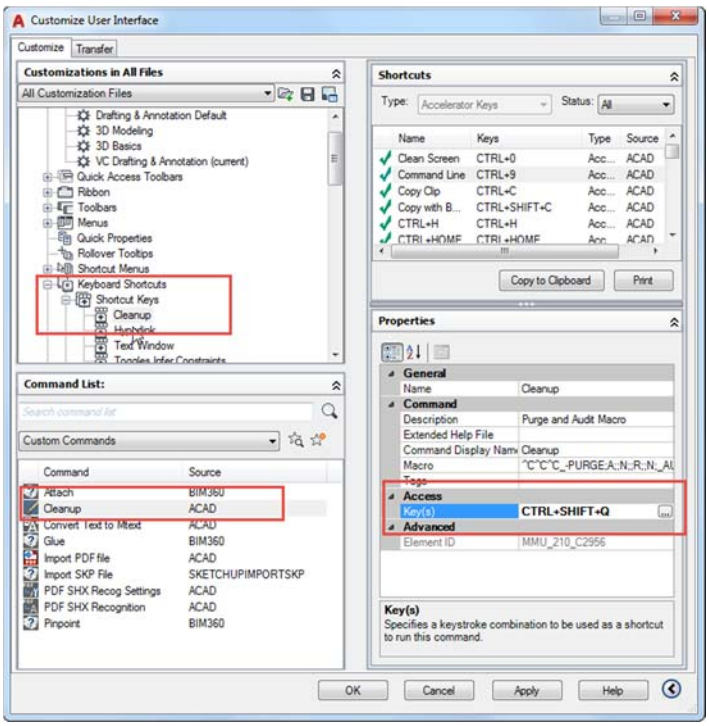
TIP: To make it easy in finding the commands, you can filter the custom commands in the CUI Editor.



Running macros from the command line

The question I'm asked the most is, "How can I run my macro from the command prompt?"

That is a bit limited. Unless one creates an AutoLISP function to create a command (which AutoCAD LT does not support), then the only option is to create a shortcut key. To do this:



1. In the CUI editor, select the command
2. Drag and Drop the command to the Keyboard Shortcuts -> Shortcut Keys section of the 'All Customization Files' pane
3. Keep the command selected in that pane
4. The currently used shortcuts are now available in the top right pane, 'Shortcuts'
5. You'll see the Properties of the command appear in the lower right.
6. Assign any available two or three key combination
7. Select Apply and then OK
8. Run the command

That's a Wrap!

I'd like to sincerely thank you for attending my session! I hope that you have found it of value and will continue to do so when applying this to your working environment.

Please note that no AutoCAD entities were harmed in the making of this script.

Additional Resources

- How to work with the CUIx to Customize AutoCAD (AU 2013 Class)
- About Creating Command Aliases
- About DIESEL and Status Bar Customization
- About Command Customization
- About Command Macros
- About Special Control Characters in Command Macros