AUTODESK UNIVERSITY 2016

Pumping up Productivity in the LAB with Macros One Character at a Time

Sam Lucido – Haley and Aldrich, Inc.

GEN16077-L This class is filled with all the tips and tricks you'll need to create macros in Autodesk® AutoCAD®. Macros are small groups of commands and routines that can be any length, without requiring any special characters at the end of the line. In this class, you learn to create custom macros that apply to your business needs and enhance your productivity on designs. You will develop a custom tool palette placing different types of macros on the palette for quick access during project design. Most importantly, what you learn will bring value to your employer and give you knowledge that can help you excel within your field.

Learning Objectives:

- Identify, change and create macros.
- Use the action recorder to create custom macros.
- Create macros to automate multiple commands.
- Create a tool palette with custom macros enabled.

Lab Assistants:

- Rick Ellis - CADapult Software Solutions, Inc., President - http://www.cadapult-software.com/
- R.K. McSwain - LJA Engineering, Inc., CAD Manager - http://cadpanacea.com/
- Paul Munford - Graitec, Application Engineer - http://cadsetterout.com/

Description

A macro in AutoCAD can be defined as a way to automate a task that you perform repeatedly with more than one command or keystroke. In the class, we are going to review how to create macros that apply to your business needs enhancing productivity. We will begin with the Action Recorder, move to Tool Palettes and finally dive into the CUI to pull out some code and apply to our project needs. I will spark your interest knowing that when you return to the office you will be able to bring value and apply some hints and tips to help you excel.

Your AU Expert

Sam Lucido is a CAD Services Manager and Civil Designer with Haley and Aldrich, Inc. Within these roles he presents workshops on CAD productivity to managers and users in both corporate and classroom settings. He provides support on a wide variety of architectural, civil, mechanical, and structural design projects. Sam has over 25 years of experience involving production design and drafting, user support and standards coordination. He continues to be very self-motivated and enjoys working in a team environment to accomplish project objectives and create high quality deliverables. Sam has authored 10 articles in AUGIWorld from 2011 to 2016. He is the owner and operator of CADproTips.com, and Expert Elite Member, and professionally certified in AutoCAD.

lucido1373@gmail.com
http://www.CadproTips.com
@CADProTips

Table of Contents

Overview 3

Action Recorder 4

Tool Palettes 7

Exercise 1 9

Exercise 2 12

Exercise 3 14

Exercise 4 18

Exercise 5 21

Exercise 6 27

Exercise 7 28

Exercise 8 31

Exercise 9 34

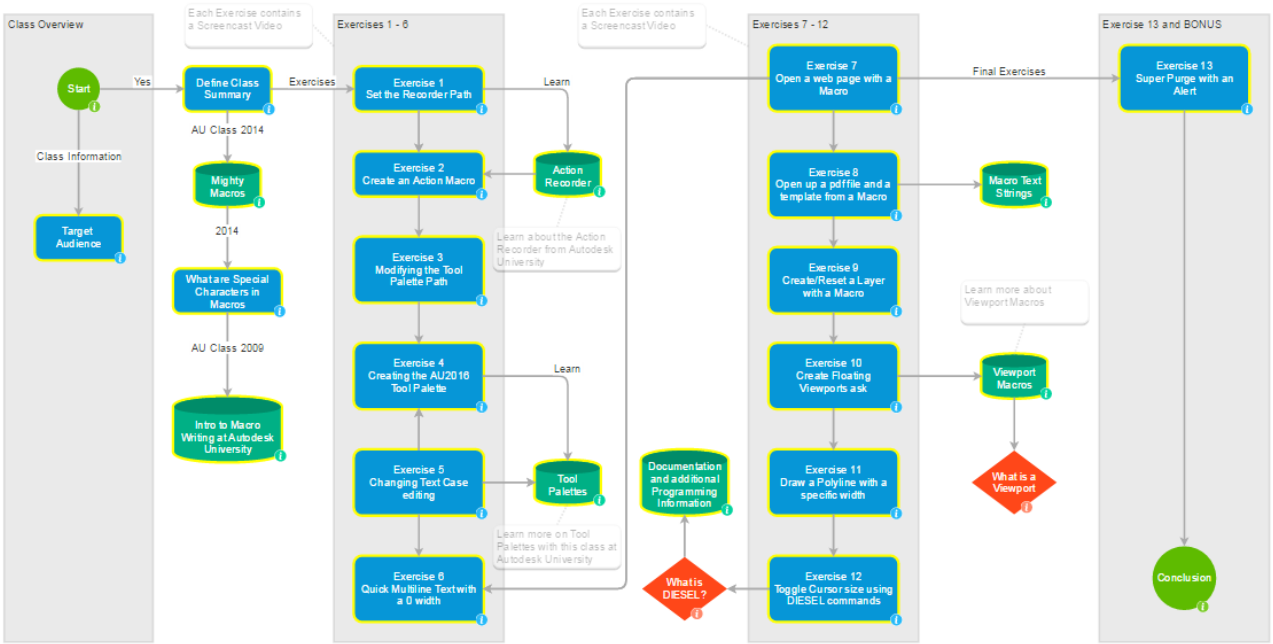
Exercise 10 38

Exercise 11 40

Exercise 12 43

Exercise 13 46

Conclusion 49

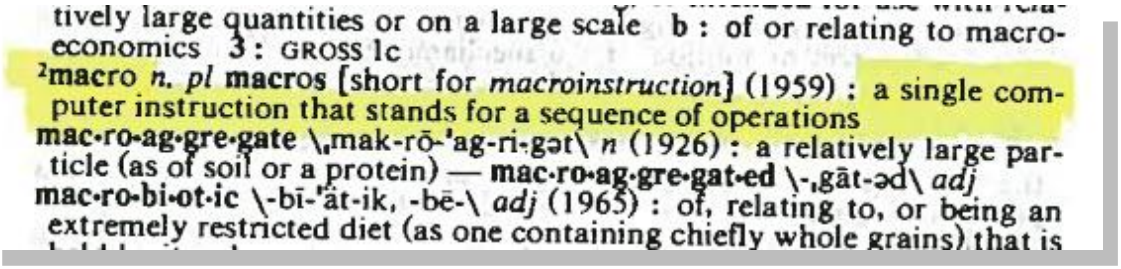


Overview

Let's begin by defining the three important topics of this class: A Macro, the Action Recorder, and Tool Palettes. This is not a tutorial on tool palettes or the action recorder but we will need to quickly review these topics as we are going to use both during this lesson. We will use the action recorder to create macros to swap our tool palette paths to help make it easier to standardize and create our new palette. We are also going to explore macros within the CUI to show how you can use existing commands to help create a new more efficient process.

What is a Macro?

A macro (in many different programs) can be defined as a way to automate a task that you perform repeatedly with more than one command or keystroke. In AutoCAD macros can be shortcuts to a series of commands to help make the process of design more efficient. In Figure 1 a macro is stated as a single instruction. Use the action recorder to record a series of commands and build a macro then run it automatically to repeat a series of steps. To write a macro, you type the commands in the macro properties section as you'd type them in at the command line. If a command displays a dialog box, you would place a dash in front of the command to suppress the dialog box. We will cover special characters as we begin to build our macros.


tively large quantities or on a large scale b : of or relating to macro-economics 3 : GROSS 1c

²**macro** *n. pl* **macros** [short for *macroinstruction*] (1959) : a single computer instruction that stands for a sequence of operations

mac·ro·ag·gre·gate \ˌmak-rō-ˈag-ri-gət\ *n* (1926) : a relatively large particle (as of soil or a protein) — **mac·ro·ag·gre·gat·ed** \-ˌgāt-əd\ *adj*

mac·ro·bi·ot·ic \-bī-ˈät-ik, -bē-\ *adj* (1965) : of, relating to, or being an extremely restricted diet (as one containing chiefly whole grains) that is

From Autodesk Help

A macro can contain commands, special characters, DIESEL (Direct Interpretively Evaluated String Expression Language) or AutoLISP programming code. (AutoLISP is not supported by AutoCAD LT)

Note: As AutoCAD-based products are revised and enhanced, the sequence of prompts for various commands (and sometimes command names) might change. Therefore, your custom macros might require minor changes when you upgrade to a new release of your application. You add macros to interface elements by using the Customize User Interface (CUI) Editor. Select an existing command or create a new command in the Command List pane. Enter macros in the Macros section of the Properties pane. There are no length limitations for macros. However, you do need to know how specific characters are used in macros and be aware of other considerations or limitations.

The Action Recorder

The action recorder is simply a tool to record macros for later use. Most users don't have the time or want to pursue the training in LISP, DIESEL and VBA. This is where the action recorder can help you. You simply start the recorder and AutoCAD will read the commands you input then save as a macro in your custom location. You can include requests for user input and messages to make the macro work interactively. When you save a macro, it has an ACTM filename extension. You'll find it in your Support\Actions folder of your AutoCAD installation. You can share ACTM files with others users on your team by placing them on a network or shared drive. You'll find the Action Recorder on the Tools tab, in the Action Recorder panel as shown in Figure 1. You can also choose Menu Browser (the A button)> Tools> Action Recorder.

The Action Recorder is a panel on the ribbon under the Manage tab (Figure 1) that contains tools to record, modify, and play back an action macro. While recording, actions, commands, and input values are captured and displayed in the Action Tree as value nodes. A value node records the input provided at any prompt within a command, including acquired points, text strings, numbers, keywords, or other values that are entered when recording a command. After recording is stopped, you can save captured commands and input to an action macro (ACTM) file which can be played back later. Once saved, you can specify base points, insert user messages, or change the behavior of recorded input values to pause for input during playback. You can also manage recorded action files with the Action Macro Manager from inside of the product or through the file system from outside the product.

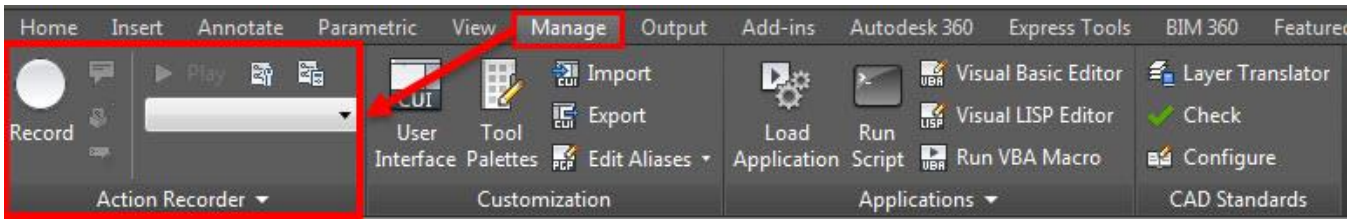


Figure 1: The Action Recorder

Modifying an Action Macro

I wanted to provide additional information within the handout that we will not review during class. This next section can also be found in Autodesk Help. After an action macro is recorded, you can change the recorded actions and values from the Action Tree of the Action Recorder. You can edit an action macro by:

a) Removing an action.
b) Editing a recorded value.
c) Inserting a user message.
d) Requesting a pause for input.
e) Inserting a base point. (Setting all points relative to the first point selected during playback, or use a combination or absolute and relative coordinates).

Loading an Action Macro

Action macro (ACTM) files found in or added to the paths defined by the system variables ACTRECPATH and ACTPATH are loaded automatically. If more than one action macro file has the same name, the first action macro file that is found is loaded. The other files with the same names are ignored and not loaded.

To identify the path of a loaded action macro, select the action macro from the Action Macros drop-down list and expand the Action Recorder panel. Right-click the action macro's node in the Action Tree and click Properties. In the Action Macro dialog box, the path to the action macro file is displayed in the Folder Path box.

Playing back an Action Macro

When playing back an action macro, note the following:
The results of an action macro can vary when played back in a drawing that has different drawing settings compared to those used in the drawing when the action macro was recorded. An action macro is paused when a dialog box is displayed, and then resumed when the dialog box is dismissed. If the dialog box is cancelled, the action macro will continue to play back but the results will likely be different. An action macro will fail if it contains custom commands or references to VBA macros when the required files are not loaded.

Sharing an Action Macro

One of the best advantages about an action macro is the ability to share with other users. Changes during upgrades may affect the result of our macros so we must keep two things in mind when upgrading our software packages.

1. Action macros recorded with an AutoCAD-based vertical product might not play back the same way in AutoCAD or a different AutoCAD-based vertical product depending on the commands that were used when the action macro was recorded.

2. Action macros might not play back correctly when recorded in one language of the product and then played back in a different language of the product.

Action Macro Manager

When saving a macro use the Action Macro Manager as shown in Figure 2. This is the dialog box that will provide a list of macros as well as an explanation below to the function of the macro.

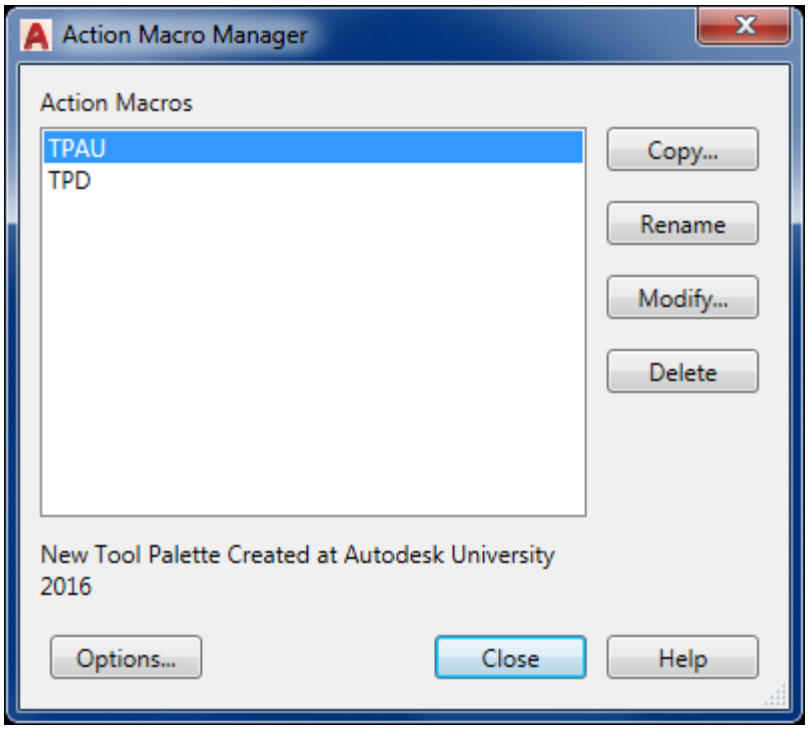


Figure 2: Action Macro Manager

There are two variables that The Action Recorder uses. The first is **ACTPATH** which controls the read-only folder or path for playback. The second is **ACTRECPATH** which controls the read/write folder for creating the action macros. The **ACTRECPATH** is the folder where you would place your custom macros. There can only be one of those but you can add multiple paths for placing the playback files. Both of these variables can be set under the Options Menu Files Tab as shown in Figure 3.

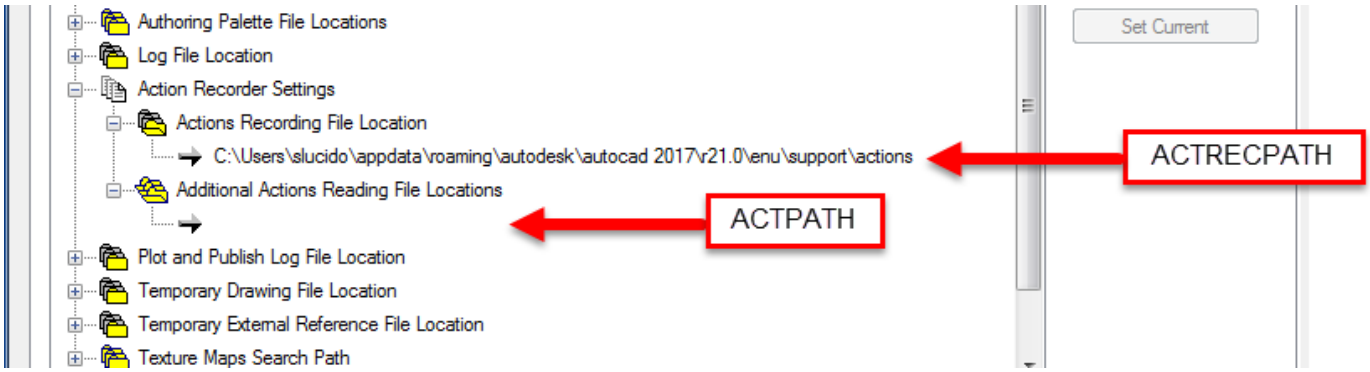


Figure 3: Action Macro Paths

Notes:

Tool Palettes

Tool Palettes provide the shortest quickest way to increase productivity without programming. A tool palette is a free-floating tab that you can bring up on screen and keep active while you work in your drawing, so you have quick access to common symbols, commands, and most any other tool you need to complete the design. You can add any command from AutoCAD to a palette, creating quick access to all of your favorites and shortcuts. Figure 4 shows the default location of the tool palette path in AutoCAD.

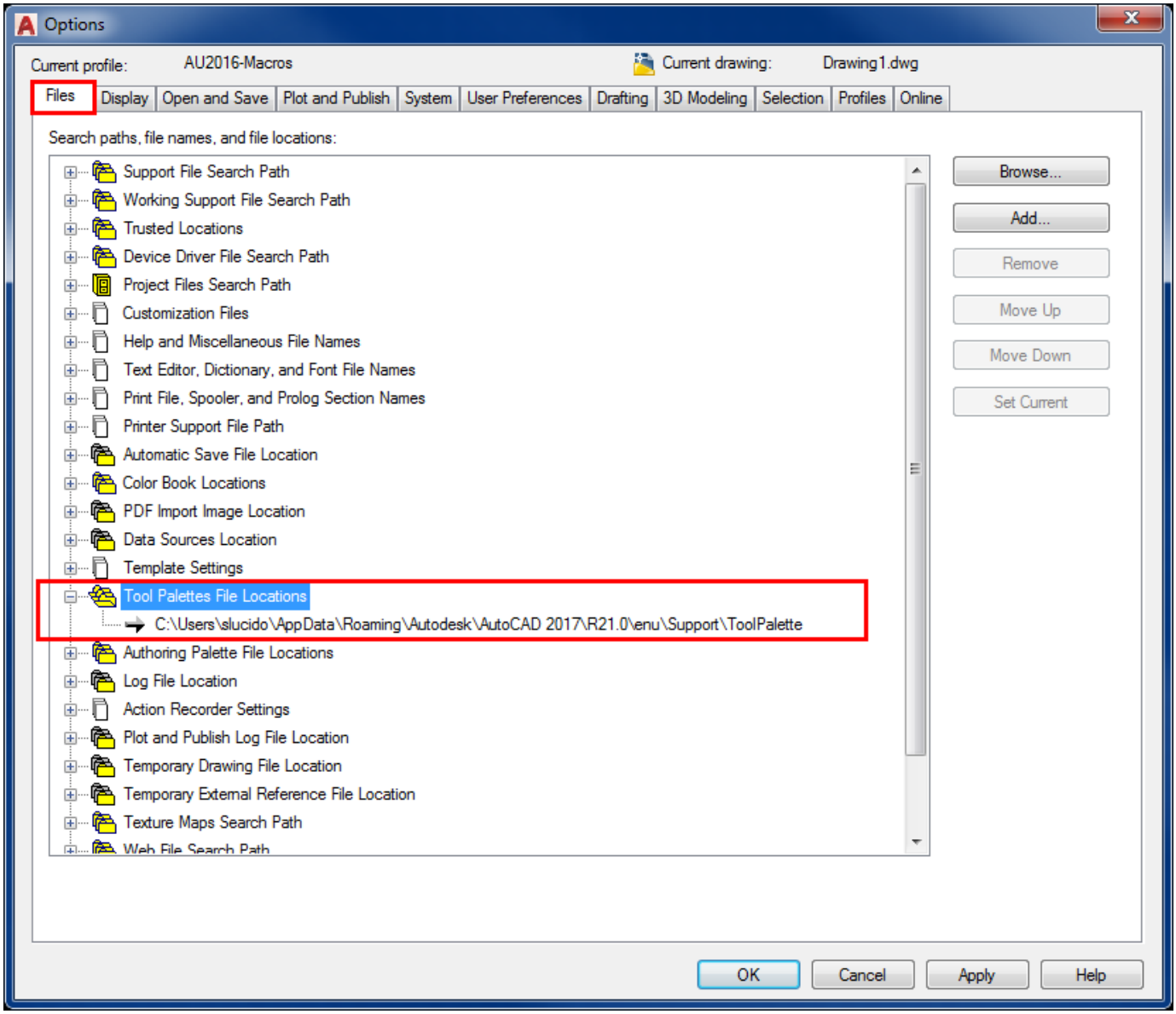


Figure 4: Default Tool Palette Path

We do not want to alter the default AutoCAD palette path so we will first make a macro to restore the default path and flip between our new palette and the default version. Don't worry, if you lose your default palette you can always leave a blank space in the file location and you will be prompted to return to default as shown in Figure 5

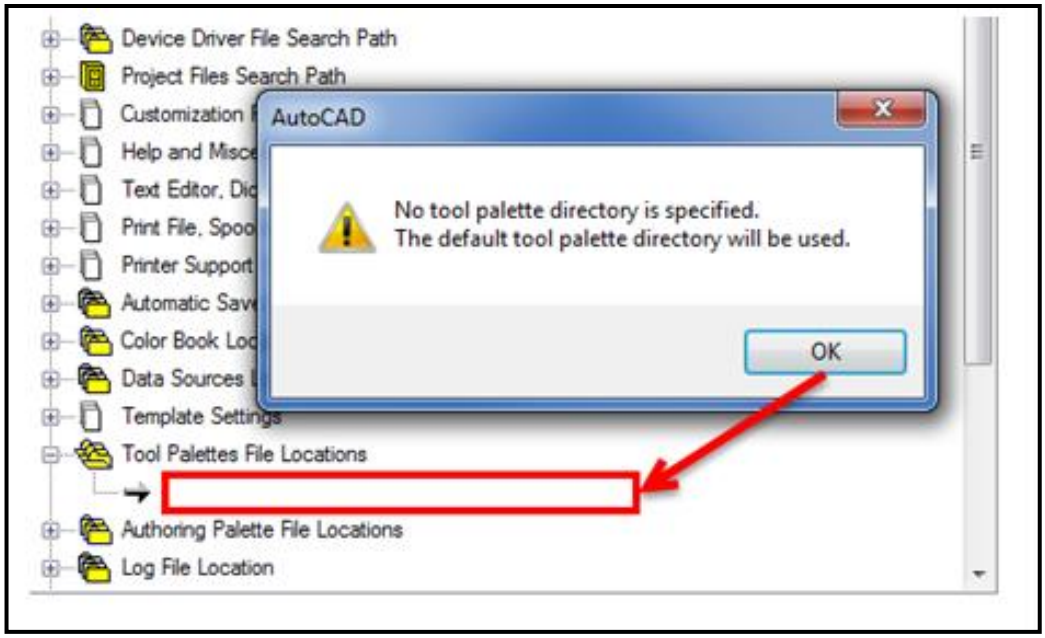


Figure 5: Restore Default Path

We are going to create a new blank tool palette so we can begin placing our macros on it. First, I would like to refer to *Matt Murphy's* class(s) at Autodesk University 2012 and 2013. I am only listing two but he has provided classes on Tool Palettes and the Action Recorder for several years. Both of these classes can be a great resource for getting started with palettes and the action recorder in AutoCAD. Look in the catalog as he is repeating Tool Palettes this year as well.

AC3441: The Productivity Power of AutoCAD® Tool Palettes – Revealed!
(This class was presented at Autodesk University 2012)

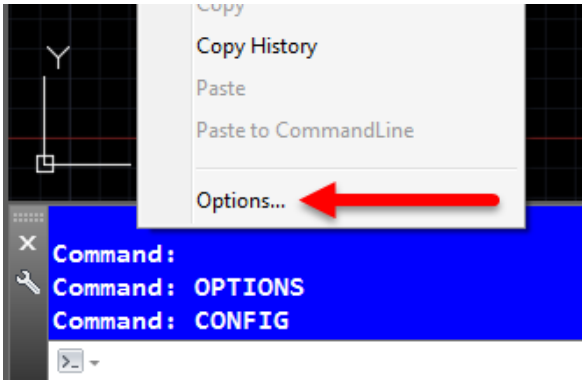
AC2098: Painless Productivity Programming with the Autodesk® AutoCAD® Action Recorder – Revealed!
(This class was presented at Autodesk University 2013)

I used Matt's technique of creating a macro to locate the tool palette path which gives the illusion of groups. This works great when switching between different groups of blocks, command tools and engineering disciplines. We will go through this briefly to assist you in getting things setup. Just like managing your support path statements in AutoCAD, you can set a Tool Palette path location and not use the group feature. Now that we have covered the basics it's time to move on to the exercises.

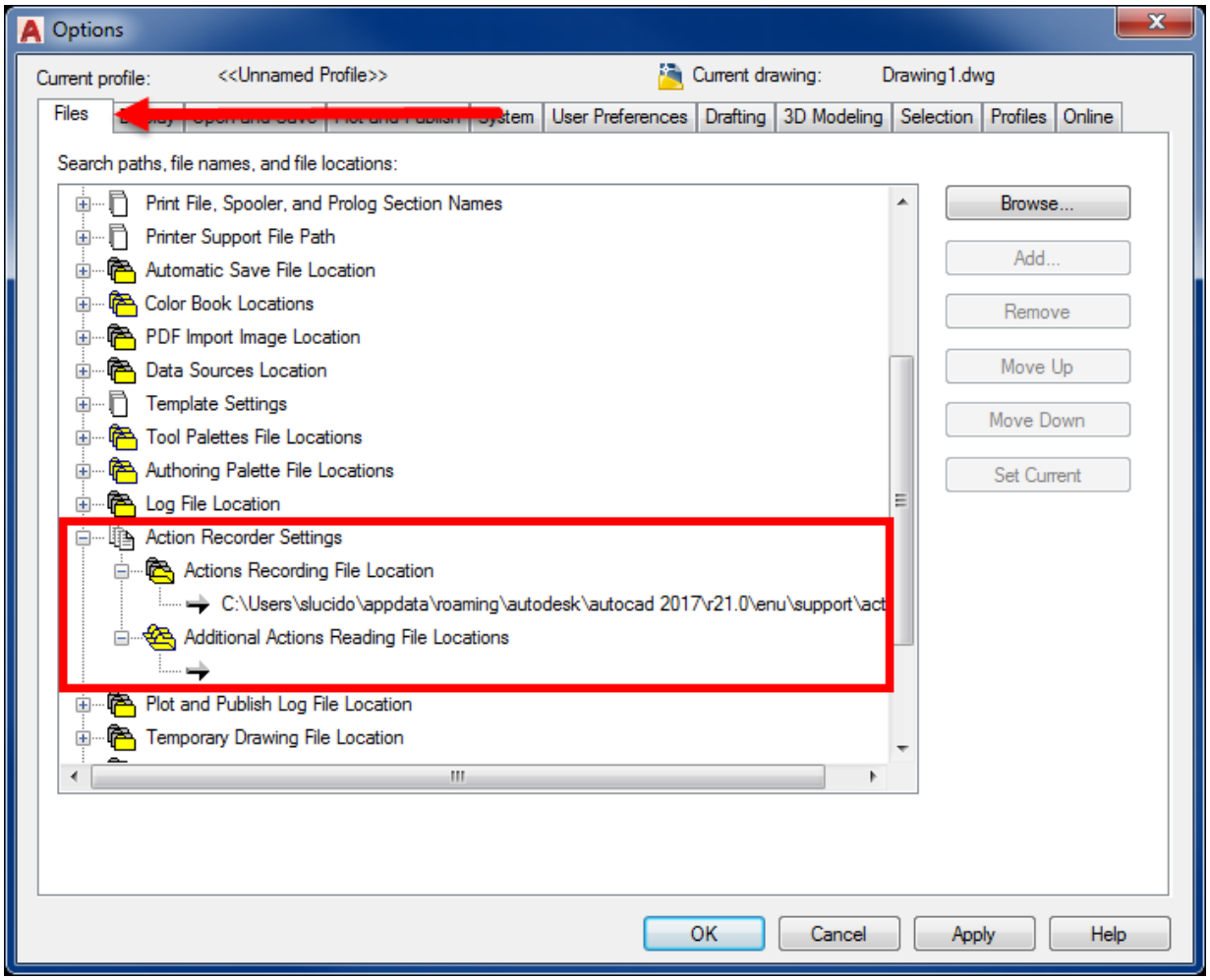
Exercise 1: Setting your Action Recorder Path for Custom Macros

In our first exercise we will set the path in our Action Recorder to save our macros in a secure location where we can transfer and/or access at a later time.

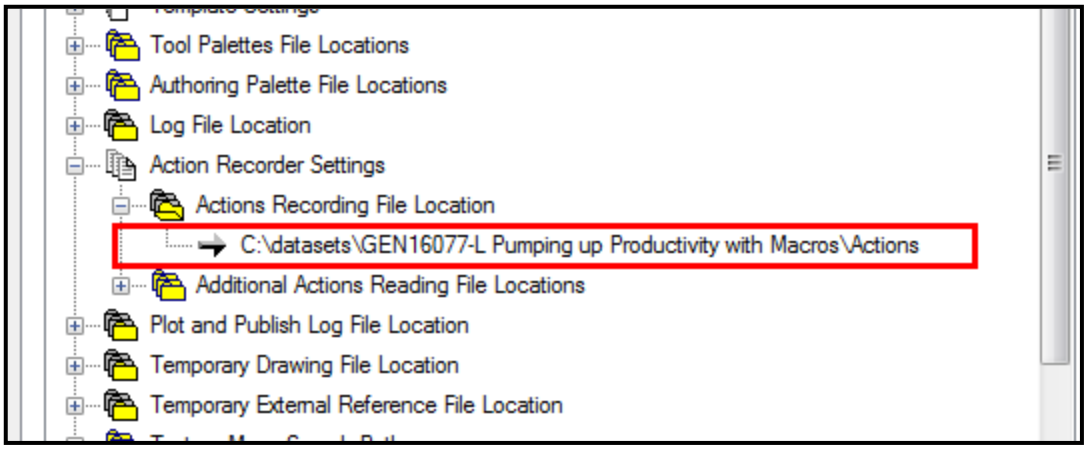
1. Type Options, Config, or right-click at the Command prompt to bring up the Options Menu.



2. Select Options or hit Enter to bring up the options menu. Pull out the setting for the Action Recorder File Location and the Additional Action Reading file locations as shown.

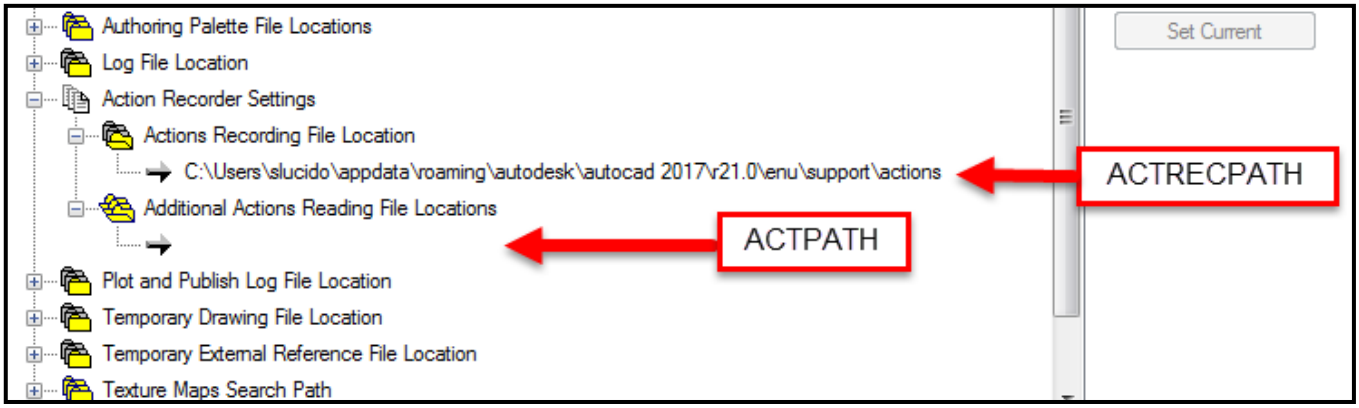


3. Select the Actions Recording File Location and Remove then browse out to our dataset folder as shown. This is the folder where are Action Macros will be stored with a file extension of .ACTM.



4. Our Action Recorder Path is set.

5. Notes: There are two variables that The Action Recorder uses. The first is **ACTPATH** which controls the read-only folder or path for playback. The second is **ACTRECPATH** which controls the read/write folder for creating the action macros. The **ACTRECPATH** is the folder where you would place your custom macros. There can only be one of those but you can add multiple paths for placing the playback files. Both of these variables can be set under the Options Menu Files Tab as shown in Figure 3.



About Playing back an Action Macro from Autodesk Help

After an action macro is recorded with the Action Recorder, you can play back the series of recorded commands and input values.

You play back an action macro from the Action Recorder panel. You can also enter the name of the action macro at the Command prompt. As an action macro is played back, you might be paused for input or requested to respond to a message or requested to insert a base point.

Based on the current action or request for user interaction in the action macro, an icon near the cursor is displayed to indicate when the action macro needs input in order to continue. A dialog box may be displayed where you enter a value or use the recorded value.

As an action macro is played back, the series of commands are performed one after the other until playback is complete or an error is encountered. Some of the reasons for the playback to stop or fail are as follows:

Invalid command. The command that is defined in the action macro is unknown. The action macro might have been recorded in a different product or contains custom commands or macros that are not loaded.

Empty selection. The current action expected a selection set of objects, but no objects were selected.

Macro cancelled. The Cancel button in one of the dialogs boxes was clicked.

Location of Action Macros

When you record an action macro, it is saved to the path defined by the system variable ACTRECPATH. For playback, paths are defined by the system variable ACTPATH.

Both sets of paths are used when loading and playing back an action macro. In the Action Recorder Settings node on the Files tab of the Options dialog box, you can to set the paths used for both recording and reading action macro files.

Record and Modify Action Macros

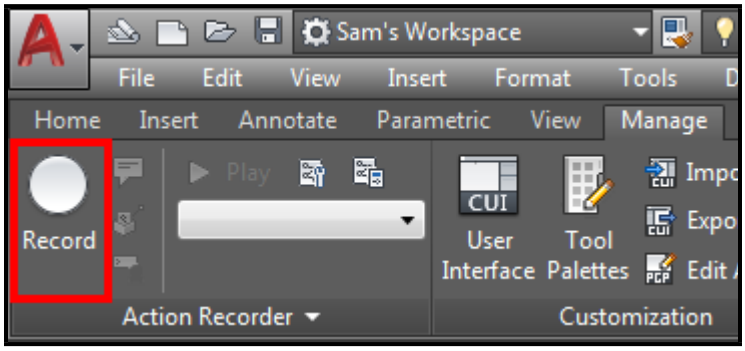
Recording and modifying action macros is done with the Action Recorder panel on the ribbon.

Exercise 2: Drawing a 10 x 10 Box with the Action Recorder

In this exercise we are going to use the action recorder to create a macro to make a new layer and draw a 10 x 10 rectangular box.

1. Open AutoCAD to a new drawing.

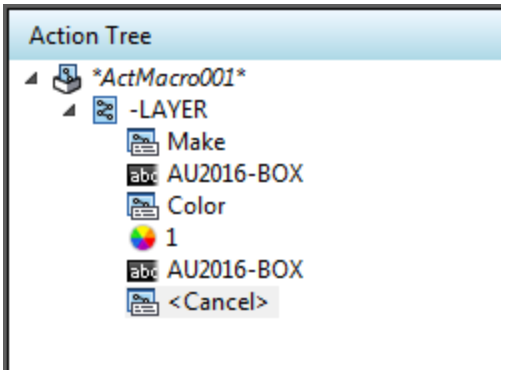
2. Move over to the manage tab on the Ribbon and select Record as shown below in Figure 1.



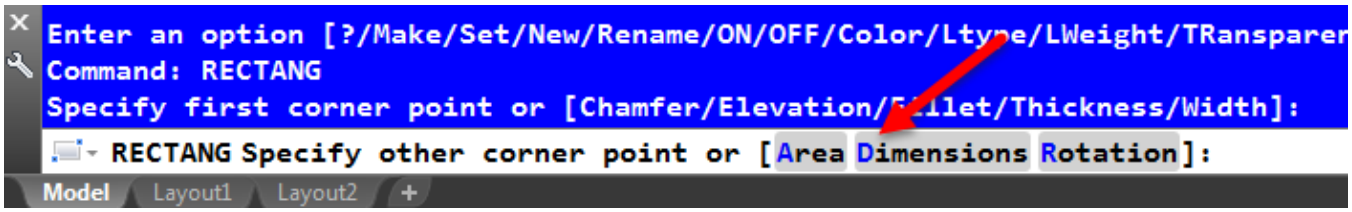
3. Hit Record and type -layer at the command prompt and follow the steps as shown below. You don't necessarily have to use the hyphen but it is easier to follow the code in the macro.

    a. **-layer**
    b. **M**
    c. **AU2016-BOX**
    d. **C**
    e. **1**
    f. **AU2016-BOX (or select enter to hit the default playback setting the current layer.**
    g. **ESC**

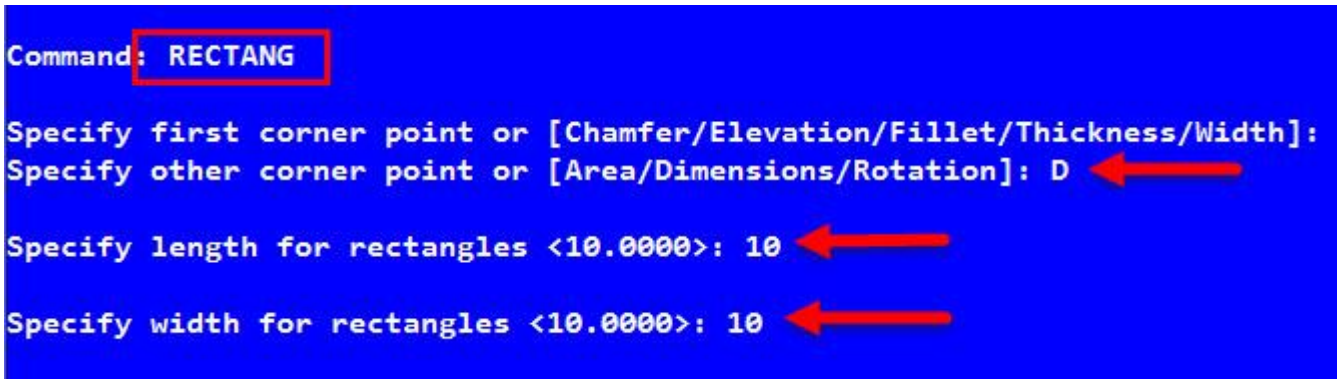
4. Your Action Tree should look like what is shown below. Hitting ESC will cancel our layer command and put us right back out to the command prompt.



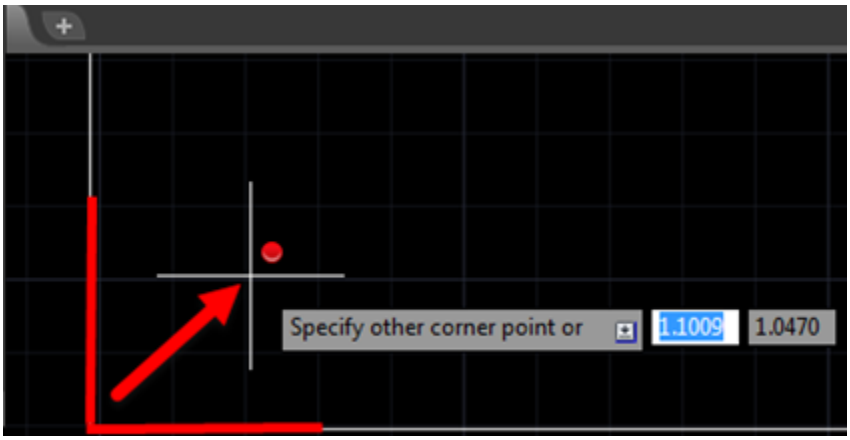
5. At the command prompt type RECT (for rectangle).
6. Select a point on the screen and hit Enter where you will see the following 3 options.



7. Type or Select D at the command prompt
.
8. Type 10 for the length and 10 for the width as shown.



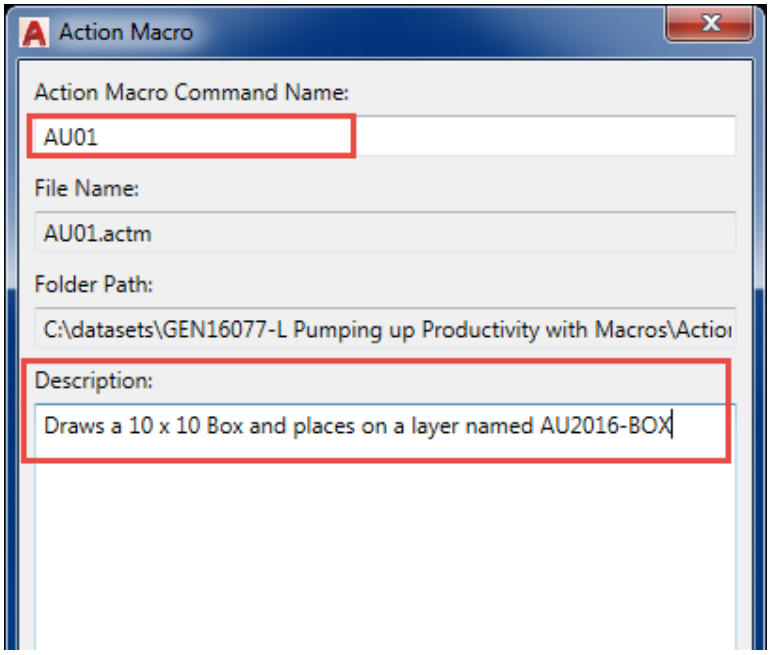
9. When asked for a specific location choose the upper right quadrant as shown.



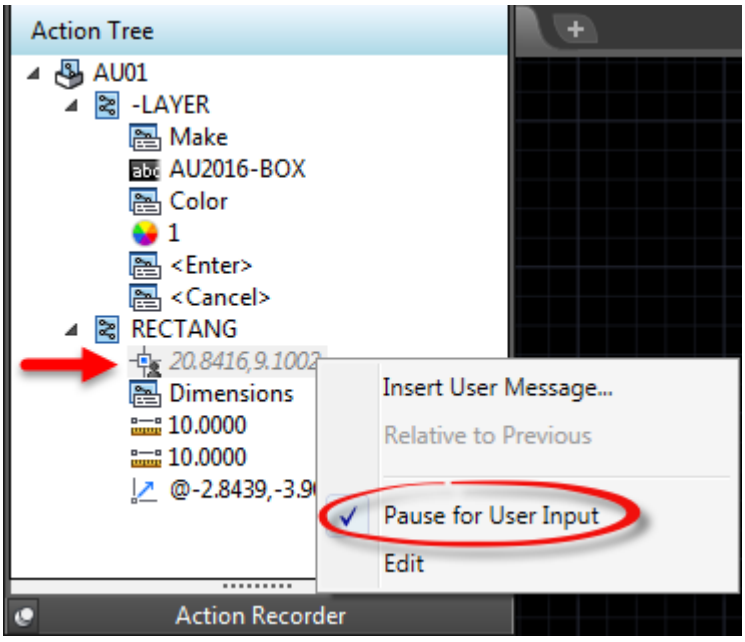
10. Stop the action recorder and save your file as shown.

11. Name the Macro as AU01 and give it a description as shown to the right.

12. One last step. Go back to the Action Tree (pull down the Action Recorder tab with your AU01 showing.



13. Right click on your first coordinate and hit Pause for User Input. We need to do this so the command will pause for us to select the starting location of our 10 x 10 box.



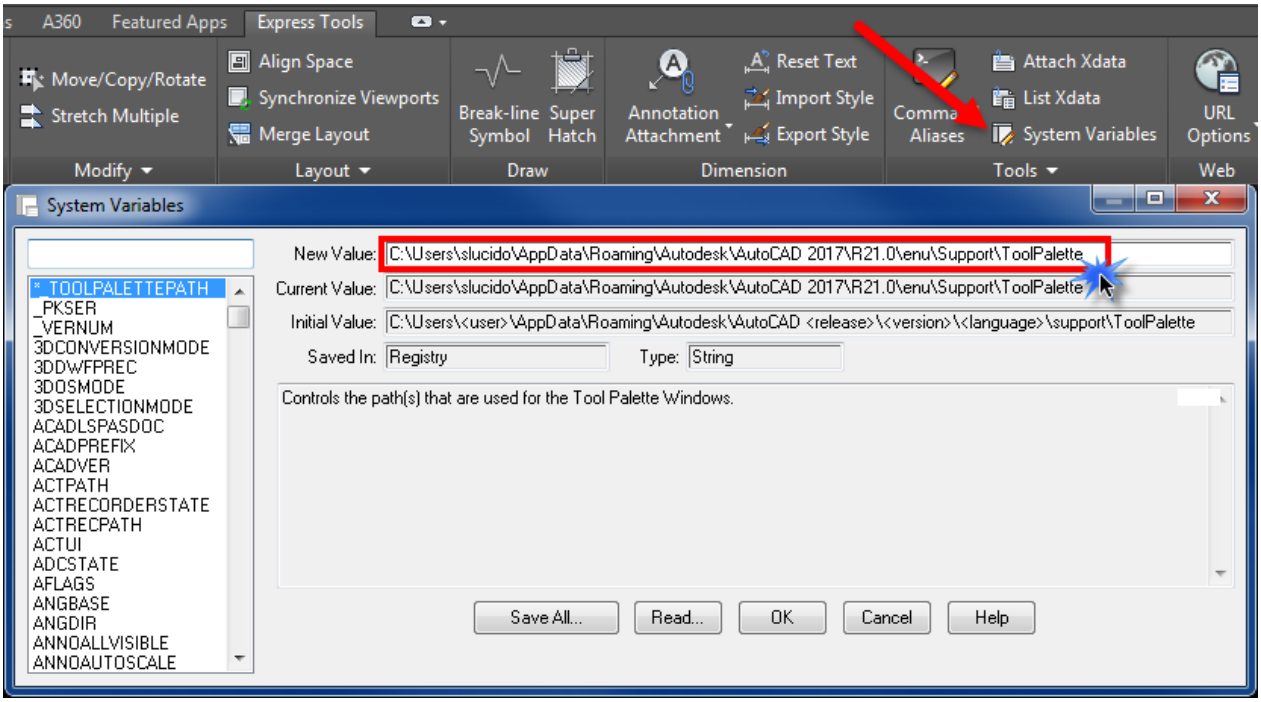
14. Close the recorder and test your macro.

Exercise 3: Modifying the Tool Palette Path with a Macro

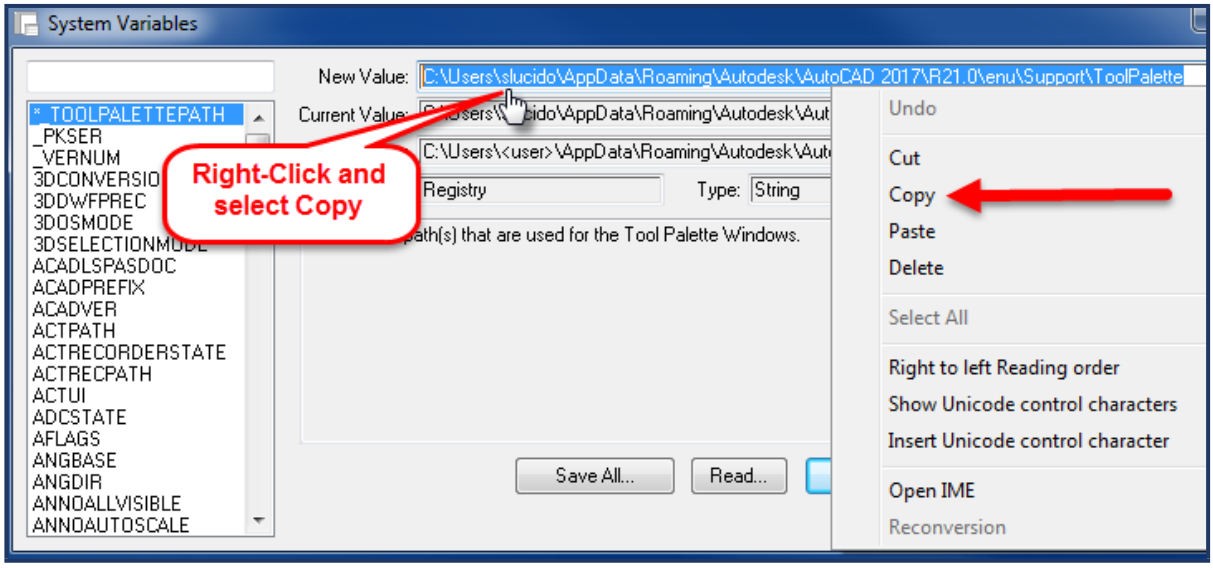
Let's begin by using the action recorder to create a macro that will change the tool palette path to a new location where we will build our macros. Understanding how the action recorder works; we are now going to setup paths using the recorder to switch to our different palettes.

First we are going to grab our default tool palette path by copying and pasting from the system variable editor.

1. On the Express Tools tab of the Ribbon select System Variables.

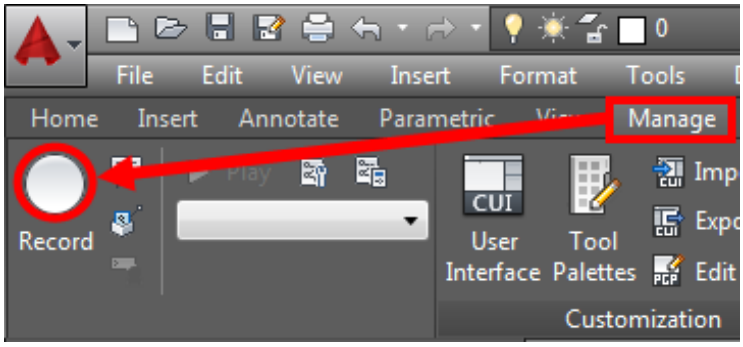


2. Left click at the end of the path and drag until you have the entire path highlighted; then right-click and select copy.

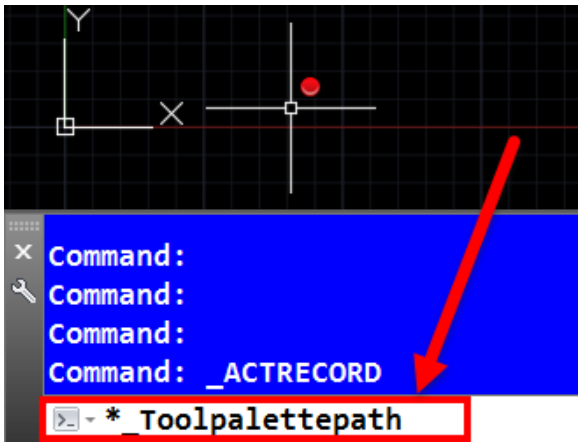


3. Cancel out of that command and move out to the Manage Tab of the Ribbon.

4. On the Manage Tab of the Ribbon move to the Action Recorder Panel and Hit Record.

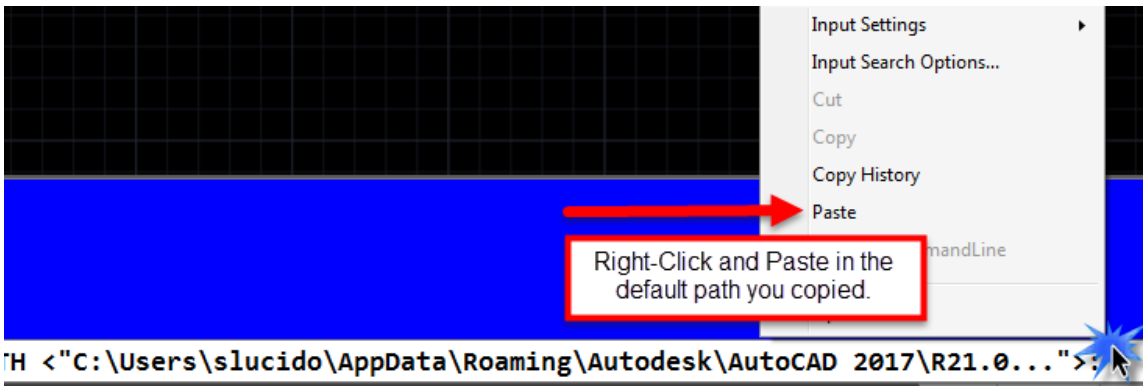


5. On the Command Line type *_Toolpalettepath as shown. *HINT: This command will not show up in AutoComplete therefore you have to type the entire command in as shown*.

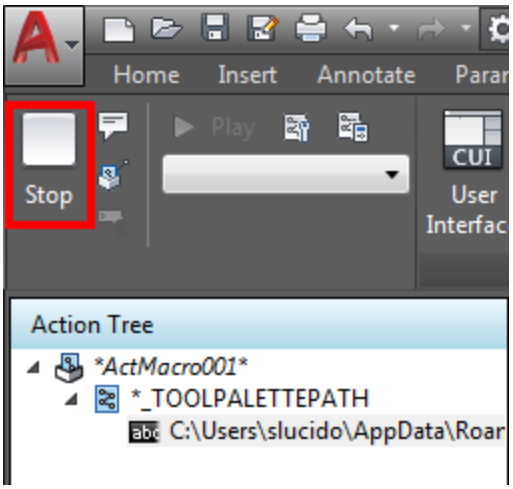


6. Hit Enter.

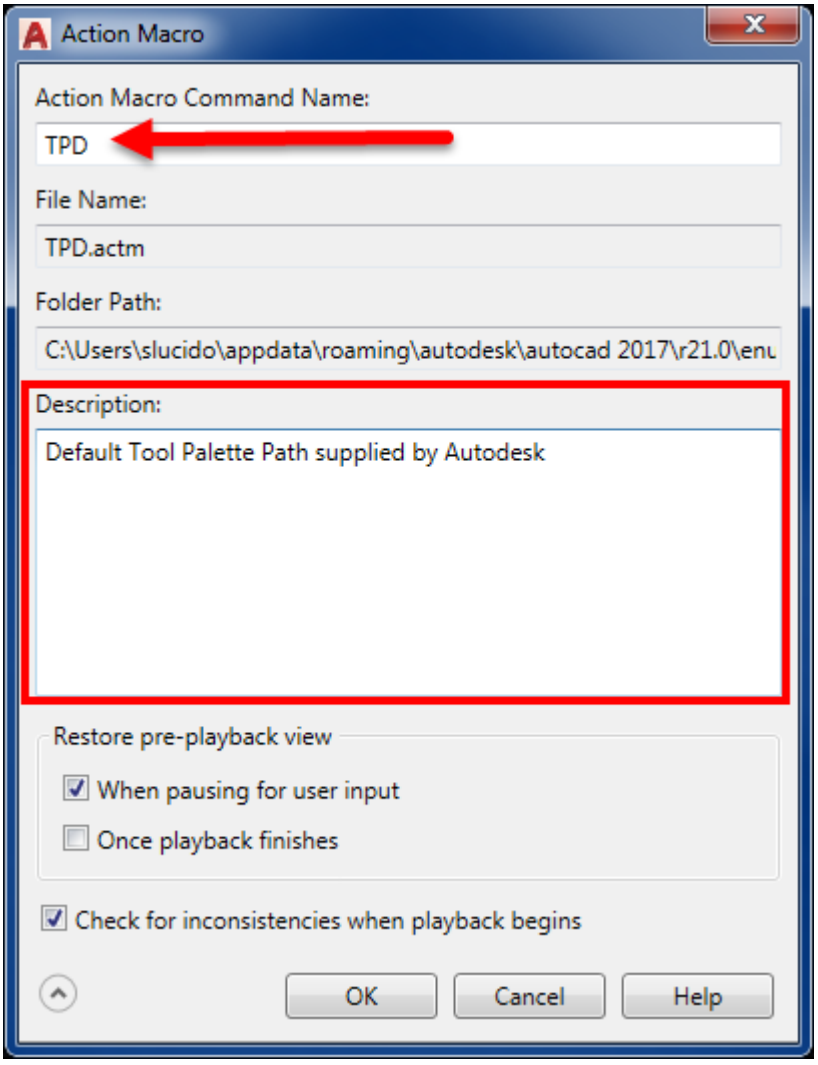
7. At the end of the command line right-click and Paste (remember we saved that default tool palette path).



8. Stop the Action Recorder.



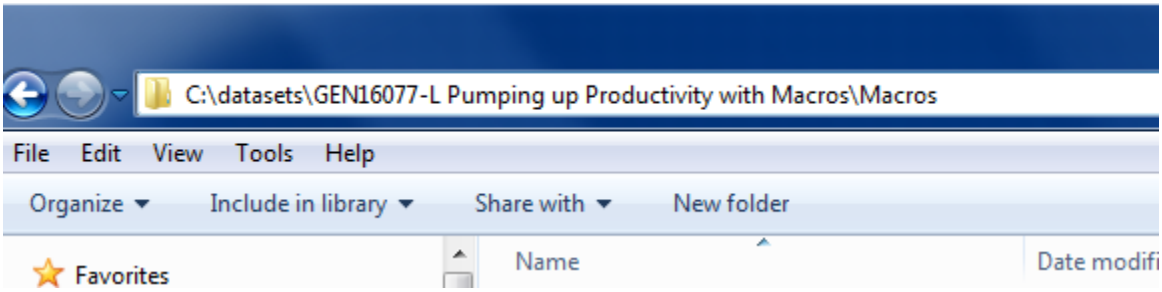
9. Stop the Action Recorder and save the Macro as TPD (i.e. Tool Palette Default). Make sure you give your macro a description so that next CAD guy or gal knows exactly what the command will do.



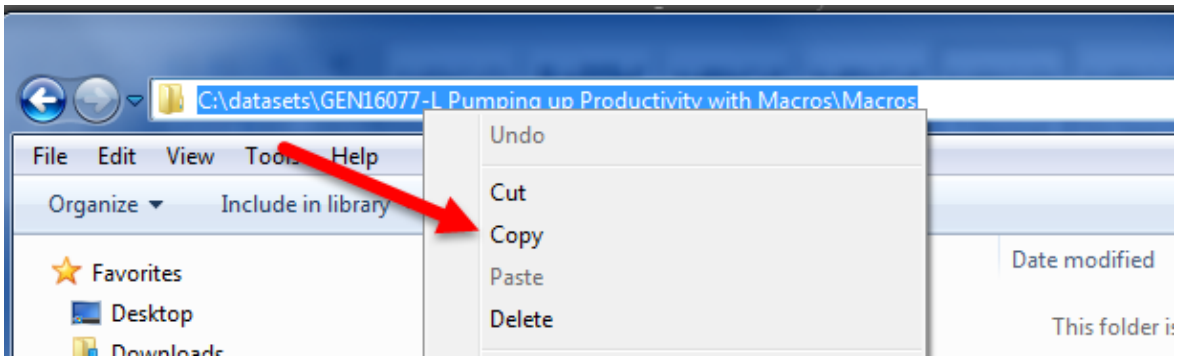
Exercise 4: Setting your AU Palette path with a Macro

We are going to use the same method by copying the path from a source and pasting into the tool palette path section using windows explorer.

1. Move out to your C drive and navigate to the dataset folder as shown below.

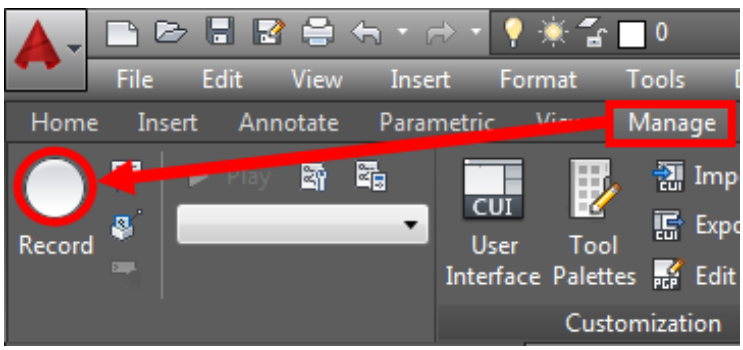


2. Left click at the end of the path and drag until you have the entire path highlighted; then right-click and select copy.

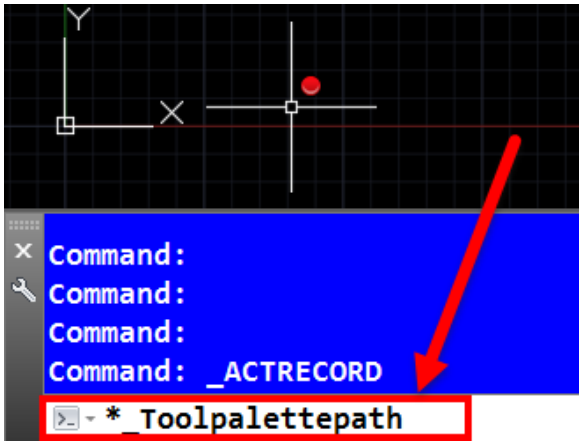


3. Cancel out of that command and move out to the Manage Tab of the Ribbon.

4. On the Manage Tab of the Ribbon move to the Action Recorder Panel and Hit Record.

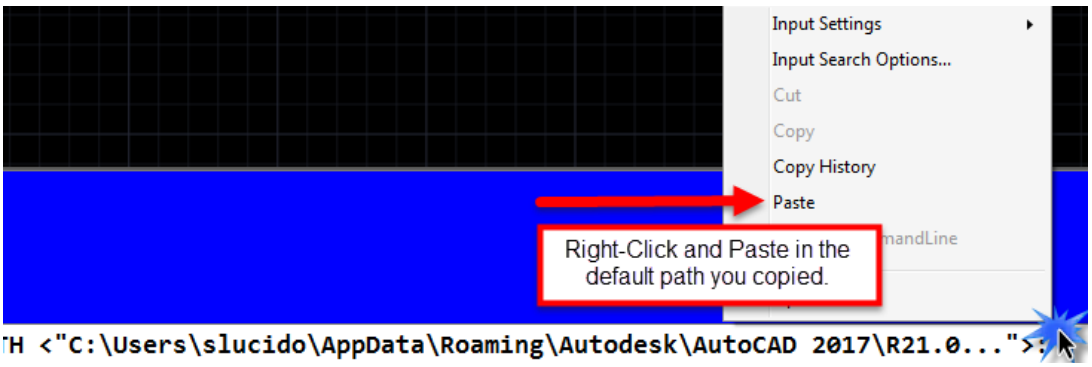


5. On the Command Line type *_Toolpalettepath as shown. *HINT: This command will not show up in AutoComplete therefore you have to type the entire command in as shown*.

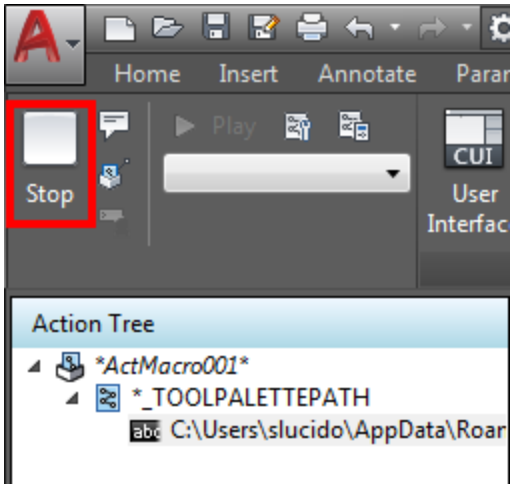


6. Hit Enter.

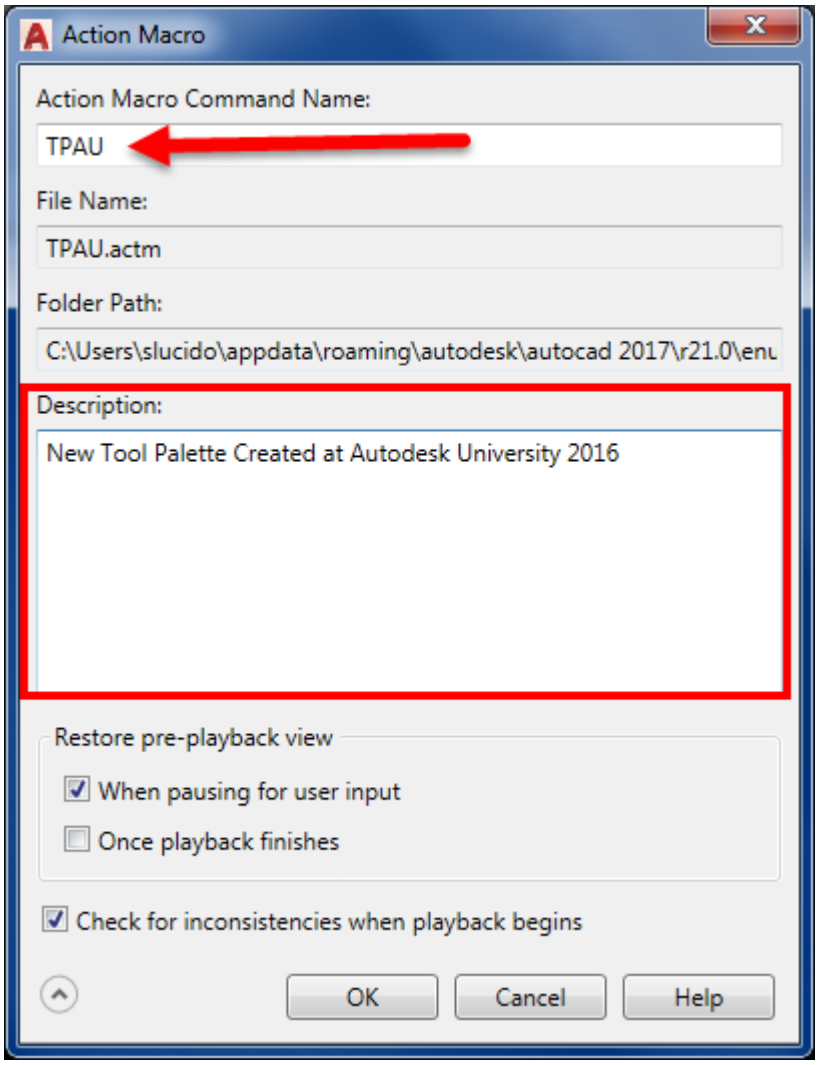
7. At the end of the command line right-click and Paste (remember we saved our new tool palette path).



8. Stop the Action Recorder.



9. Stop the Action Recorder and save the Macro as TPAU (i.e. Tool Palette Autodesk University). Make sure you give your macro a description so that next CAD guy or gal knows exactly what the command will do.

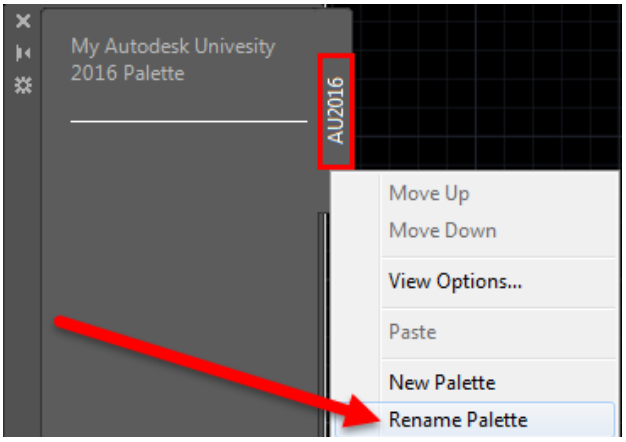


10. Test out your macros by typing TPD at the command prompt then typing TPAU. You can now flip between the default palettes and your personalized palettes we will create.

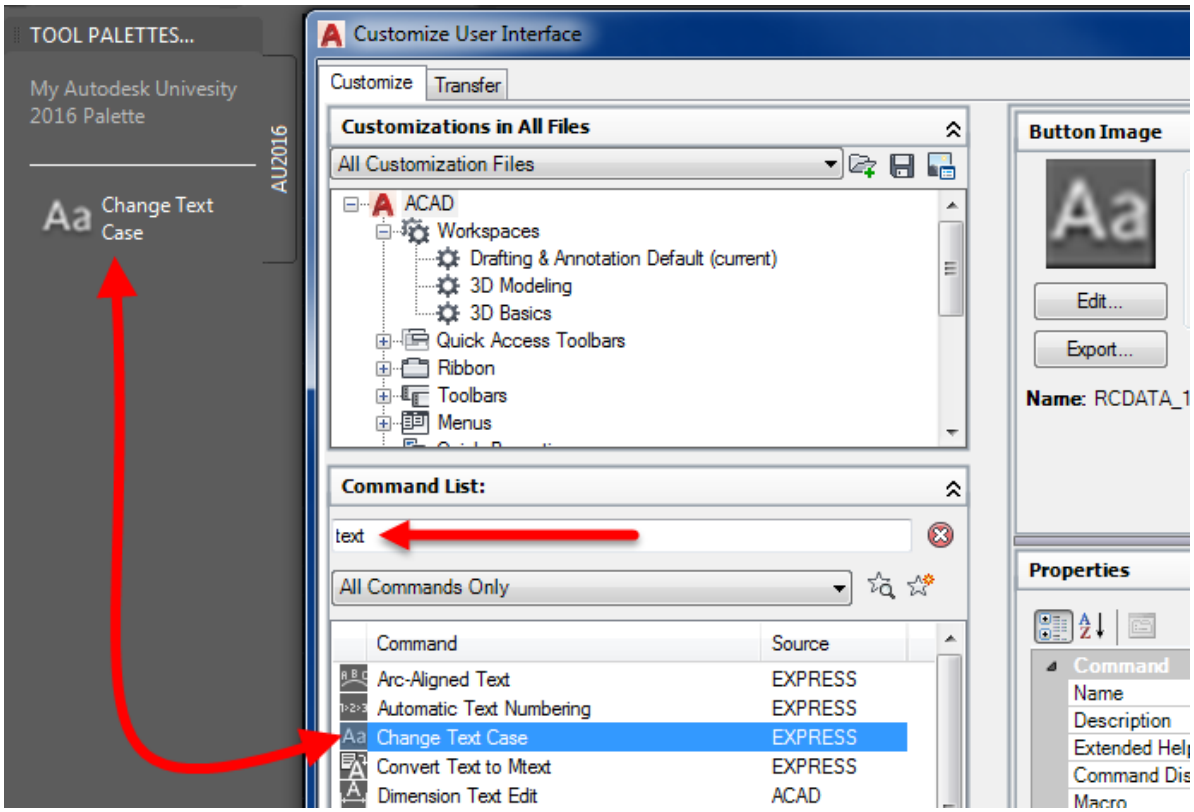
Exercise 5: Creating Macros to change Text Case

We are now going to populate our AU Palette with macros.

1. With your blank AU Palette open right-click the tab and rename to AU2016. You can also right-click in the palette area and add a piece of text and a separator as shown.



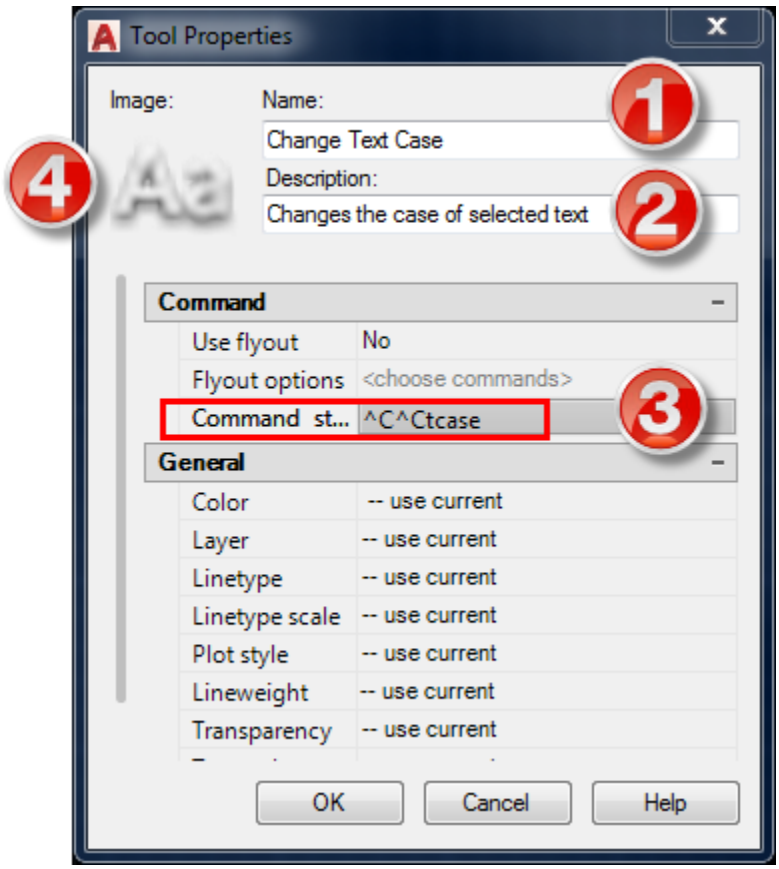
2. We are going to start by creating a Macro to change text from lowercase to uppercase and vise-versa. Type CUI at the command prompt and in the Command list pane type text. Left-click and drag that command to your new palette.



Now that we have our command in there let's make some modifications to the code, then finally the button.

3. Close out of the CUI and right-click your command to bring up the tool properties where we will perform 4 tasks as listed below.

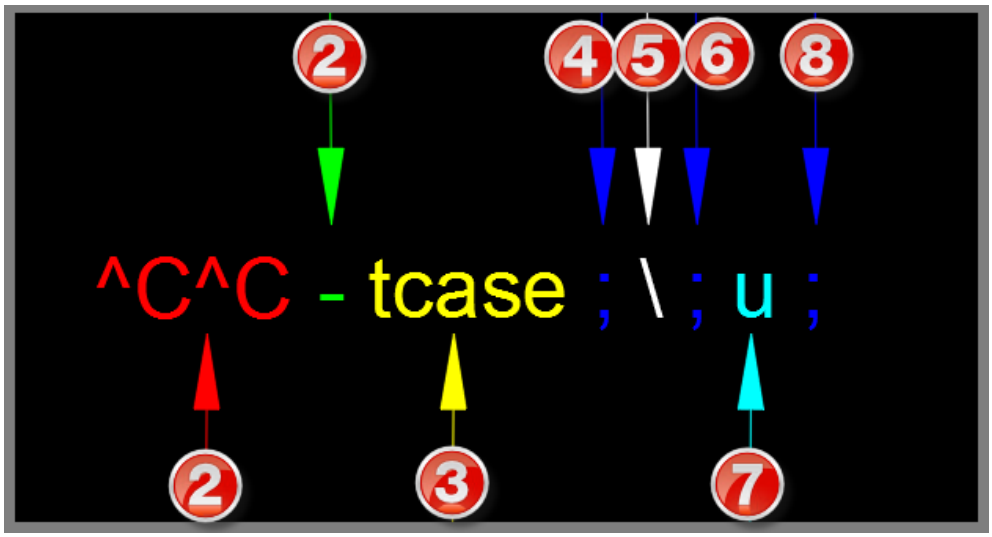
    1. Rename your New Command: **Change Text to Uppercase**
    2. Describe your New Command: **Changes the case of selected text to Uppercase**
    3. Create the Macro: **^C^C_-tcase;\;u;**
    4. Change the button.



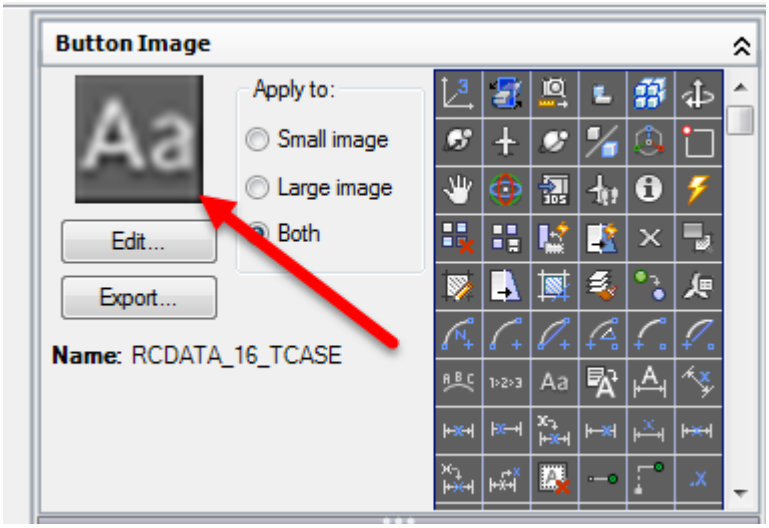
4. Taking a closer look at the macro and what we typed in.

    1. The ^C^C represents a cancel of a command. Not only once but twice because we all know in AutoCAD it takes 2 to cancel a command.
    2. The hyphen (-). This will suppress the dialog box; we need to have the command be driven from the command line, bringing up a dialog box will halt the process.
    3. Launch the command from the command line.
    4. The semi-colon represents pressing an enter or return on the keyboard.

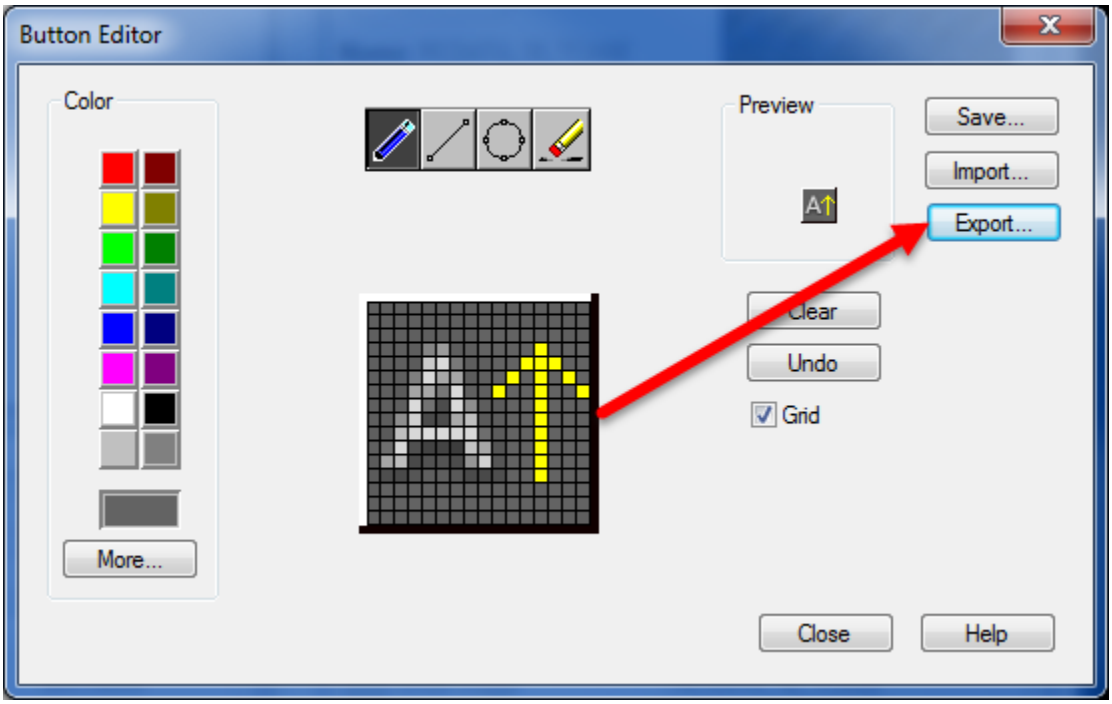
5. The forward slash represents a pause for user input. Note: When using macros to access a path to locate a file you must reverse the slash in the path to represent the path.
6. Another semi-colon to hit a return on the keyboard.
7. The u will tell the user that we are going to choose uppercase for our text.
8. Finally, one more return to complete the command.



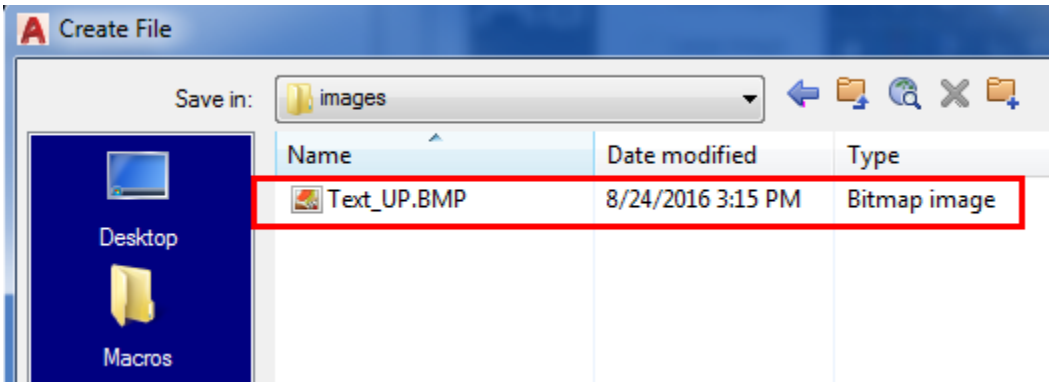
5. Open up the CUI and go back to the change text case command and select the command then edit on the button as shown.



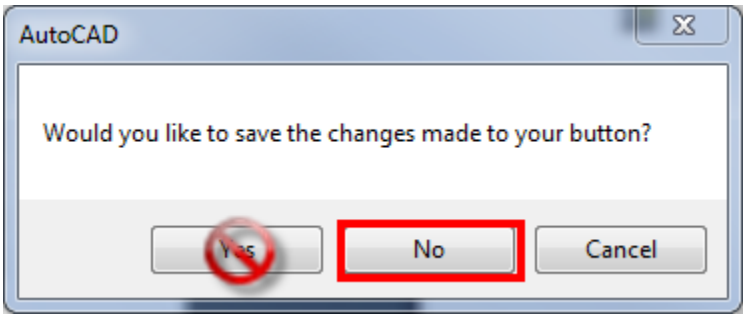
6. Edit the Image as shown using the button editor. Erase the other piece of text then draw an UP arrow to indicate the text case being changed to upper format.



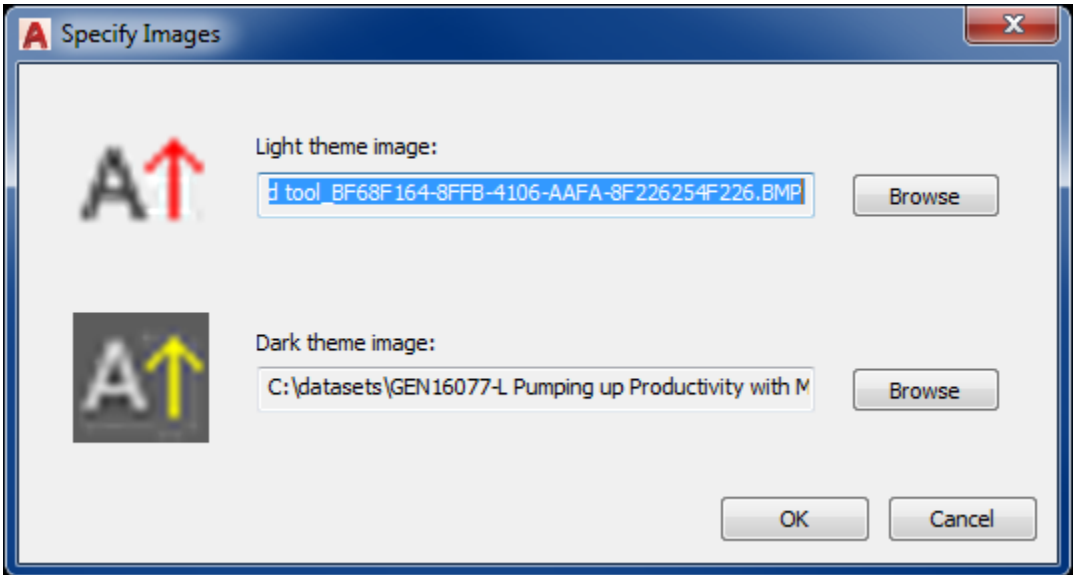
7. Export the image to your images folder and give it a name Text_UP.bmp as shown.



8. When prompted to save your changes DO NOT save the changes and hit NO. Saving the changes will alter the original button contained within the CUI. Alternatively, you can import a BMP or JPG file which will work with both dark and light themes displaying the image in it's original content.



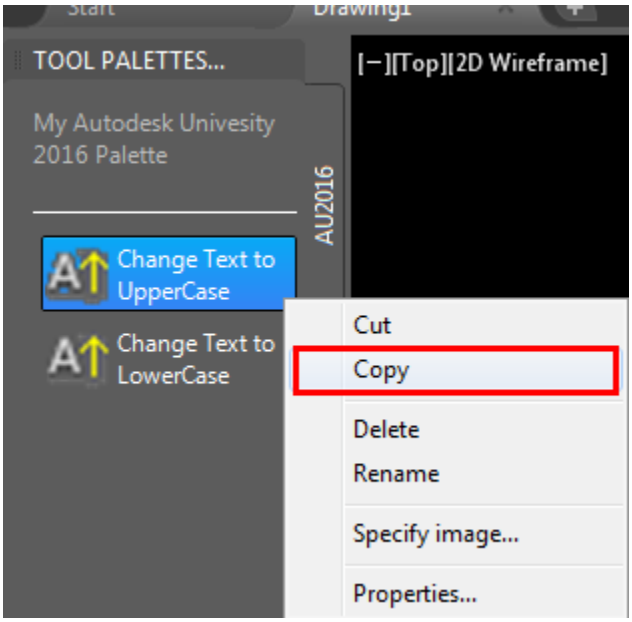
9. Last move out to your command and select your text for the dark theme button. If you select the image you created for the dark them it will not display correctly. In some cases, you may have to create two images depending on your company standards. There is another image in the folder for you to use for the light them in AutoCAD.



10. Hit OK and your image and palette will be there to test out. Type in Dynamic Text or Mtext and test out your new button.

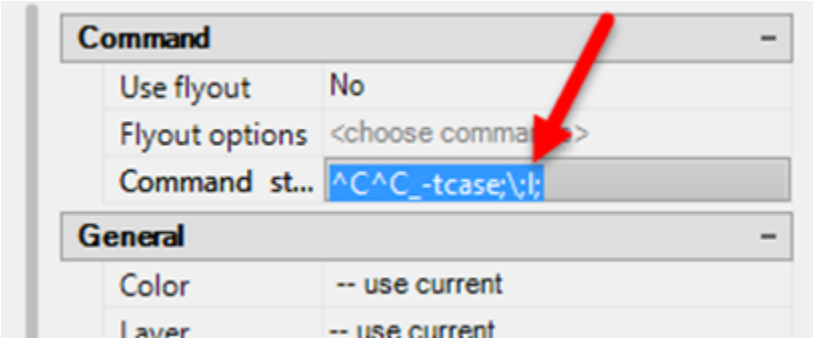
11. We are now going to copy that entire macro and just change a letter.

12. Right click your macro on the palette and hit copy then right click anywhere in the tool palette area and hit paste. Your command has been duplicated as shown.

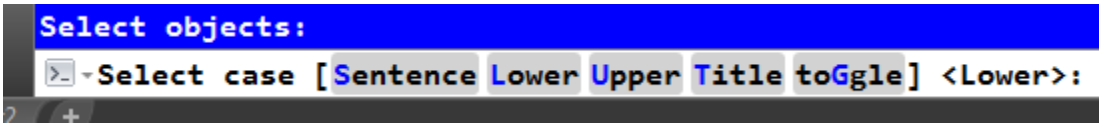


13. Rename the Copied Tool to Change Text to Lowercase as shown then right click the command.

14. In the Command String change the letter “u” to an “l” for lower case.



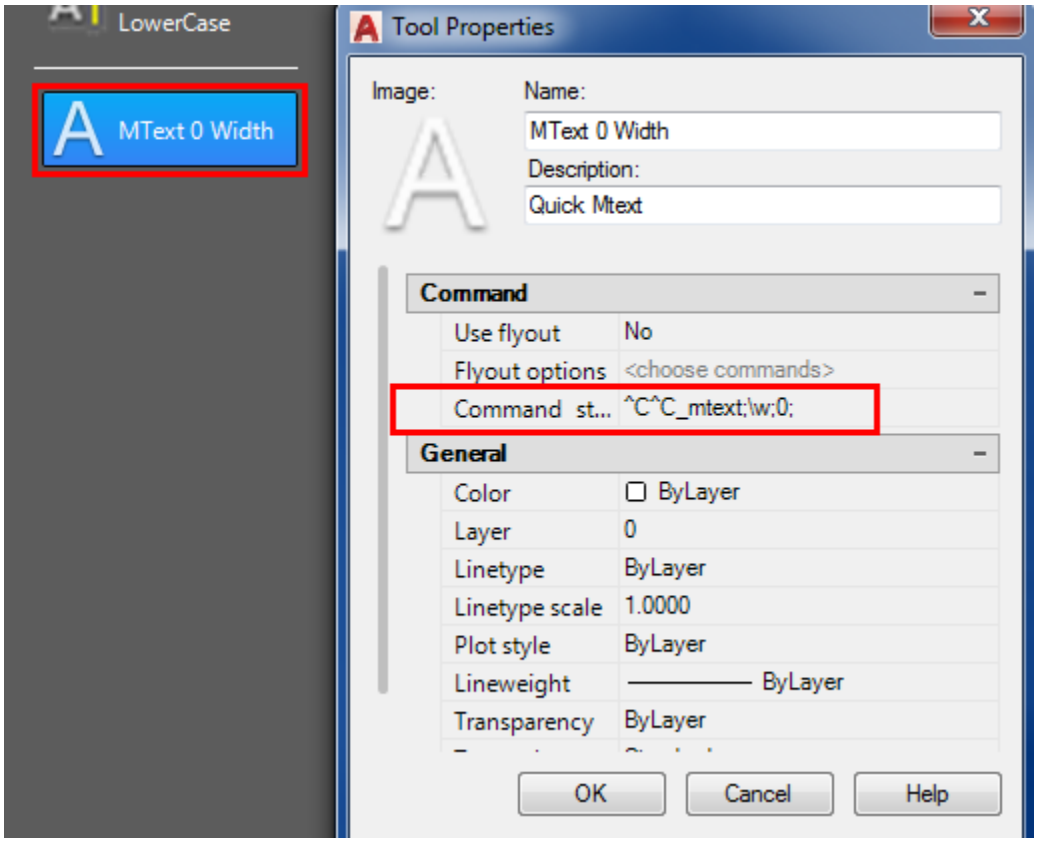
15. You are going to want to change the image at some point but we want to continue focusing on the macro. You could potentially copy that command 5 times and set for each individual option within the tcase command as shown changing only one letter as shown on the command line below.



Exercise 6: Quick Mtext.

Have you ever typed the Mtext command and just wanted it to begin like dtext? You can do that with a macro.

1. Type any piece of MTEXT in the command window and drag onto your palette.
2. Right click the Mtext object and change the command string as shown.



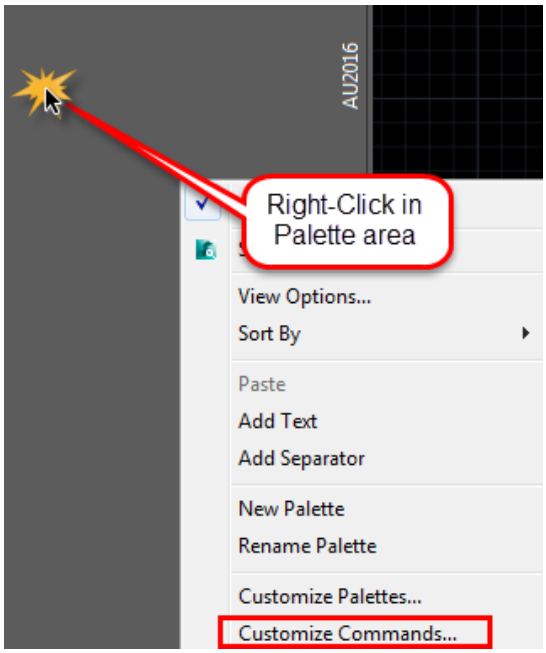
^C^C_mtext;\w;0;

3. You now have a button for Quick Mtext in AutoCAD

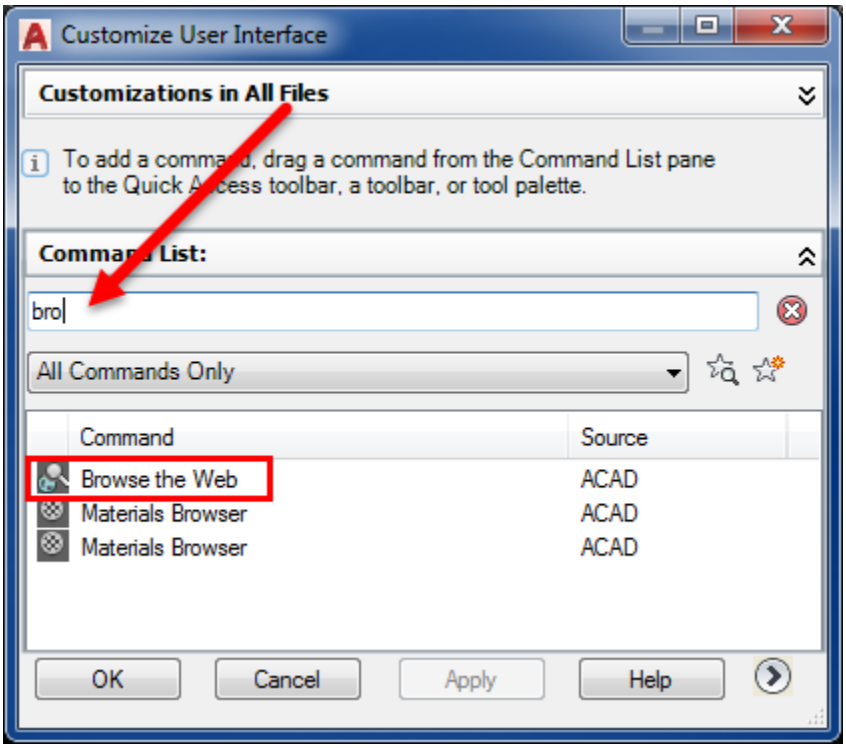
Exercise 7: Open a web page with a Macro

In exercise 7 we are going to setup macros to open up a web browser to the Autodesk University Website. You can use this technique to open up several files including pdf and mp3.

1. Right click in an open area of the tool palette and select Customize Commands…



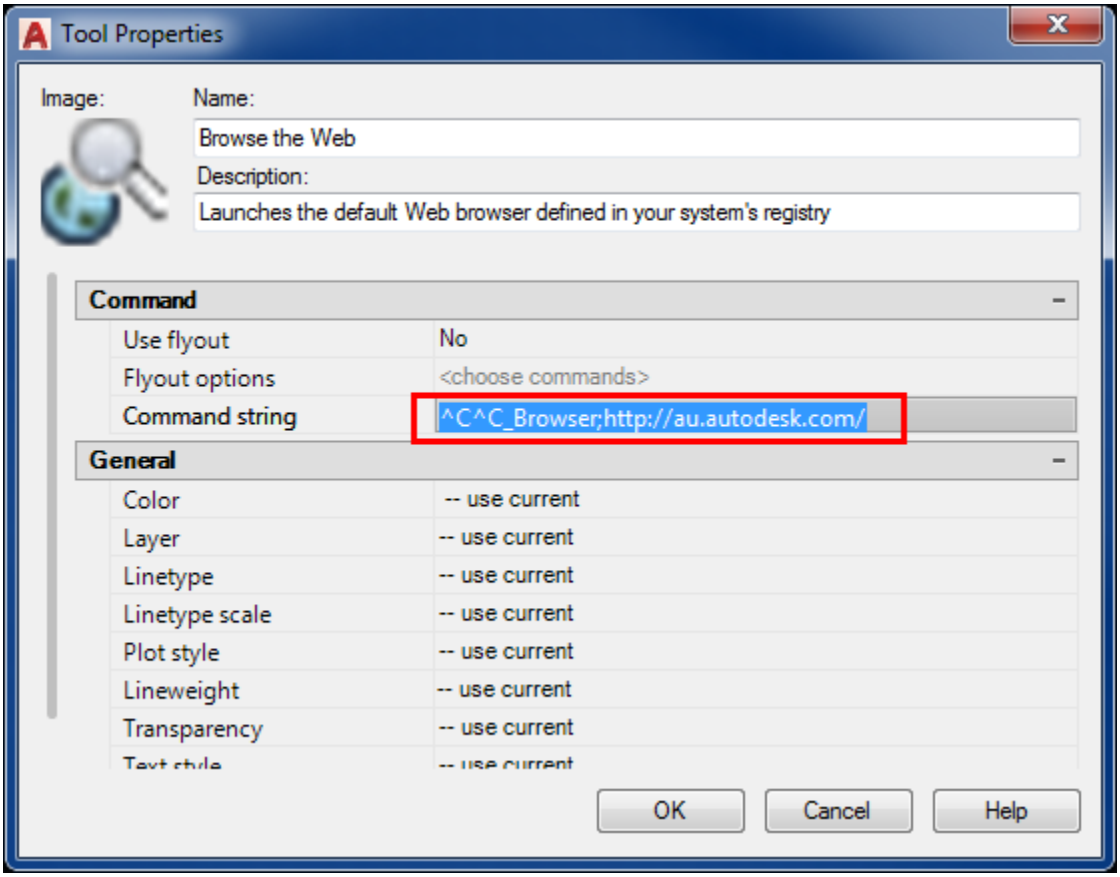
2. You are now in the CUI (customize user interface). Start typing bro…at the command line to locate the browser command in AutoCAD.



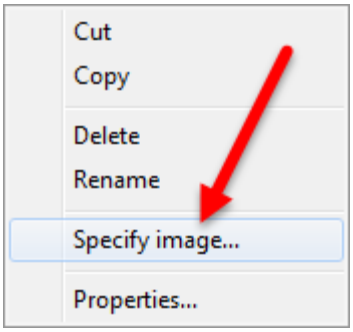
3. Left click and drag the browser command onto the palette as shown.

4. Right-click the command to bring up the properties palette and change the command string to a new macro as shown.

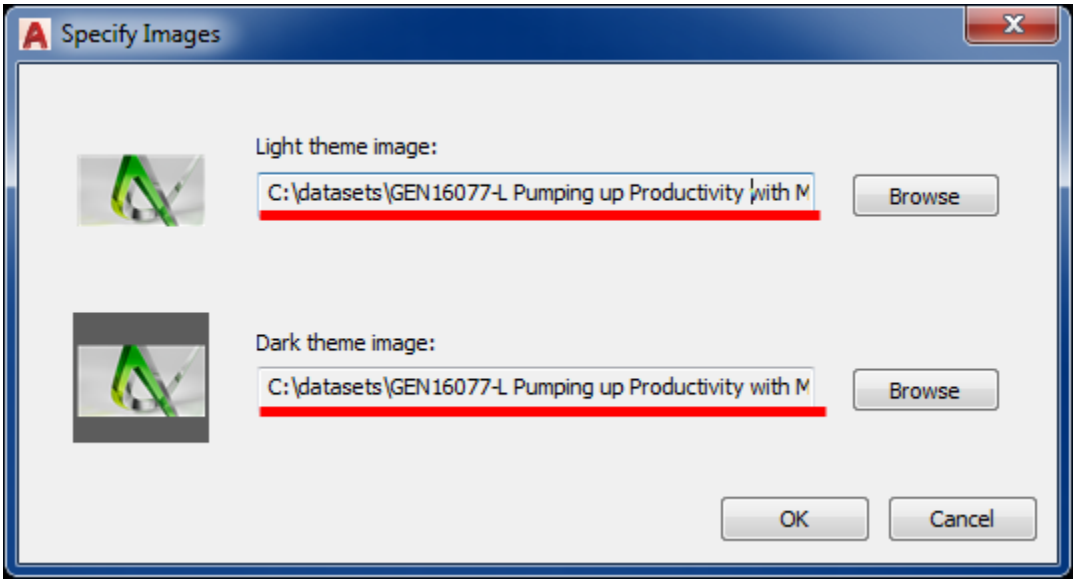
^C^C_Browser;http://au.autodesk.com/



5. Right click the button again and select Specify image…..



6. Browse out to the dataset folder and select the AU2016.jpg file as shown. Select the same image for the light and dark themes.



7. Hit OK, rename the button to Autodesk University by right-clicking the text.

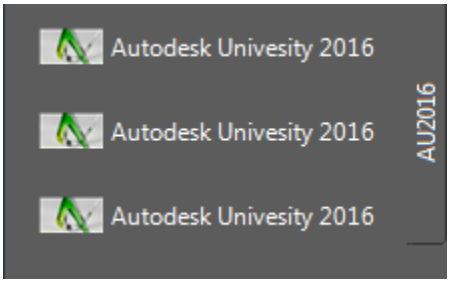
8. Hit the button to launch your new webpage from a macro. See the code as shown below.

Browser Command Macro	
Command	**Description**
^C^C	Cancels any previous action.
_browser	Runs the Browser command.
;	The semi colon represents a return on the keyboard.
http://au.autodesk.com/	Launches default browser pointing to Autodesk University webpage

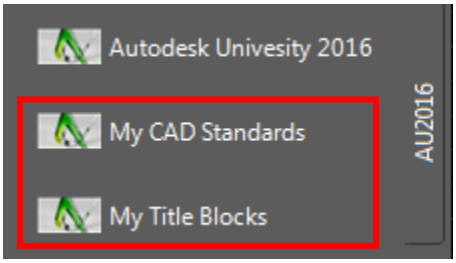
Exercise 8: Open a pdf file and a template from a Macro

In exercise 8 we are going to setup macros to open a pdf file (CAD Standard Manual). How about having all your CAD Standards available to your users with one-click? My advice would be to point to a network location where you can update the file(s) without the user knowing there is an update.

1. Right click your Autodesk University button and copy then paste twice as shown below.

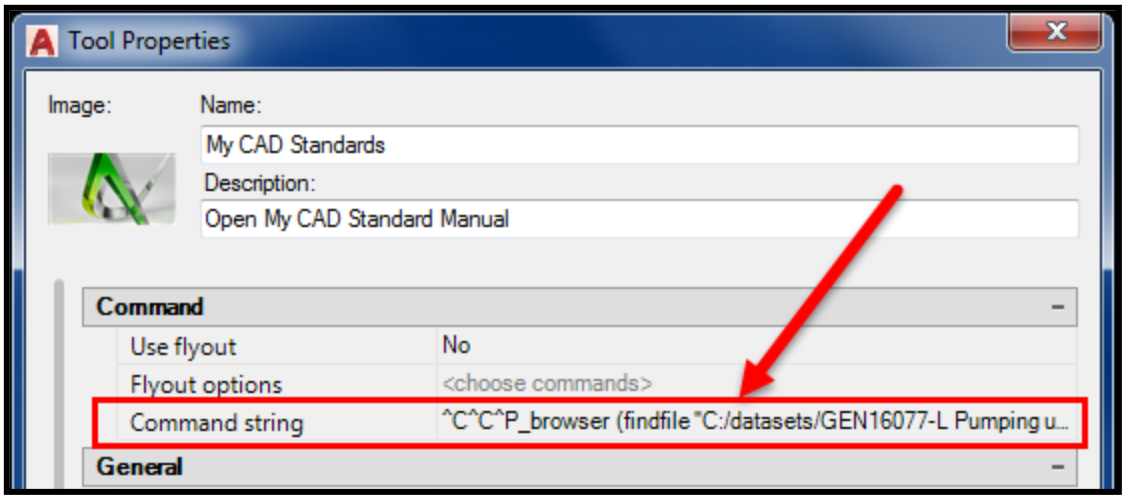


2. Right-click the second button and rename to My CAD Standards then do the same for the third image and name it My CAD Templates.



3. We are now going to right-click My CAD Standards and add the following macro to the command string.

^C^C^P_browser (findfile "C:/datasets/GEN16077-L/CAD_STD_AU2016.pdf")



4. Let's break down the code.

Browser Command Macro	
Command	**Description**
^C^C	Cancels any previous action.
^P	Turns MENUECHO on or off
browser	Runs the Browser command.
;	The semi colon represents a return on the keyboard.
^C^C^P_browser (findfile "C:/datasets/GEN16077-L/CAD_STD_AU2016.pdf")	The macro as shown above to open the pdf file from the dataset included in the class.

My Company CAD Standards Manual: ^C^C^P_browser (findfile "C:/datasets/CAD_Standards/CAD.pdf")

Note: This Macro has been updated to a new path. After further testing, the file path for the Macro must not contain spaces to work in all browsers. The class name has been removed and underscores added to the title to ensure that a clean path is loaded.

5. We are now going to right-click My Title Blocks and add the following macro to the command string.

^C^C^R_layout _template;"C:/datasets/GEN16077-L Pumping up Productivity with Macros/template/AU2016_Template.dwt"

6. Let's break down the code.

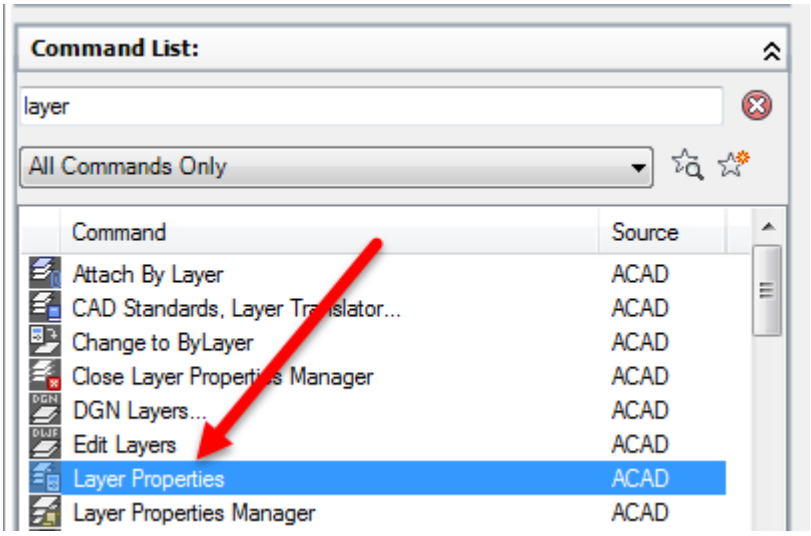
Layout Template Macro	
Command	**Description**
^C^C	Cancels any previous action.
^R	Command Versioning
Layout_Template	Layout_Template command
;	The semi colon represents a return on the keyboard.
"C:/datasets/GEN16077-L Pumping up Productivity with Macros/template/AU2016_Template.dwt"	The macro as shown above to open the template from the dataset included in the class.

My Company CAD Standards Manual:
"C:/datasets/GEN16077-L Pumping up Productivity with Macros/template/AU2016_Template.dwt"")

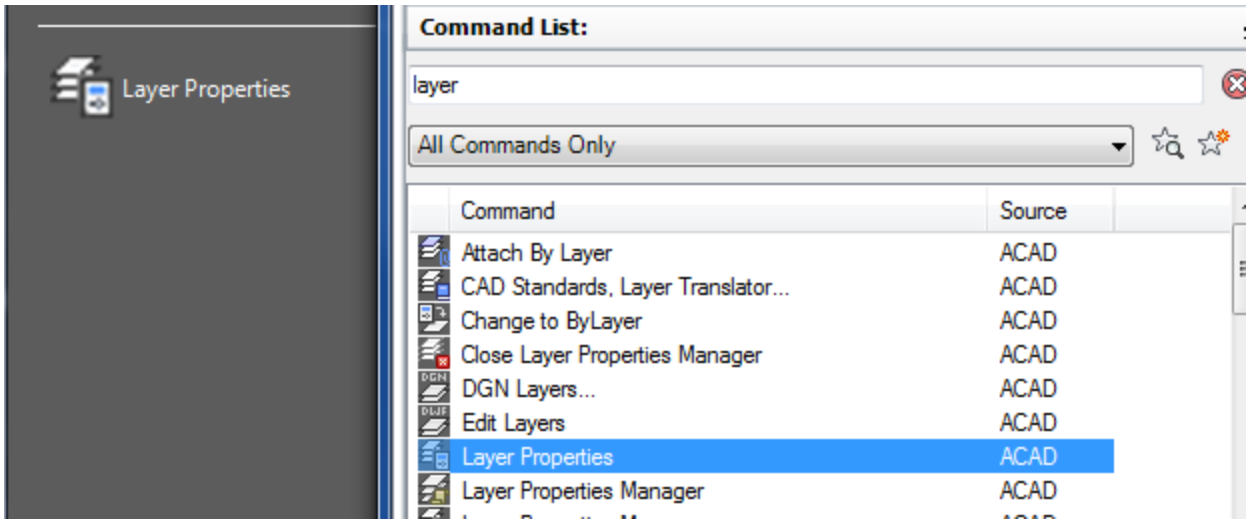
Exercise 9: Create/Reset a Layer with a Macro

In exercise 9 we are going to use a macro to create and/or reset a default layer.

1. Type CUI at the command prompt then layer in the command list.

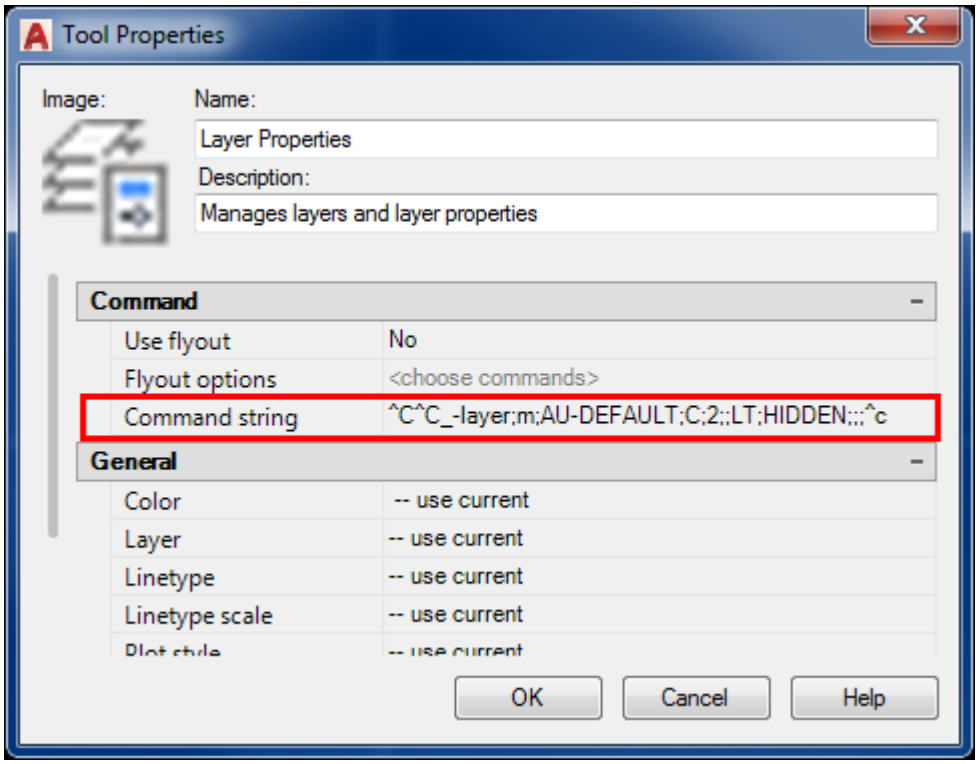


2. Left click and drag the Layer Properties Command onto your Palette. Note: we only want to get the image and the start of the command in there. Saves time!



3. Right-Click you layer command and change the macro as shown.

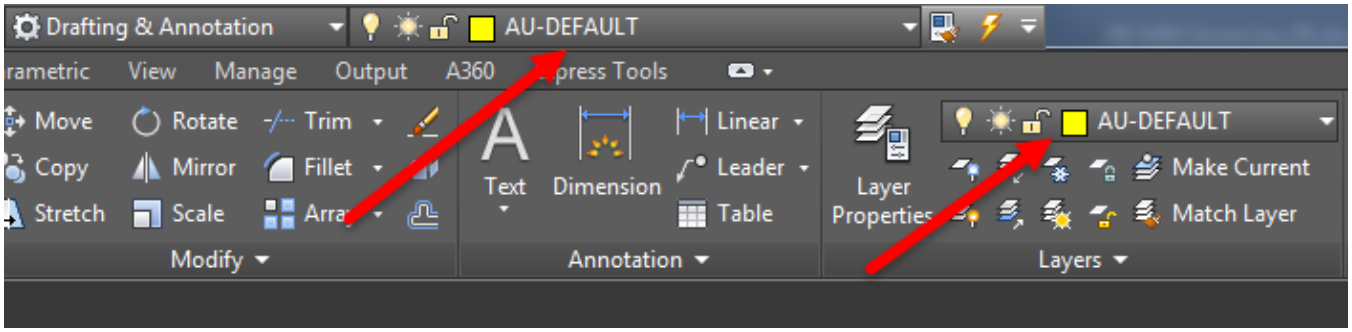
^C^C_-layer;m;AU-DEFAULT;C;2;;LT;HIDDEN;;;^C



Follow the sequence as is on the command line. Notice the number of returns which are represented by a semi-colon in the command string.

Command: -LAYER
Current layer: "0"
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Plot/Freeze/Thaw/
LOck/Unlock/stAte/Description/rEconcile]: m
Enter name for new layer (becomes the current layer) <0>: AU-DEFAULT
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Plot/Freeze/Thaw/
LOck/Unlock/stAte/Description/rEconcile]: C
New color [Truecolor/COlorbook] : 2
Enter name list of layer(s) for color 2 (yellow) <AU-DEFAULT>:
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Plot/Freeze/Thaw/
LOck/Unlock/stAte/Description/rEconcile]: LT
Enter loaded linetype name or [?] <Continuous>: HIDDEN
Enter name list of layer(s) for linetype "HIDDEN" <AU-DEFAULT>:
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Plot/Freeze/Thaw/
LOck/Unlock/stAte/Description/rEconcile]:

4. Test out your new layer command. Change the color of you layer and watch it reset when you run the macro.

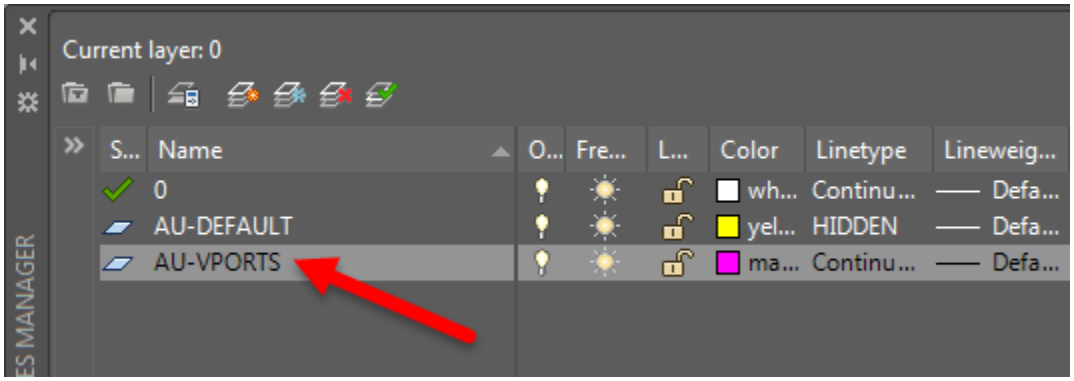


Notes:

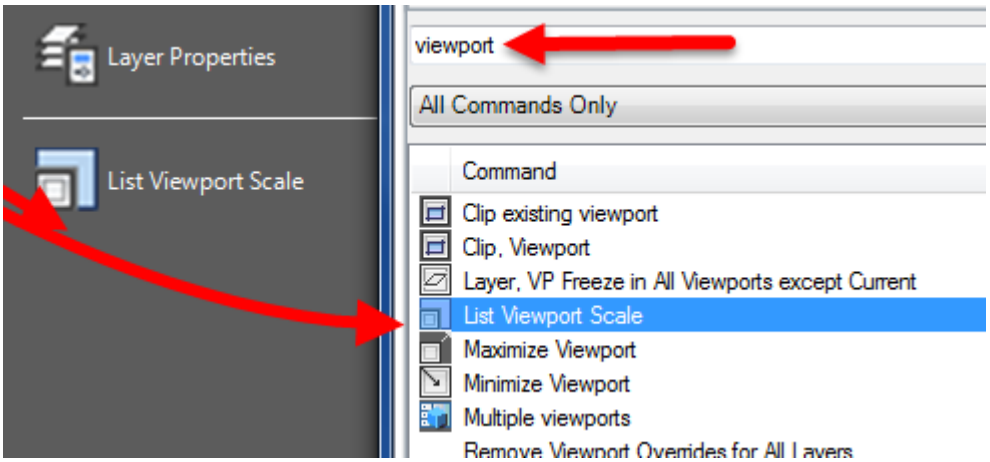
Exercise 10: Create Floating Viewports

In exercise 10 we are going to draw 3 objects, a circle, ellipse, and rectangle. We are then going to convert those objects to floating viewports.

1. Make a layer named AU-VPORTS, give it a color, and set the layer to current.



2. Type CUI at the command prompt then viewport. Again, we are just getting the command in there with an image that is similar. You can customize all of your button images using the button editor as shown in.

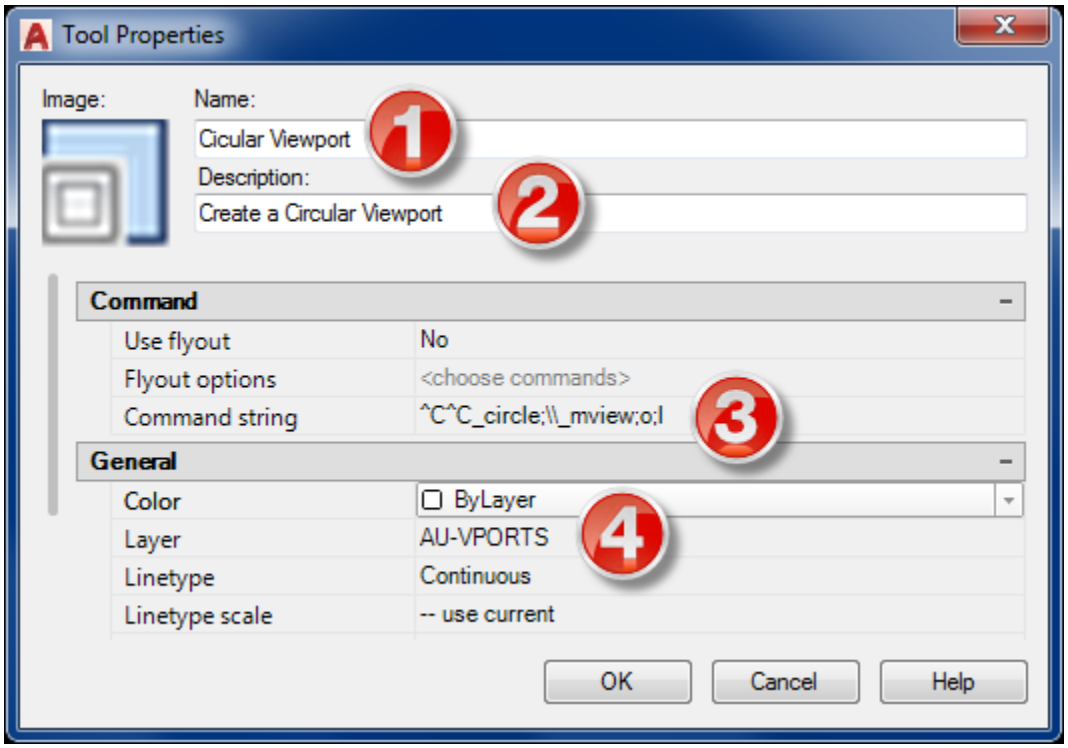


3. Right Click your command and follow these 4 steps.

    1. Change the name to Circular Viewport
    2. Give a description to the new command.
    3. Enter the macro as shown.

^C^C_circle;_mview;o;l

    4. Set the layer to AU-VPORTS.



5. Move to your layout tab and test your code.

Circular Viewport Macro	
Command	**Description**
^C^C	Cancels any previous action.
_circle	Draw a circle.
;	The semi colon represents a return on the keyboard.
****	2 forward slashes. Remember to draw the circle we need 2 clicks therefore 2 pauses for user input at the command prompt.
_mview	Run the Mview command
;	Return
o	Object (the circle)
;	Return
l	Last (selecting the last object drawn)

6. Right click and copy your command and create two additional commands following the same steps as previously entered.

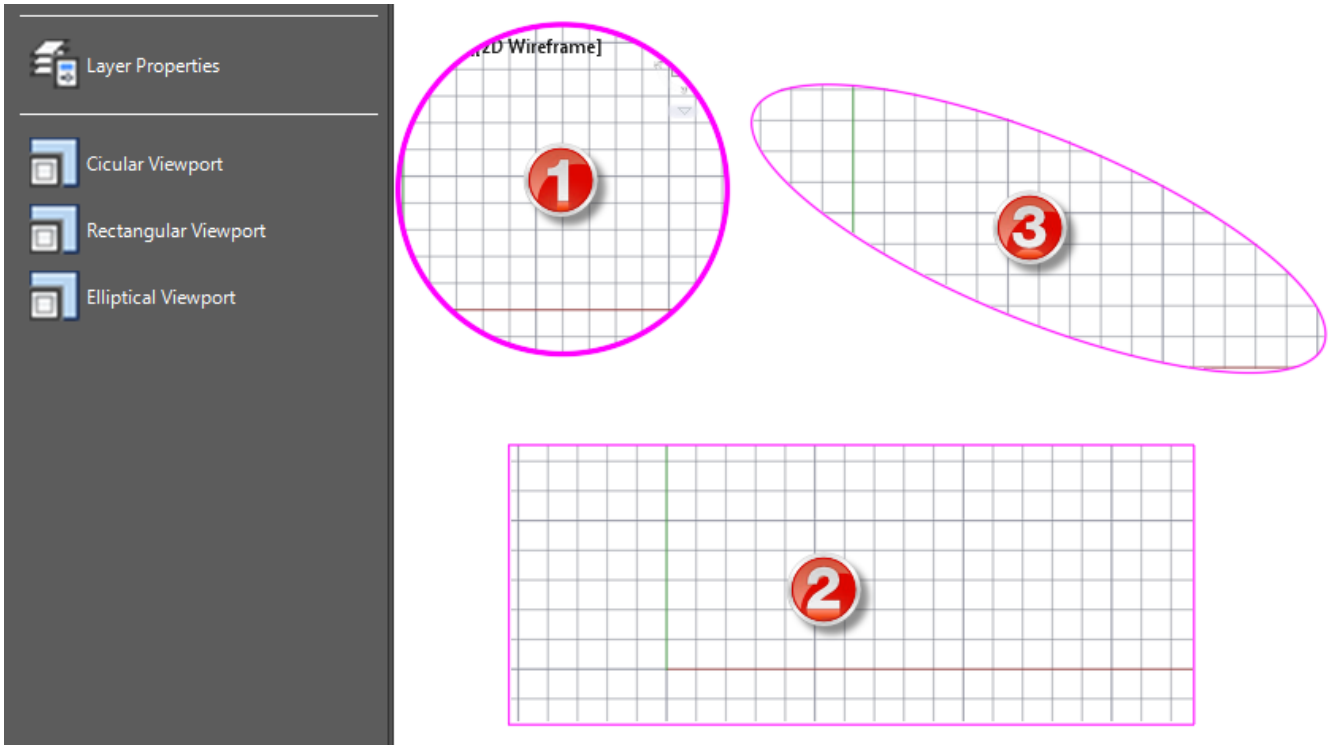
7. Create the rectangular viewport. Everything is the same except we add one more forward slash. It takes 3 clicks to draw the ellipse. How simple is that!

^C^C_circle;_rec;o;l

8. Create the elliptical viewport. Everything is the same except we add one more forward slash. It takes 3 clicks to draw the ellipse.

^C^C_circle;_ellipse;o;l

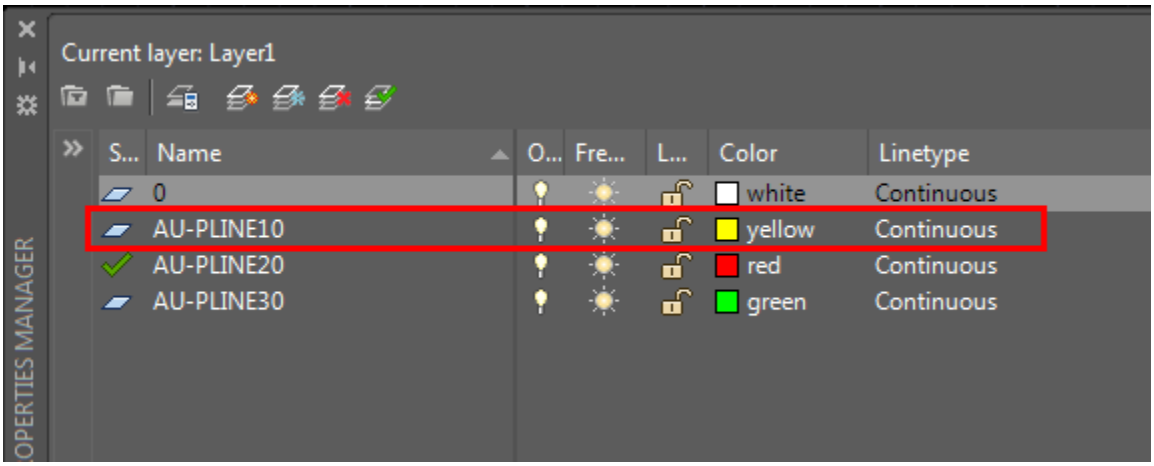
9. Test out all of your new viewports on any layout tab. You have added the layer to the tool which means you do not have to create that layer in a new or existing drawing.



Exercise 11: Draw a Polyline with a specific width

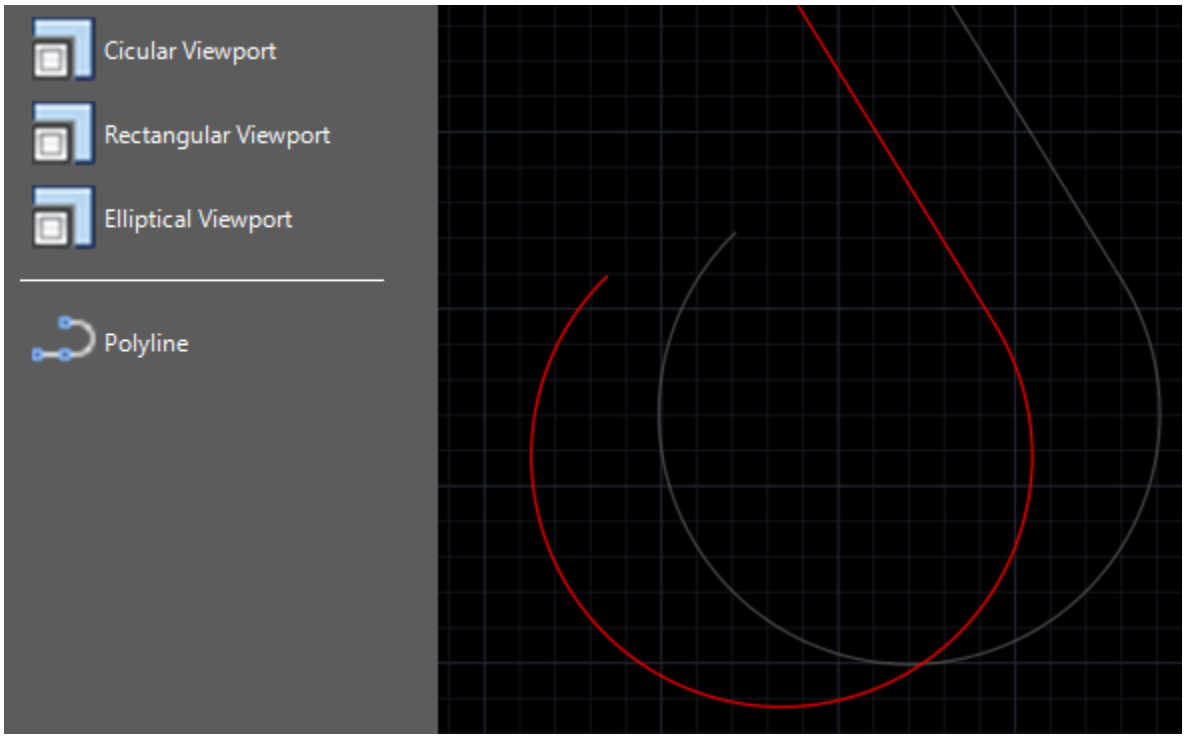
In exercise 11 we are going to create a macro to draw a polyline with a width.

1. Make a layer named AU-PLINE10, give it a color, and set the layer current. Notice how I have created 2 additional layers. These layers are for you to practice with other pline widths.

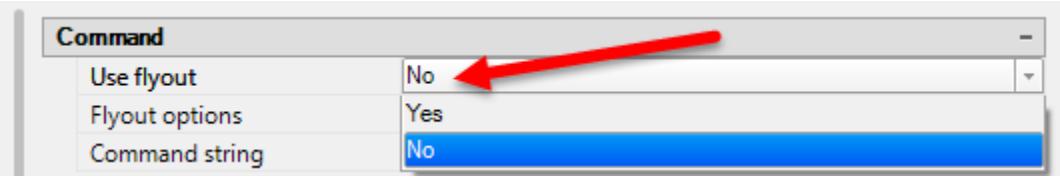


2. Type Pline at the command prompt and draw a polyline.

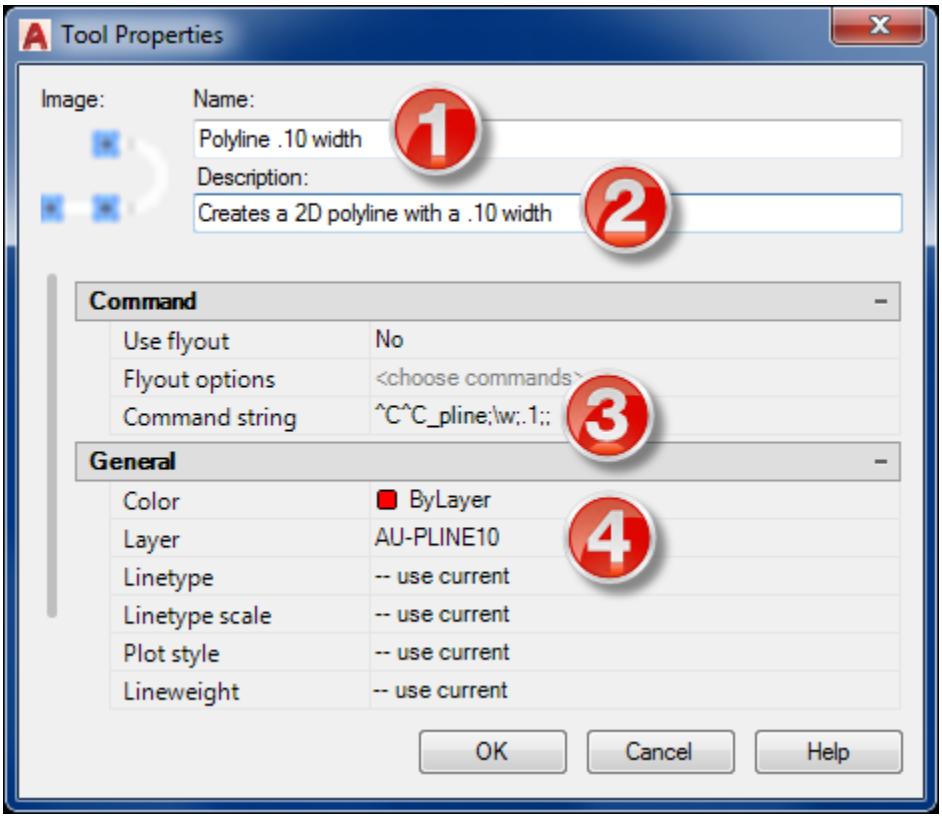
3. Left-click and drag your polyline onto your new tool palette.



4. Right Click your command and change the flyout option to none.



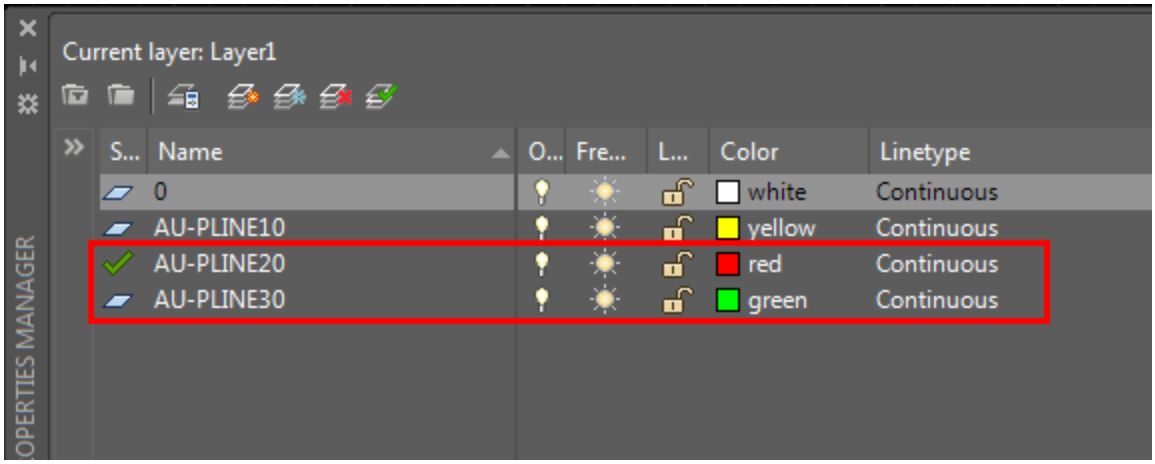
5. Complete the 4 steps as shown below.



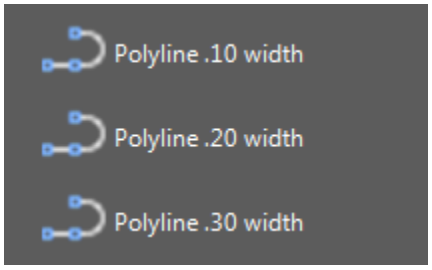
^C^C_pline;\w;.10;;

5. Move to your layout tab and test your code.

6. Copy the image two more time and change the width of the pline in the macro and the layers as shown in below. Create the first one with a .20 width and the second a .30 width and the layers as shown.



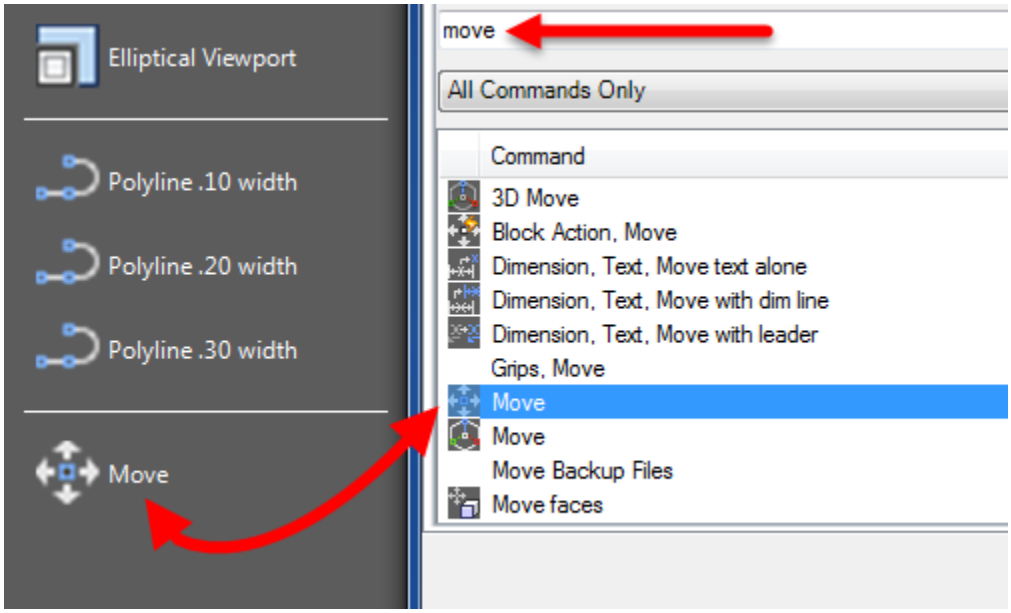
7. Your final tool palette should look as shown.



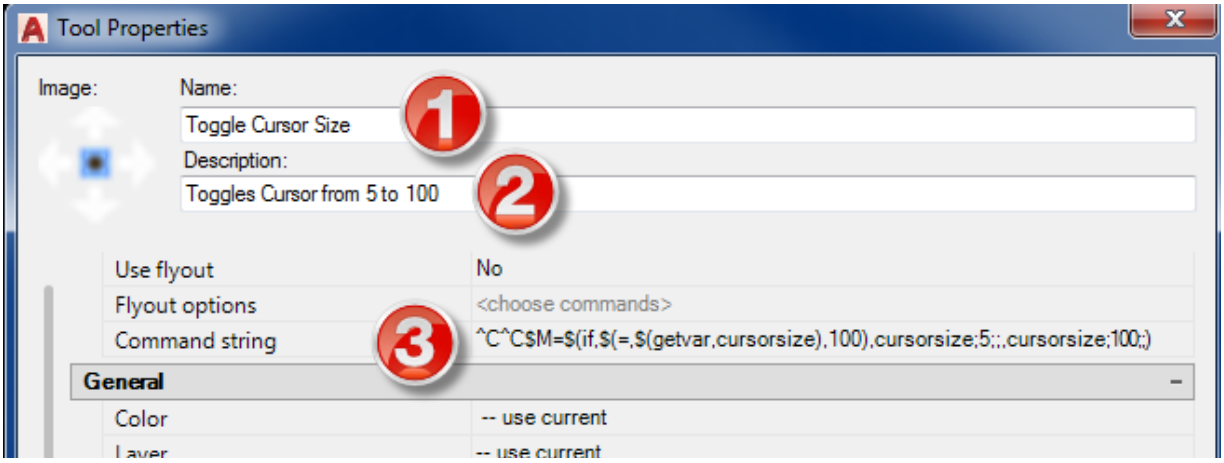
Exercise 12: Toggle Cursor size with Diesel

In exercise 12 we are going to use **DIESEL** programming to create a macro to toggle your cursor size. The toggle cursor size issues commands that use Diesel. Take a look at random commands in the CUI and look for an expression beginning with a **$**. That is a good indicator that diesel is used in the command as explained in the previous section. Notice how the macro shown below uses the cursor variable size and runs an "if" statement, if it is 100 then switch to 5 and vice versa.

1. Type CUI at the command prompt and search for Move. This icon looked similar to a cursor therefore we are going to use that for our macro.



2. Right-click the command and change the following 3 items as shown below.



3. Let's review the DIESEL code as shown below.

^C^C$M=$(if,$(=,$(getvar,cursorsize),100),cursorsize;5;;,cursorsize;100;;)

4. We still begin the macro with a ^C^C then we start our diesel string with the $M=. Notice how the command is followed by an if-then statement enclosed in parenthesis. The command gets the current value of the system variable and changes to the other value if it is equal.

5. Homework: Use another system variable to replace cursorsize and toggle the switch.

The History of Diesel by Ralph Grabrowski

Then with AutoCAD Release 12 for DOS, Autodesk introduced the fully customizable status line. Unfortunately, the user couldn't simply select options from a dialog box. Instead, the user needed to learn *Yet Another Programming Language*, this one called "Diesel" and the sixth different programming interface added to AutoCAD at the time.

Short for "direct interactively evaluated string expression language," the programming logic of **Diesel** is as clear as the acronym's meaning. Despite the word "string," Diesel mostly operates on numbers, not strings. While its purpose is to customize the status line, Diesel has found its way into menu macros and became the most powerful programming environment available in AutoCAD LT -- much to the chagrin of Autodesk, who deliberately disabled the AutoLISP that was supposed to ship with LT because its retailer was worried LT's low price would cannibalize sales of full- blown AutoCAD. Despite the handicap, European programmers have done some amazing things for LT third-party software with Diesel's limited facilities.

Is Diesel a true programming language? For me, the line of differentiation between a macro language and a programming language is whether there is logical functions, such as **If**, **While**, or even **Greater Than**, etc. (Logic functions make it possible for the program to make decisions.) Diesel has logic functions but the syntax is so obscure that it begs to be known as a simple macro language -- and that's how I'll refer to it from now on.

What Diesel Does

Diesel allows me to change AutoCAD's status line so that it displays other useful information, such as the z-coordinate, the DWG filename, and the time. There is a limitation, though: the text displayed by Diesel is truncated after 32 characters, no matter how big I make the window (39 characters in Release 13 for Windows and LT for Windows 95). Diesel has an unusual format for its macro language. Every function begins with the dollar sign and a bracket:

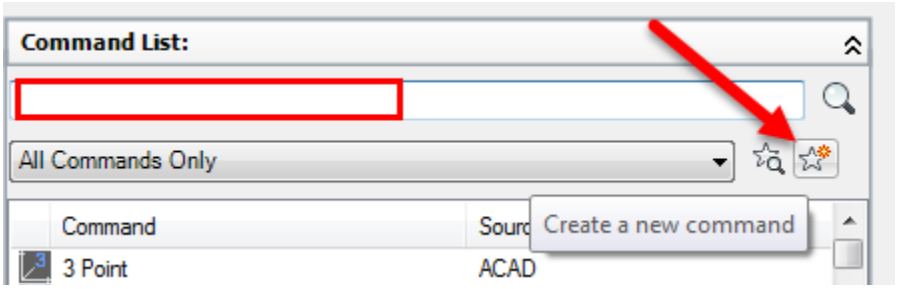
$(function, variable)

No doubt, the purpose of the $-sign is to alert the AutoCAD command processor that a Diesel expression is on the way, just as the (-symbol alerts AutoCAD that an AutoLISP expression is coming up. The opening and closing parentheses signals the beginning and end of the function. This allows Diesel functions to be nested, where the variable to one function is another function. The parentheses also allow Diesel to work on more than one variable at a time; the closing parenthesis alerts Diesel that there aren't any more variables.

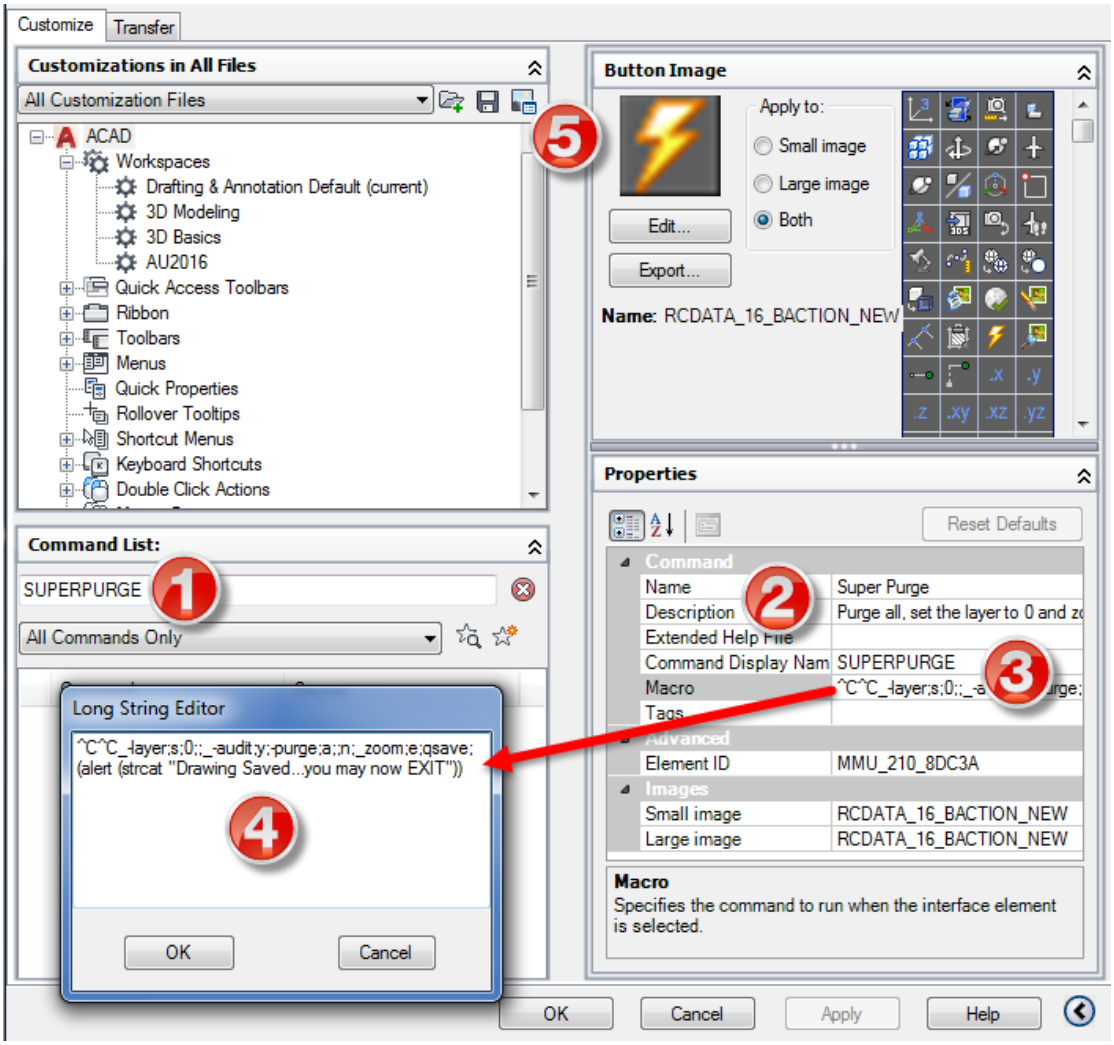
Exercise 13: Super Purge

In exercise 13 we are going to create a macro to finalize our drawing session using an audit then a purge. This time we are going to create a new command, a button on the QAT

1. Type CUI at the command prompt and select the star as shown to create a new command.



2. We are now going to follow the 5 steps as shown to create our new command.

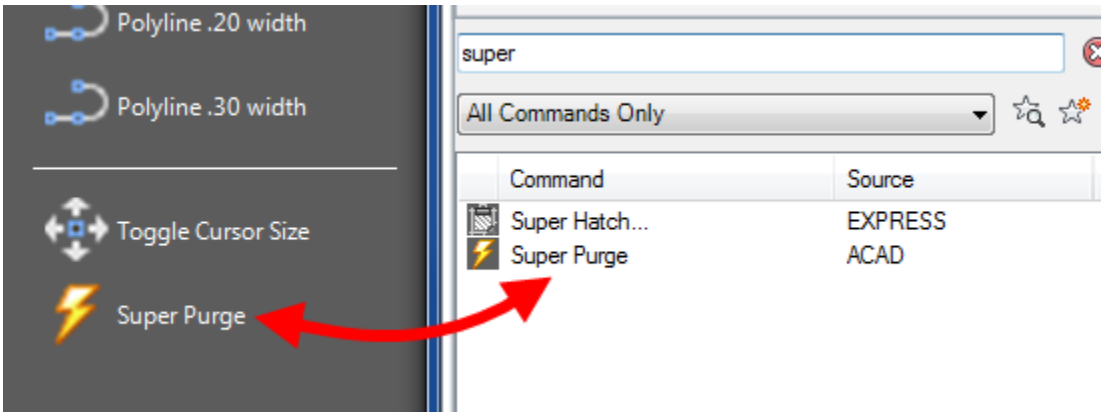


1. Name our command to SUPERPURGE (one word).
2. Name your command Super Purge and give it a description.
    a. Purge all, Set the layer to 0 and zoom to the extents of the drawing.
3. Enter SUPERPURGE for the command display name.
4. Click the ellipses at the end of the command window and enter the macro as shown below.

```
^C^C_-layer;s;0;;_-audit;y;-purge;a;;n;_zoom;e;qsave;(alert (strcat "Drawing Saved...you may now EXIT"))
```

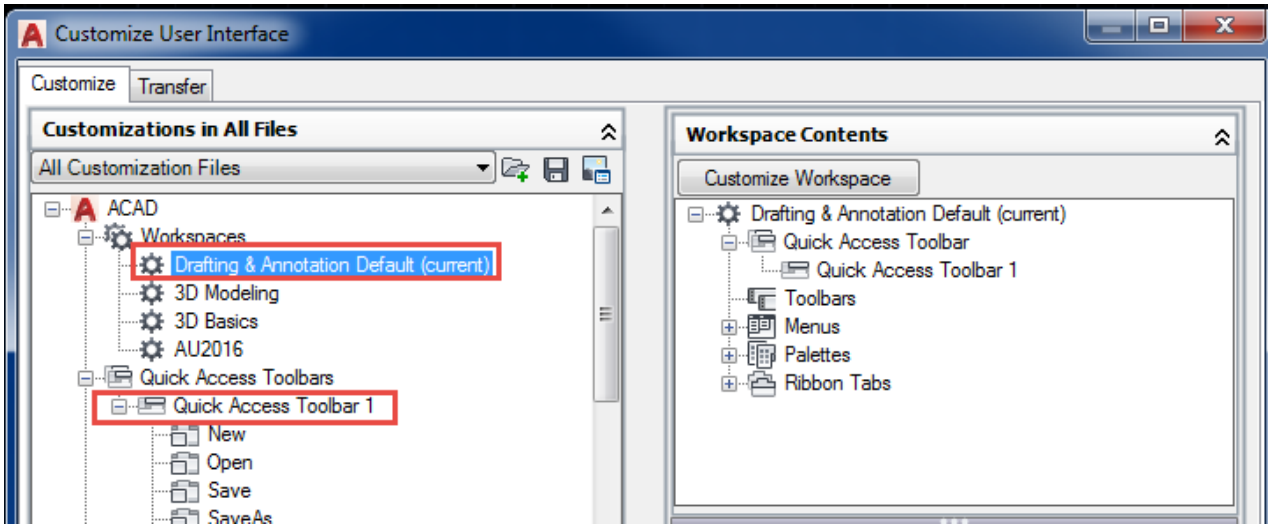
5. Select any button image you feel suits the needs of your new command.
6. Hit apply-OK and close the command window.

3. We have now created a new command.
4. Type CUI at the command prompt again and enter SUPER in the command list to show your command.
5. Left click and drag that command onto our AU tool palette.

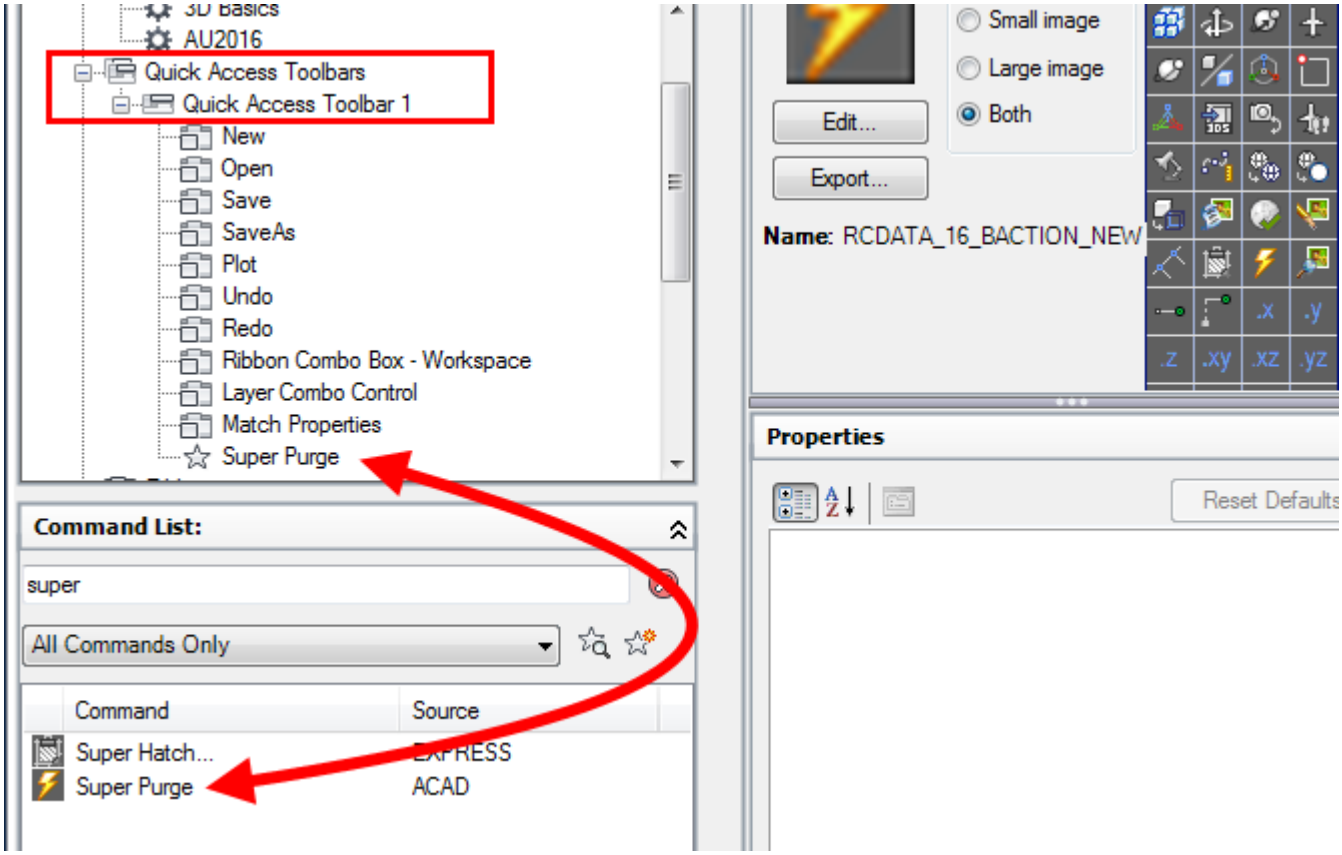


6. We are not done yet. Let's add that command to our current QAT.
7. Keeping the same window open (or reopen the CUI and follow step 4 above.

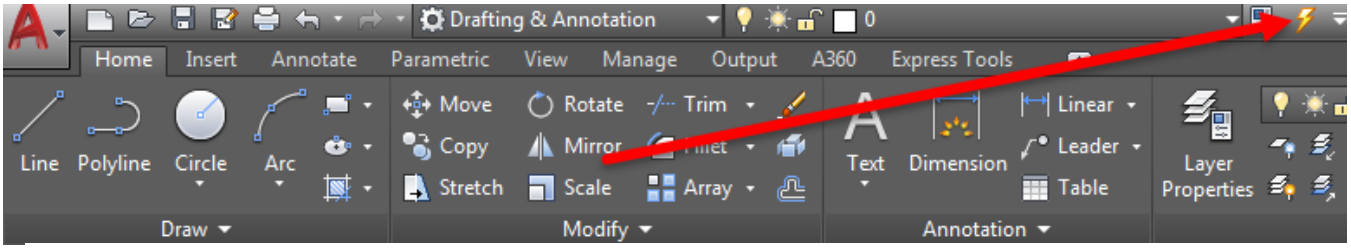
8. Pull out the current QAT as shown by unchecking the + next to quick access toolbars.



9. You are now going to drag your command into the current QAT as shown below. Select the superpurge command and drag anywhere into your QAT.



10. Success! You have now added your command to the QAT and to the AU Toolbar.



Conclusion

A macro can be defined as a way to automate a task that you perform repeatedly with more than one command or keystroke. In my opinion macros are the easiest way to program with little knowledge of programming. In AutoCAD® macros can be shortcuts to a series of commands to help make the process of design more efficient. In this class we learned how to use the Action Recorder, create and modify Tool Palettes using macros, and we jumped right into the CUI capturing existing code to create new macros in AutoCAD®.

All of these tools help us as professionals become more efficient and perform our job at a high level. Take a moment and think about workflow and what you do every day at work. Can you take four clicks and turn those into one? Think about how many times you apply that task in one day, then in one year. Being efficient and productive will only make you more valuable and provide profit to the company you work for. Today we pumped up productivity with some cool new macros one character at a time.

Time is money and you have just learned how to save a little of both.

Thank you!

"It's not who I am underneath but what I do that defines me" BATMAN

Enjoy the rest of your time at Autodesk University 2016!

AUTODESK UNIVERSITY 2016

Pumping up Productivity in AutoCAD with Project PRAXIS

PRAXIS is a cloud-based app for sharing your product knowledge and best practices through simple, step-by-step workflow diagrams.

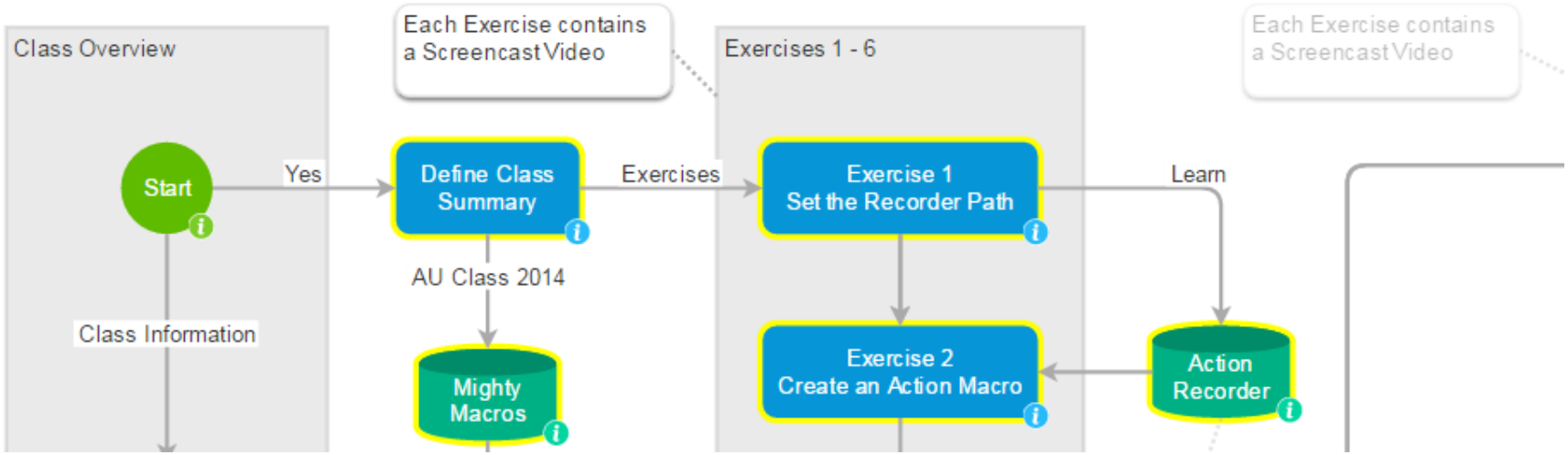
- There's nothing to download and install. Because PRAXIS runs on your browser, you always have access to the most up-to-date version.
- PRAXIS works on Google Chrome, version 20 and later, on Windows and Mac OS. We plan to support other browsers in the future.
- There is no concept of files in PRAXIS. Your work is saved automatically to the cloud.
- For the time being, anyone can view a shared workflow, but only in Read-Only mode. Private sharing is in the planning stages.

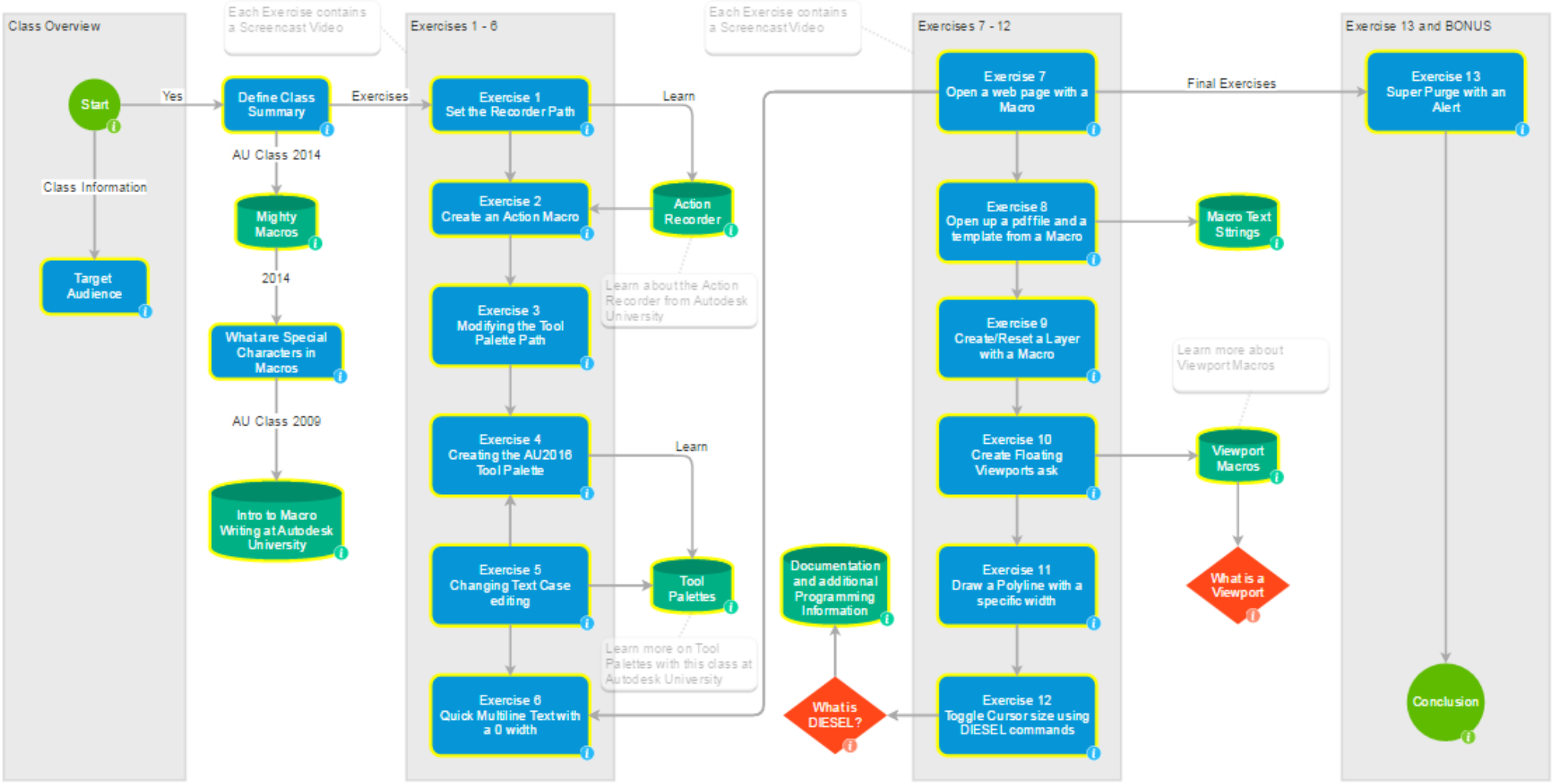
These goals informed the design of PRAXIS:

- Communicating your thought processes quickly and easily. PRAXIS is fast and intuitive and does not get in the way of your thought and communication.
- Clearly giving the context behind each step of the workflow—no confusing long commentaries to explain your process.
- Easily sharing your work with team members. Others see your process through a browser, without installing software or downloading documents to share.

Note: Use in tablet or smartphone browsers is not currently supported

https://praxis.autodesk.com/praxis/viewer/96e32e06bec22b972c6f1432f5c5ebdd5c630139




Class Overview
Start
Yes
Define Class Summary
Exercises
Class Information
Target Audience
AU Class 2014
Mighty Macros
2014
What are Special Characters in Macros
AU Class 2009
Intro to Macro Writing at Autodesk University
Each Exercise contains a Screencast Video
Exercises 1 - 6
Exercise 1 Set the Recorder Path
Learn
Exercise 2 Create an Action Macro
Action Recorder
Learn about the Action Recorder from Autodesk University
Exercise 3 Modifying the Tool Palette Path
Exercise 4 Creating the AU2016 Tool Palette
Learn
Exercise 5 Changing Text Case editing
Tool Palettes
Learn more on Tool Palettes with this class at Autodesk University
Exercise 6 Quick Multiline Text with a 0 width
Each Exercise contains a Screencast Video
Exercises 7 - 12
Exercise 7 Open a web page with a Macro
Final Exercises
Exercise 8 Open up a pdf file and a template from a Macro
Macro Text Strings
Exercise 9 Create/Reset a Layer with a Macro
Exercise 10 Create Floating Viewports ask
Learn more about Viewport Macros
Viewport Macros
What is a Viewport
Exercise 11 Draw a Polyline with a specific width
Exercise 12 Toggle Cursor size using DIESEL commands
Documentation and additional Programming Information
What is DIESEL?
Exercise 13 and BONUS
Exercise 13 Super Purge with an Alert
Conclusion

Companion Text

The following tables are used along with the GEN16077_L lab handout at Autodesk University 2016. These tables include macros for use in Autodesk AutoCAD. Some of the Macros will be developed during the class, others are for your use and testing. File paths will need to be changed to your company standard; examples shown are for reference and guidance on creating macros in AutoCAD.

Table 1 - Special Characters in Macros	
Character	Description
;	Issues Enter
^M	Issues Enter
^I	Issues Tab
[blank space]	Enters a space; a blank space between command sequences in a command is equivalent to pressing the Spacebar
****	Pauses for user input (cannot be used with accelerators)
. (period)	Allows you to access a built-in command even if it was undefined using the UNDEFINE command. (Not Available in AutoCAD LT)
_	Translates commands and options that follow
=*	Displays the current top-level pull-down, shortcut, or image menu
***^C^C**	Repeats a command until another command is chosen
$	Introduces a conditional DIESEL macro expression ($M=)
^B	Turns Snap on or off (equivalent to Ctrl+B)
^C	Cancels the active command or command option (equivalent to Esc)
^D	Turns Dynamic UCS on or off (equivalent to Ctrl+D)
^E	Sets the next isometric plane (equivalent to Ctrl+E)
^G	Turns Grid on or off (equivalent to Ctrl+G)
^H	Issues Backspace
^O	Turns Ortho on or off
^P	Turns MENUECHO on or off
^Q	Echoes all prompts, status listings, and input to the printer (equivalent to Ctrl+Q)
^R	Turns command versioning on or off. Command versioning is required for some commands to ensure command macros written in an older release work properly in the latest release.
^T	Turns tablet on or off (equivalent to Ctrl+T)
^V	Changes the current viewport
^Z	Null character that suppresses the automatic addition of Spacebar at the end of a command

Table 2 - Change Text Case	
Command	**Description**
^C^C	Cancel any previous action
_-TCASE	Runs the text case (tcase) command. Be sure to include the hyphen in front to suppress the dialog box.
;	The semi colon represents a return on the keyboard
****	Pauses for user input (select the text object)
;	Ends the "select objects" mode (Issues a return or enter)
U	Uppercase selection
;	Issues an enter and ends the command

Macro: ^C^C_-Tcase;\;U;

Table 3 - Browser command	
Command	**Description**
^C^C	Cancels any previous action.
_browser	Runs the Browser command.
;	The semi colon represents a return on the keyboard.
http://au.autodesk.com/	Launches default browser pointing to Autodesk University webpage

Macro 1 - AU Website:
^C^C_Browser;http://au.autodesk.com/

Macro 2 - Knowledge Network:
^C^C_browser;http://usa.autodesk.com/adsk/servlet/index?id=7460177&siteID=123112

Macro 3 - Company CAD Standards Manual:
^C^C^P_browser (findfile "C:/datasets/GEN16077-L/CAD_STD_AU2016.pdf")

Table 4 - Title Block Template	
Command	**Description**
^C^C	Cancel any previous action.
^R	The ^R will issue the new version of the command.
_layout_template	This command is taken from the cui to insert a layout from a template
;	Issues a return on the keyboard
"C:/path"	Path to your template file. Notice that the slashes are forward and not the traditional backward slash. That is because the backward slash pauses for user input in a macro.

^C^C^R_layout _template;" C:/datasets/GEN16077-L Pumping up Productivity with Macros/AU2016_Template.dwt"

Table 5 - Import Page Setups	
Command	**Description**
^C^C	Cancel any previous action.
filedia	System variable to turn off dialog boxes
;	Issues a return on the keyboard
0	Turn off dialog box
;	Issues a return on the keyboard
^R	The ^R will issue the new version of the command.
_psetupin	Import Page setup command
;	Issues a return on the keyboard
"C:/AU2013/…"	Path to your template file. Notice that the slashes are forward and not the traditional backward slash. That is because the backward slash pauses for user input in a macro.
;	Issues a return on the keyboard
filedia	System variable to turn off dialog boxes
1	Turn those dialog boxes back on!

^C^C_filedia;0;^R_psetupin;"C:/AU2013/Mighty Macros/Class Files/Title Blocks/ AU-ENG-2013.dwt";filedia;1

Table 6 – Multileader Styles	
Command	Description
^C^C	Cancel any previous action.
_-insert	Insert command with the special character (-) preceding to suppress the dialog box.
"C:/path"	Path to your multileader template file. Notice that the slashes are forward and not the traditional backward slash. That is because the backward slash pauses for user input in a macro.
;	Issues a return on the keyboard
^C^C	At this point you want to cancel the command since the drawing settings have already been inserted within the drawing
_multileaderstyle	Runs the multileader style command. You certainly do not have to run the style command. This has been added to show the class the styles that have been imported into our file
;	Issues a return on the keyboard

^C^C_-insert;"C:/AU2013/Mighty Macros/Class Files/03 Multileaders.dwg";^C^C_mleaderstyle;

GEN16077_AU2016_Macros in AutoCAD

Share Report

18 ITEMS

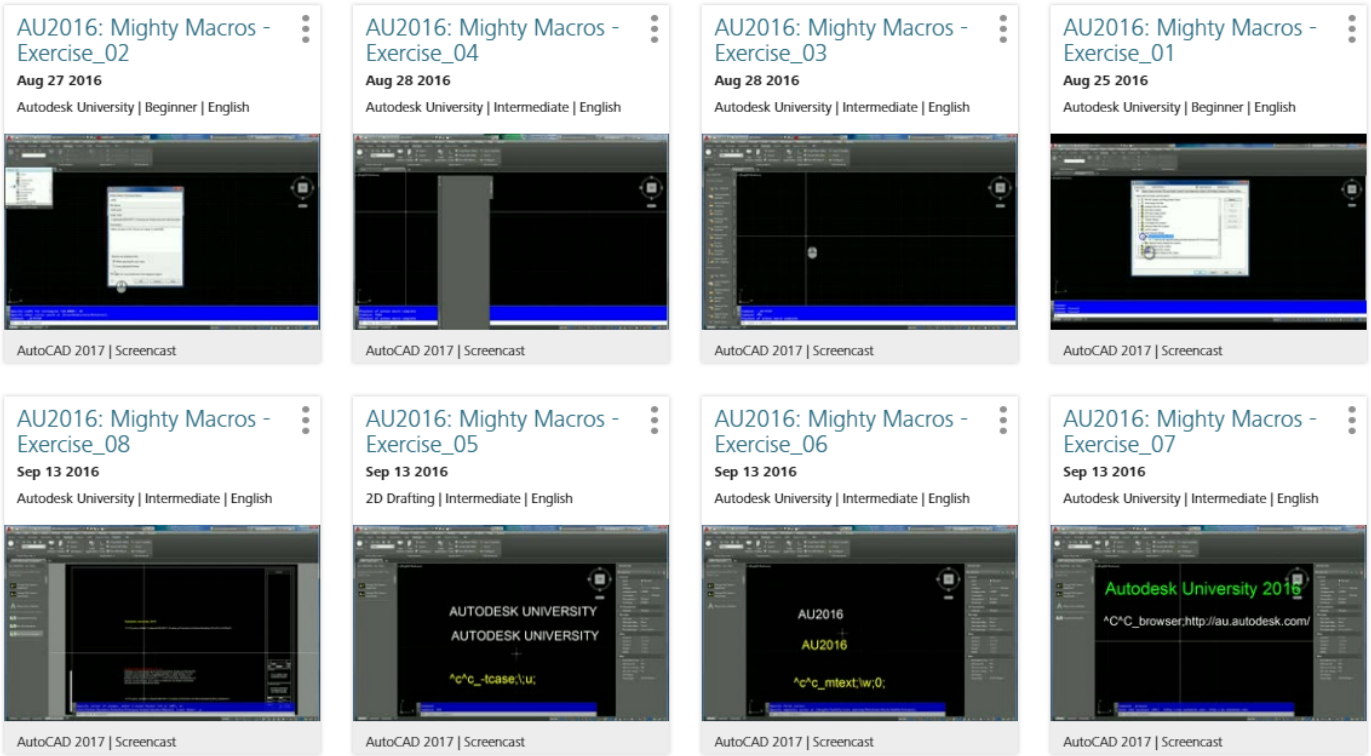


Table 7 - System Variables Setup	
Command	Description
^C^C	Cancel any previous action.
cecolor	cecolor system variable
;	Issues a return on the keyboard
bylayer	Sets the variable to bylayer
;	Issues a return on the keyboard
msltscale	This variable scales linetypes displayed on the model tab by the annotation scale. This is a great way to see your linetypes as they are displayed both in paper and model space
;	Issues a return on the keyboard
1	Enters the value for the system variable
Continue to enter all of the system variables that you typically set within your drawing file. Use the same format and input the variable followed by a semi-colon	
mtjigstring	This variable sets the content of the sample text displayed at the cursor location when the MTEXT command is started
alert	Issues the alert command to let the operator know that the commands have been loaded

```
^C^C_cecolor;bylayer;celtype;bylayer;cmdecho;0;mirrtext;0;menuecho;0;olestartup;1;
peditaccept;1;plinegen;1;lwdefault;0;visretain;1;psltscale;1;msltcale;1;imageframe;0;
savetime;5;mtjigstring"(getvar "loginname"));(alert (strcat "System Variables Set!"))
```

Note: I have added an alert lisp function at the end to alert the user that the variables have been changed. You can delete that but nothing will show up when you select the button, sometimes it's good to be sure we made the right pick!

Quick Editing with Macros

Table 8 - Rotate 90d then repeat	
Command	Description
*	Repeat the macro when action has been performed
^C^C	Cancel any previous action.
_rotate	Starts the Rotate command
;	Issues a return on the keyboard
\	Pause for user input
;	Issues a return on the keyboard
@	The @ symbol in AutoCAD will retrieve the last picked point
;	Issues a return on the keyboard
90	Enters the rotation angle of 90d
;	Issues a return on the keyboard

*^C^C_rotate;\;@;90;

Table 9 - Rotate 1d then repeat
*^C^C_rotate;\;@;1;

Table 10 - Polyline with a specified width
^C^C_PLINE;\W;.1;;

Table 11 - pline with a specified width and layer
^C^C_-layer;m;newline;c;2;;;_PLINE;\W;.1;;

The snap angle macro will select a line using the object snap nearest point(s) and changing the cursor angle to that point. This can be very helpful when drawing things to a specific angle. We need to reset the snap angle back

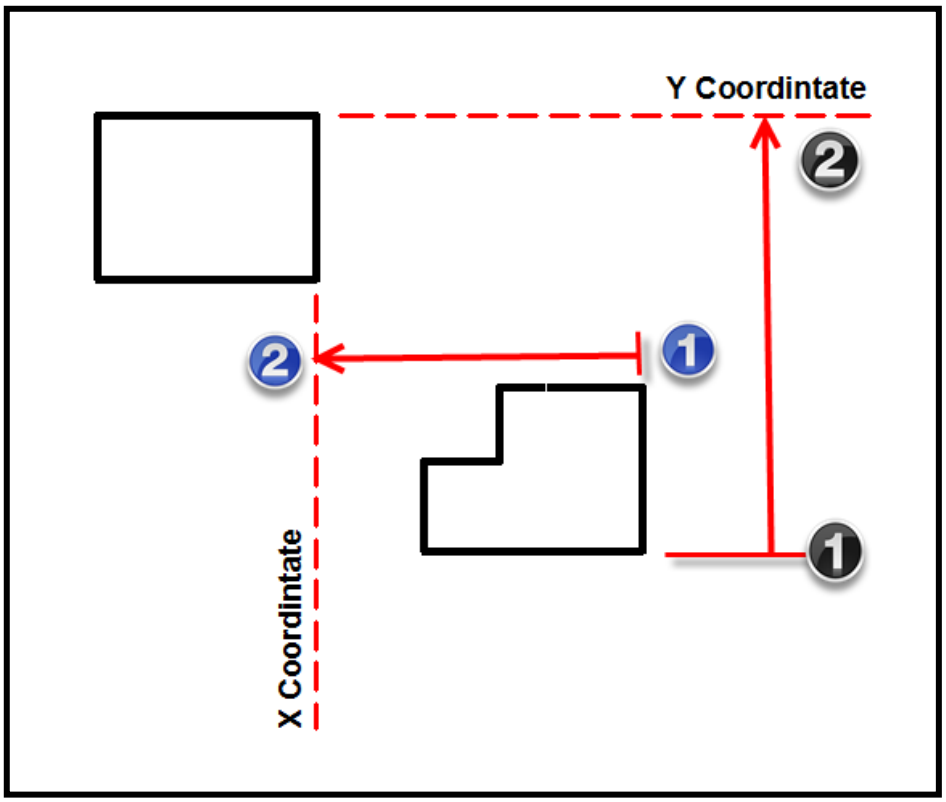
to 0 which is why the following button is directly below it. These macros use the object snap code of 512 which is the nearest (yes you can use the transparent command near). You can enter these values in a macro or create your own by simply entering the correct integer.

Table 12 - Set the snap angle to an object
^C^C_OSMODE;512;_SNAPANG;\\;;

Let's run another macro to reset it back.

Table 13 - Reset back to 0
^C^C_SNAPANG;0;

The next two commands use point filters and are one of the most forgotten things in the 2D world. Most of us who do not work in the 3D world sometimes forget about how valuable point filters can be in 2D. Point filters are used in conjunction with base point selections for the move, rotate and scale commands as well as any other command requiring a base point.



Both macros show in Tables 14 and 15 start off by changing the object snap mode to an endpoint. You could issue a few osnaps to be sure you are grabbing the correct point. Select your object to move along with the x coordinate. The end of these use the object snap code of 1024 to clear all snap settings.

Table 14 - Move an object using X point filter
^C^Cosmode;1;_move;\;\.x;\OSMODE;1024

Table 15 - Move an object using Y point filter
^C^Cosmode;1;_move;\;\.y;\OSMODE;1024

The next three macros on our list try to accommodate solutions for a few different scenarios. The first is to draw a building using the rectangle command with dimensions, and then hatch that building with an annotative hatch. The second will break a line at a pick point and change the first part of that line to hidden. The last one is a quick macro on how to make the mtext editor just a step quicker by entering a 0 mtext width.

Table 16 - Draw a building then hatch the building
^C^C_layer;m;C-BLDG;c;5;;;_rectang;\d;\\\-layer;m;C-BLDG-HTCH;c;8;;; -bhatch;s;l;;an;y;p;ansi31;1;0;;

Next up is changing or breaking a line at a point using the nearest osnap then taking that line and making the original section hidden. You could break this macro down further to change layers. Many times, in the design process you will need to keep a line on the same layer and just change it to hidden, it does not happen often but it does happen.

Table 17 - Breaks a line at a point then changes the original line to hidden
^C^C_OSMODE;512;_break;\f;\@;chprop;l;;lt;hidden;;OSMODE;1024

This next command is a launch the mtext command with a 0 width. It just gives the operator more of a dtext feel if they still are having a hard time transitioning from single line text. See Table 17a below for the command string.

Table 18 - Quick Mtext Start
^C^C_Mtext;\w;0;

Changing the Cursor Size with a Macro

For years, I have been using a lisp routine to switch the cursor size from 3 to 100, flipping back and forth. Sometimes we need that cursor size to be small so some areas of our drawing are not obstructed. I decided to try this same concept by using a macro. The first macro will set the cursor size to 100 and the one below will change back to a 3. Table 18 shows how this simple macro can change that system variable quickly.

Table 19 - Change the cursor from 100 to 3
^C^C_CURSORSIZE;100
^C^C_CURSORSIZE;3

Toggle Cursor Size using Diesel

Table 20 - DIESEL to toggle the cursor size from 3 to 100
^C^C$M=$(if,$(=,$(getvar,cursorsize),100),cursorsize;3;;,cursorsize;100;;)

The next command will toggle between paper space and model space. This specific macro is designed to be used with one viewport and can be found in the AutoCAD help file. Much like if you double click inside the view to activate the viewport. This macro will save you a keystroke by switching from paper to model quickly.

Table 21 - DIESEL to toggle between paper and model space
^C^C^P$M=$(if,$(=,$(getvar,cvport),1),mspace,pspace)

The clean screen command (CTRL+0) cleans the screen in AutoCAD. Look at the code as shown below in Table 22. Grab that code and let's alter it. We want a clean screen but we also want the command line off, menubar off, and the status bar off. One exception we still want to see out tool palettes so we can toggle back. With some altering this can be accomplished in a macro as shown below in Table 22.

Table 22 - Clean Screen with Options
$M=$(if,$(and,$(getvar,CleanScreenState),1),^C^C_CleanScreenOFF;statusbar;1;menubar;1; commandline,^C^C_CleanScreenON;statusbar;0;menubar;0;commandlinehide);tp;

Load and Unload Xrefs quickly

Table 23 - Unload and Reload Xrefs
^C^C-xref;u;*
^C^C-xref;r;*

Cool tricks with Layers

Table 24 - Layer Merge with Name
^C^C^R_laymrg;\;n

Table 24 shows a macro brings up the filter named AU. Yes, you need to have a name of the filter to bring it up using a macro. Maybe you company uses a prefix to define layers. You can then use the macro with the filter name to bring up just those layers quickly.

Table 25 - Load a layer filter
^C^C+layer;AU;

This next macro as shown in Table 25 I added in there since I like to see what layer I am deleting. Deleting a layer but with a dialog box. I want to know the name of the layer I am deleting and confident that my file structure is intact. In this instance, you must use the ^R to get the name dialog box to appear. This is like the layer merge command simply just bringing up the list for you to view.

.

Table 26 - Layer delete Name
^C^C^R_laydel;n

Viewports

You can create a single layout viewport that fits the entire layout or create multiple layout viewports in the layout. Once you create the viewports, you can change their size, their properties, and scale and move them as needed. With MVIEW, you have several options for creating one or more layout viewports. You can also use COPY and ARRAY to create multiple layout viewport.

Table 27 - Lock Viewports in a Layout
^C^C_-vports;l;on;all;;(alert (strcat "All viewports have been locked"))

Table 28 - Unlock Viewports in a Layout
^C^C_-vports;u;on;all;;(alert (strcat "All viewports have been locked"))

Table 29 - Create a Circular Viewport
^C^C_circle;_mview;o;l

Table 30 - Create an Elliptical Viewport
^C^C_ellipse;_mview;o;l

Table 31 - Create a rectangular Viewport
^C^C_rec;_mview;o;l;

Altering the UCS with Macros

Both macros shown can be found in his book. Michael's Corner can be found at www.cadtutor.net and you can purchase the book through his site www.cadtrainerguy.com, scroll down at the bottom of all the pages. Michael's book provides hundreds of tips and tricks including some macros in there to help increase productivity. Michael has been an inspiration to me and many others as his style is very easy to follow and well thought out.

Table 32 - Align the UCS to an Object
^C^C_UCS;_OB;_PLAN;;

Table 33 - Reset the UCS back to World
^C^C_UCS;_W;_PLAN;;

Table 34 - Load a Script file to reset the Scale List
^C^C_filedia;0;_SCRIPT;"C:/AU2014/Mighty Macros/Class Files/Script files/AU-scale-dec.scr";filedia;1

Table 35 - Load a Script file to reset and load an Engineering Scale List
^C^C_filedia;0;_SCRIPT;"C:/AU2013/Mighty Macros/Class Files/Script files/AU-scale-eng.scr";filedia;1

Table 35 - Load a Script file to reset and load the Architectural Scale List
^C^C_filedia;0;_SCRIPT;"C:/AU2013/Mighty Macros/Class Files/Script files/AU-scale-arc.scr";filedia;1

Table 36 - Set Dimension standards and set the style with a macro
dim;dimasz;.14;dimtsz;.05;dimexe;.05;dimcen;.05;dimgap;.06;dimdli;.08;dimdle;.00;dimexo; .12;dimtp;.06;dimtm;.06;dimdle;.00;dimtxt;.10;dimtad;1;dimtih;0;dimclrt;bylayer;dimclre; bylayer;dimclrd;bylayer;dimsho;1;dimtofl;1;dimaso;1;_sav;AU_STD;Y;exit;redraw;ddim;

Table 37 - Accessing Open Save Tab using Options
^C^C+options;2;

Table 38 - Bring up the current drawing folder
^C^C^P(startapp "explorer" (strcat "/n,/e," (getvar "dwgprefix")))

Table 39 - Bring up the current drawing folder
^C^C^P(startapp "explorer" (strcat "/n,/e," (getvar "savefilepath")))

Table 40 - Bring up the current drawing folder
^C^C^P(startapp "explorer" (strcat "/n,/e," (getvar "_toolpalettepath")))

Table 41 - Create a Revision Cloud
^C^C_revcloud;a;(if (= (getvar "CTAB") "Model") (* (getvar "dimscale") 0.3) 0.3);;s;n;layerp;

Table 42 - Create a Revision Cloud from a Rectangle
^C^C_RECTANGLE;_revcloud;a;(if (= (getvar "CTAB") "Model") (* (getvar "dimscale") 0.3) 0.3);;S;N;O;;L;N;layerp;

Table 43 - Create a Revision Cloud from a Rectangle and insert a Marker
^C^C_RECTANGLE;_revcloud;a;(if (= (getvar "CTAB") "Model") (* (getvar "dimscale") 0.3) 0.3);;S;N;O;;L;N;_insert;"C:/AU2013/Mighty Macros/Class Files/01 Revision Cloud Marker.dwg";^C^C_-layer;m;Rev-marker;c;2;;;-insert;revm;s;(if (= (getvar "CTAB") "Model") (* (getvar "dimscale") 1));\;;^C^Clayerp;;

Table 44 - Launch the Reference Manager
^C^C^P(startapp "adrefman" (strcat "C:/Program Files/Autodesk/AutoCAD 2014/AdRefMan.exe"))

Table 45 - Finalize my Drawing Session
^C^C_-layer;s;0;;_-audit;y;-purge;a;;n;_zoom;e;qsave;(alert (strcat "Drawing Saved...you may now EXIT"))

Table 46 - Linesanity! Load my linetypes with a macro	
Command	Description
^C^C	Cancel any previous action
_-STYLE	Create the style or verify that the style is in the current drawing (this has been added since Linesanity at AU2012)
;	The semi colon represents a return on the keyboard
linetypes	Create a new unique style named "linetypes"
;	The semi colon represents a return on the keyboard
Arial	Set the font to true type Arial
;	The semi colon represents a return on the keyboard
0;1;0;n;n;	This next sequence I grouped together. These are the parameters that set the style. Height, Width, Oblique angle, backwards, upside down. Type -style at the command prompt and follow the sequence.
_-LINETYPE	The command to start loading the linetypes. The hyphen will disable any dialog boxes
;	The semi colon represents a return on the keyboard
L	Selects the option "Load"
AIR	The name of the linetype to be loaded
;	The semi colon represents a return on the keyboard
…..path/Mylines.lin	Linetype file name and path. Remember the slash will be a backward one instead of the normal forward path. A forward slash will indicate a pause for user input and will cancel the command string.
;	The semi colon represents a return on the keyboard
y	Reload the linetype if it already has been loaded
;	The semi colon represents a return on the keyboard
^C^C	Cancel any previous action
_-LAYER	Selects the layer command

^C^C_style;linetypes;arial;0;1;0;n;n;_-LINETYPE;L;AIR;"C:/AU2012/The Linesanity Project/MYLINES.LIN";y;^C^C_-layer;lt;AIR;;;_pline