

AS468502-L

AutoCAD Customization Boot Camp: Basic (No Experience Required)

Lee Ambrosius
Autodesk, Inc.

Learning Objectives

- Learn how to create command aliases
- Learn how to write and run script files
- Learn how to create and modify tool palettes
- Learn how to modify the ribbon and Quick Access toolbar

Description

Customizing the AutoCAD software or AutoCAD LT software is a great way to increase productivity, but many users are unsure of how to get started. In this lab, you'll learn about several of the basic customization features that you can use to simplify workflows and reduce repetitive tasks. You can customize features such as command aliases and the ribbon to improve access to frequently used commands, and you can create tool palettes and script files to reduce the number of steps required to complete a task.

Speaker(s)

Lee Ambrosius is a Principal Learning Experience Designer at Autodesk, Inc., for the AutoCAD® and AutoCAD LT products on Windows and Mac. He works primarily on the customization, developer, and CAD administration documentation along with the user documentation. Lee has presented at Autodesk University for about 15 years on a range of topics, from general AutoCAD customization to programming with the ObjectARX technology. He has authored several AutoCAD-related books, with his most recent project being *AutoCAD Platform Customization: User Interface, AutoLISP, VBA, and Beyond*. When Lee isn't writing, you can find him roaming various AutoCAD community forums, posting articles on his or the AutoCAD blog, or tweeting information regarding the AutoCAD product.

Twitter: @leeambrosius

Email: lee.ambrosius@autodesk.com

Blog: <http://hyperpics.blogs.com>

1 Introduction

The AutoCAD software is an extensive 2D drafting and 3D modeling program that has grown in functionality since it was first introduced over 35 years ago back in 1982. What sets AutoCAD apart from many other CAD programs is its expansive customization and automation capabilities. The customization and programming features of AutoCAD allow individuals and companies to simplify everyday workflows, such as:

- Initial drawing setup; establish drawing units and format, create layers, insert a title block and populate attribute values
- Extraction of design data for use downstream in a bill of materials or order entry system
- Consumption of project information from a data source such as a spreadsheet or database

This hands-on lab will provide you with the opportunity to roll-up your sleeves and get some experience with customizing AutoCAD and how to apply the techniques covered to your everyday workflows. While knowing how to program isn't a requirement to customizing AutoCAD, learning how to program does provide you with a greater set of tools to automate everyday tasks in AutoCAD.

2 Which Customization and Programming Options are Available

Not all customization and programming options are created equally, some options are easy to learn and well integrated into the AutoCAD program that many don't even realize they are customizing the program. For example, creating new layers and named styles are forms of customization that many frequently perform. There are two types for customization and programming that are available; drawing and application.

The following lists many of the customization and programming options available:

Basic

Drawing	Application
<ul style="list-style-type: none"> • Layers • Annotation styles (text, dimensions, multileaders, and tables) • Blocks • Materials and visual styles • Drawing templates 	<ul style="list-style-type: none"> • Desktop shortcut • Command aliases • Tool palettes • Workspaces • User profiles • Plot styles

Intermediate

Drawing	Application
<ul style="list-style-type: none"> • Dynamic blocks 	<ul style="list-style-type: none"> • Scripts • Action macros • User interface (CUI Editor) • DIESEL • Custom linetypes and hatch patterns • Custom shapes and text styles

Advanced (Application Only)

- AutoLISP / Visual LISP
- Visual Basic for Applications (VBA)
- ActiveX / COM (VBA, VBScript, VB.NET, C#, C++)
- Database connectivity
- Managed .NET (VB.NET, C#)
- ObjectARX (C++)
- JavaScript
- Sheet Set Manager API
- CAD Standards plug-ins
- Transmittal API
- Connectivity Automation API
- Forge Platform APIs

3 What You Need to Get Started

Many of the customization and programming options available for use within the AutoCAD program are supported through utilities or commands found inside the program, or applications installed with the operating system (Windows or Mac OS). It is when you want to extend the functionality of the AutoCAD program using ObjectARX or Managed .NET that you will need to download and install additional software which may have an additional cost. The Forge APIs will require you to obtain a developer account and pay for the services utilized.

4 Command Aliases

Commands are the heart and soul of the AutoCAD program, but they take time to type or locate in the user interface (UI). While there have been many improvements to the AutoCAD UI since its first release, the Command prompt remains the fastest and most efficient way to execute a command. Command aliases are an alternative to typing a command's full name and they have been fairly consistent over the past 20+ releases. Because of their efficiency and consistency, many AutoCAD veterans prefer to use them to start commonly used commands.

Command aliases are stored in a Program Parameters (PGP) file named *acad.pgp* (or *acadlt.pgp* for AutoCAD LT users) which can be edited using a plain text editor, such as Notepad on Windows or TextEdit on Mac OS. Two separate pieces of information are required when defining a command alias; an abbreviation or the text to be entered at the Command prompt and the name of the AutoCAD command to be started by the command alias.

Each command alias has the follow syntax:

```
abbreviation, *command
```

Here are some examples of commonly used command aliases:

```
C,      *CIRCLE
E,      *ERASE
L,      *LINE
P,      *PAN
Z,      *ZOOM
```

When defining new command aliases, it is recommended to place them at the bottom of the PGP file to make it easier to identify which command aliases need to be migrated after upgrading to a newer release. If you define a command alias that has the same abbreviation as another command alias in the PGP file, the last definition of the command alias found within the PGP file is the one loaded into memory and made accessible from the Command prompt.

The following shows an example of redefining the C command alias to start the COPY command rather than the CIRCLE command:

```
C, *CIRCLE
E, *ERASE
C, *COPY
```

Tip: If you are using AutoCAD, not AutoCAD LT, you can use the Alias Editor (ALIASEDIT command) that is part of the Express Tools to define and modify the command aliases in the *acad.pgp* file. The Alias Editor is not available in AutoCAD for Mac OS.

Command aliases can only be used to start a command, they can't be used to start a command with a specific sequence of options or values. A command can be executed with a specific sequence of options and values a script, action macro, or custom AutoLISP function. AutoLISP functions can be used to define commands that feel similar to a command alias, but have the ability to pass options and values to a command. You can find out more about action macros and AutoLISP in the AutoCAD Online Help system or refer to the companion session "AS468504-L AutoCAD Customization Boot Camp: Automate Workflows and Tasks."

The AutoCAD (not AutoCAD LT) PGP file can also contain special command aliases known as *external command aliases*. An external command alias can be used to start an installed application, which you might normally launch outside of the AutoCAD program. External command aliases are not as commonly used these days since it is much easier to launch an application from the Windows or Mac OS user interface (UI), as opposed to shelling out to a MS DOS or Bash Shell prompt.

The following is an example of an external command alias defined with the name NOTEPAD and is used to start the Windows Notepad application:

```
NOTEPAD,    START NOTEPAD,  1,*File to edit: ,
```

For more information about external command aliases, see the comments at the beginning of the AutoCAD PGP (*acad.pgp*) file.

Note: External command aliases are not supported on AutoCAD for Mac OS.

Additional Information

You can find additional information on command aliases with these topics in the AutoCAD Online Help system:

- [About Command Aliases](#)
- [About Creating Command Aliases](#)
- [About Defining External Commands](#)

E1 Define Custom Command Aliases

This exercise explains how to define a command alias to start the REVCLLOUD command, override the C command alias to start the COPY command rather than the CIRCLE command, and create a new alias for the CIRCLE command.

See the Exercises section in the handout for this hands-on lab.

5 Scripts

Scripts are one of the oldest ways of automating repetitive tasks in the AutoCAD program. A script contains a sequence of commands to execute along with the options and values that those commands expect. AutoLISP expressions may also be included in a script; AutoLISP isn't available in AutoCAD LT. Scripts are stored in ASCII text files that can be created with a plain text editor, such as Notepad that is installed with Windows or TextEdit on Mac OS and they must have the .scr file extension.

There are some limitations to which commands and AutoLISP expressions that can be executed by a script file and how many scripts can be ran at a time. Here are the limitations of scripts:

- Can't accept user input; a user can't provide a value or coordinate point during execution.
- Can't display a dialog box; all input must be provided at the Command prompt. As an alternative to a command that displays a dialog box, check to see if there is a hyphenated version of the command or a system variable to achieve the same task.
- Only one script file can be running at a time except in AutoCAD 2016 and later when using the SCRIPTCALL command.

Even with the previously mentioned limitations, scripts can be very helpful in automating everyday tasks. The following lists some of the advantages and reasons why you might want to use scripts:

- Low learning curve; the creation of scripts builds upon your existing knowledge of commands and their options
- No need for a specialized editor, just need an ASCII text editor such as Notepad or TextEdit
- Typically work with multiple releases and toolsets with no or minimal changes
- Supported transparently; a script can be ran while another command is active
- AutoLISP functions can be executed along with commands defined by other loaded executable files, but the user can't be prompted for input

Prior to creating a script, it is recommended to enter the commands, options, and values that will be executed by the script at the Command prompt. By entering everything at the Command prompt first, you will be able to identify all the input needed for the script and this effort also doubles as the text for the script; saving you from mistyping a command name and leaving out a necessary option or required value.

Here is an example of a command sequence (text in bold reflects the input typed) that sets the limits of a drawing, zooms to the extents of Model space, and displays the grid only within the area of the drawing limits:

```
Command: LIMITS
Reset Model space limits:
Specify lower left corner or [ON/OFF] <0.0000,0.0000>: 0,0
Specify upper right corner <12.0000,9.0000>: 1056,816
Command: ZOOM
```



```
Specify corner of window, enter a scale factor (nX or nXP), or
[All/Center/Dynamic/Extents/Previous/Scale/Window/Object] <real
time>: E
Regenerating model.
Command: GRIDDISPLAY
Enter new value for GRIDDISPLAY <3>: 2
```

The following shows how a script file might look based on the previous command sequence:

```
LIMITS
0,0
1056,816
ZOOM
E
GRIDDISPLAY
2
```

Tip: You can create a script file from the text in the Command Line History window by copying and pasting it into an ASCII text file. Once the text has been pasted, you would remove all the Command prompts and messages displayed to create the script, and then save the file with a .scr extension.

Commands, options, and values in a script can be placed on separate lines as shown in the previous example or on the same line. Placing commands and values on the same line can make it easier to figure out which commands the script is executing, but harder to interpret multiple presses of the Enter key:

```
LIMITS 0,0 1056,816
ZOOM E
GRIDDISPLAY 2
```

The text in a script uses many of the same conventions as those used when entering commands and values at the Command prompt with a few exceptions. The following conventions affect how the text in a script file is interpreted when the script is ran:

- A space is typically equivalent to pressing Enter, just like pressing the Spacebar. In some instances, such as with the WSCURRENT system variable, a space is interpreted as a literal space. Often when a space is required in a value, it usually needs to surround by double quotation marks.
- An end of a line marker is equivalent to pressing Enter.
- A period in front of a command name ensures the use of a command's built-in definition and not a redefined version of a command, if one exists.
- An underscore in front of a command or option name forces the use of its global name, making it easier to run the same script in different languages of the product.
- A semi-colon placed at the start of a line indicates a comment; all text on that line is ignored and not executed.
- A script file must always end with a blank line to ensure the last line is evaluated.

In addition to understanding the text conventions required to properly format a script, you should understand how the AutoCAD environment is affected when a script is running. Just before a script file is ran, the values of the FILEDIA and CMDDIA system variables are set to 0 and restored to their original values upon the completion of the script.

The following provides an explanation of the FILEDIA and CMDDIA system variables:

- **FILEDIA** – Controls the display of file navigation dialog boxes, such as those used when opening or saving a drawing file
- **CMDDIA** – Controls the display of dialog boxes for certain but not all commands, such as QLEADER and PLOT

Before going through the sequence of commands at the Command prompt for a script, you might want to set the FILEDIA and CMDDIA system variables to 0 so the commands behave as they will when ran in a script.

Once a script file has been defined, it can be ran using one of the following methods:

- SCRIPT command
- Drag and drop a SCR file onto a drawing window (Windows only)
- /b (Windows) or -b (Mac OS) command line switch used with a desktop shortcut/alias
`<path\acad.exe> /b "startup.scr"`
- ScriptPro which is a stand-alone application (Windows only) -
<http://knowledge.autodesk.com/support/autocad/downloads/caas/downloads/content/autodesk-customization-conversion-tools.html>

Tip: ScriptPro can be used to execute a script on hundreds of drawings, making it possible to extract block attribute values and design information, and updating CAD standards such as layers and styles.

While a script is running, a script can be paused by pressing the Backspace key. After a script has been paused, it can be resumed with the RESUME command.

The following commands are related to running a script in the AutoCAD product:

Command Name	Description
DELAY	Pauses a script for a specified duration of time measured in milliseconds. The command accepts an integer value up to 32,767 which is equal to just over 32.7 seconds.
RESUME	Resumes the script that was recently paused by a press of the Backspace key.
RSCRIPT	Repeats the most recently ran script in the current AutoCAD session.
SCRIPT	Displays the Select Script File dialog box and allows you to run a script.
SCRIPTCALL	Runs a script from another script; available in AutoCAD 2016 and later.

Additional Information

You can find additional information on scripts and SCR files with these topics in the AutoCAD Online Help system:

- [About Scripts](#)
- [About Command Scripts](#)
- [About Running Scripts at Startup](#)

E2 Create and Run a Script

This exercise explains how to create and run a script file. The script file inserts a title block and configures some basic drafting settings.

See the Exercises section in the handout for this hands-on lab.

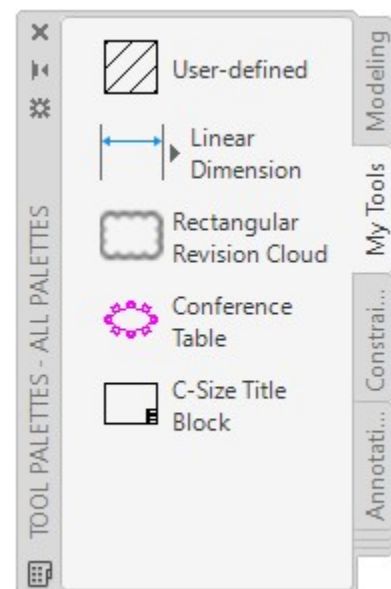
6 Tool Palettes

Tool palettes allow for visual access to commonly used commands, drafting and annotation objects, reference files, hatch pattern and gradient fills among other types of tools. First introduced with AutoCAD 2004, tool palettes have been expanded to support new object types and named styles in later releases.

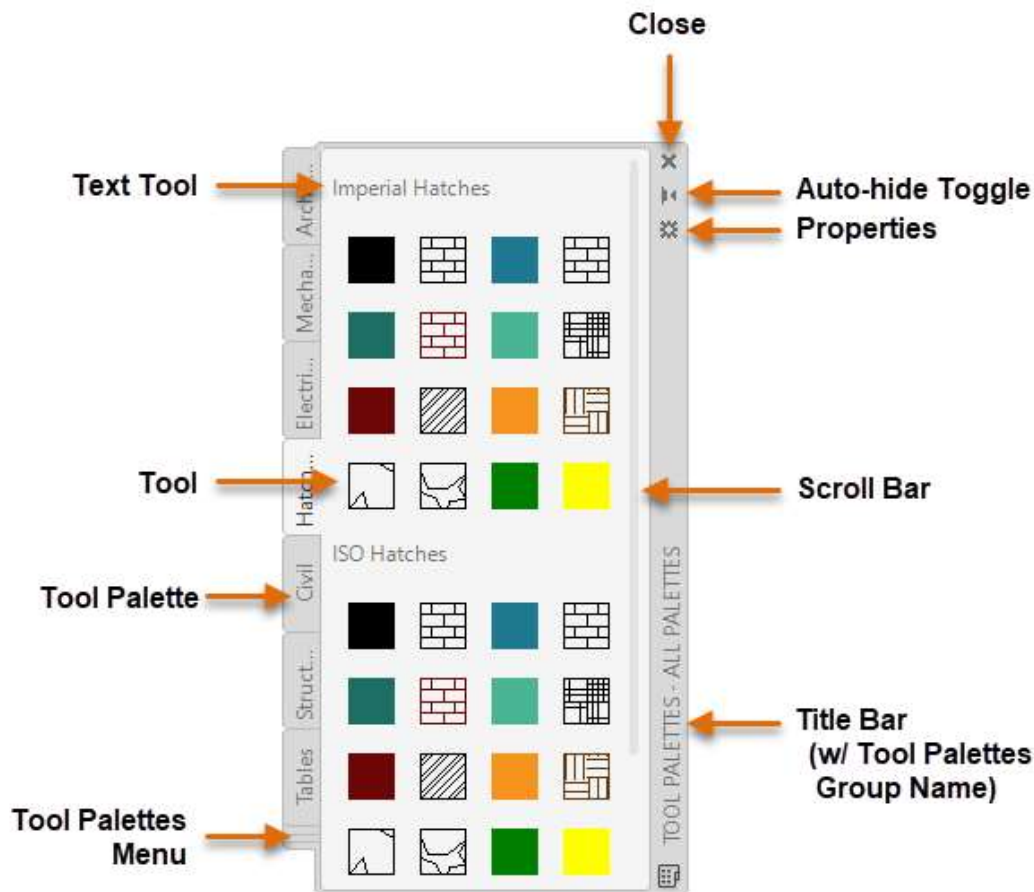
Tool palettes are organized and displayed on the Tool Palettes window. The Tool Palettes window can be displayed using one of the following methods:

- On the ribbon, click View tab > Palettes panel > Tool Palettes
- At the Command prompt, enter TOOLPALETTES or TP
- Press Ctrl+3

The basic framework of the Tool Palettes window is like other palettes in the AutoCAD user interface, but it also contains some unique controls which are related to the organization of tools.



The following illustration identifies the controls and elements of the Tool Palettes window.



1. **Close** – Closes the window.
2. **Scroll Bar** – Indicates additional tools are available; click or drag the scroll bar to move other tools into the current view. Click and hold the mouse button over an empty space within the content area and drag to scroll the listing of tools.
3. **Title Bar** – Displays the window's title along with the current tool palette group's name.
4. **Auto-hide Toggle** – Sets the window's display state for when focus is lost.
5. **Properties** – Displays the window's contextual menu.
6. **Text (Comment) Tool** – Displays information about the tools on the current tool palette.
7. **Tool** – Command or specialty tool that performs an action when clicked or dragged and dropped into the drawing area.
8. **Tool Palette** – Displays the tools associated with the tool palette in the window.
9. **Tool Palettes Menu** – Displays a listing of all tool palettes available in the current tool palette group; choose a tool palette from the list to make it current. A tool palette group can be set current by clicking the Properties control on the window.

Tools and Tool Palettes

A tool is the smallest component of a tool palette that is displayed as part of the Tool Palettes window. Tools can be one of the following:

- Basic geometry (lines, arcs, circles, polylines, ...)
- Annotation (dimensions, text, tables, multileaders, ...)
- External references (xrefs, raster images, underlays, ...)
- Hatch pattern and gradient fills
- Blocks (static and dynamic)
- Visual styles
- Materials
- Lights
- Cameras
- Commands with a command sequence

Create and Modify Tools

New tools can be added to a tool palette using one of these methods:

- Drag an object from the drawing area onto a tool palette; based on the type of object being dropped to a tool palette, the source drawing might need to be saved first
- Drag a command from the Commands List pane in the Customize User Interface (CUI) Editor onto a tool palette
- Drag a supported file type from Windows Explorer or File Explorer (DWG, PNG, JPG, PDF, DGN, ...) onto a tool palette
- Use DesignCenter to add xrefs, blocks, raster images, and hatch patterns

An existing tool on a tool palette can be copied or removed by right-clicking over the tool and choosing Copy or Delete respectfully. If you copy a tool, it can be pasted as a duplicate of the original tool on the same tool palette or it can be pasted on a different tool palette.

Warning: Removing a tool from a tool palette can't be undone.

Once a tool has been created, its properties can be modified using the Tool Properties dialog box. The Tool Properties dialog box is displayed by right-clicking over a tool and choosing Properties. Each tool contains two different types of properties; those that are common to most tools and those that are tool specific. The properties that are common to most tools are known as *general properties*. The general properties control these object characteristics:

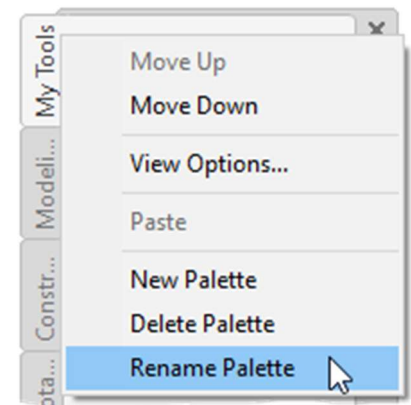
- Color
- Layer name
- Linetype
- Plot Style
- Lineweight
- Transparency

Tool Palettes

Tools are easy to create, and as a result can be hard to find without some planning and organization. Tool palettes are used to not only control which tools are displayed, but they are also used to group and order tools. In addition to using tool palettes to group tools, you can also add elements to a tool palette that allow you to identify and group tools on a tool palette. The elements that can be used to identify and group tools on a tool palette are text and separators.

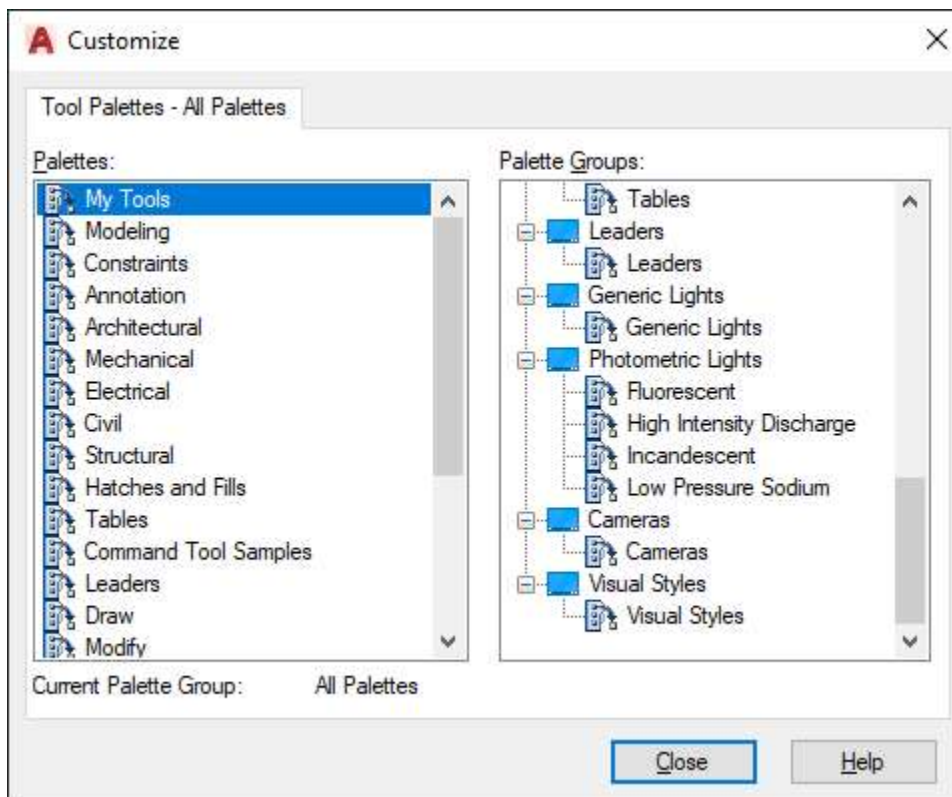
Once a tool palette has been created, you can rename or remove it by right-clicking over its associated tab along the side of the Tool Palettes window and choosing the appropriate action.

Warning: Removing a tool palette cannot be undone.



Tool Palette Groups

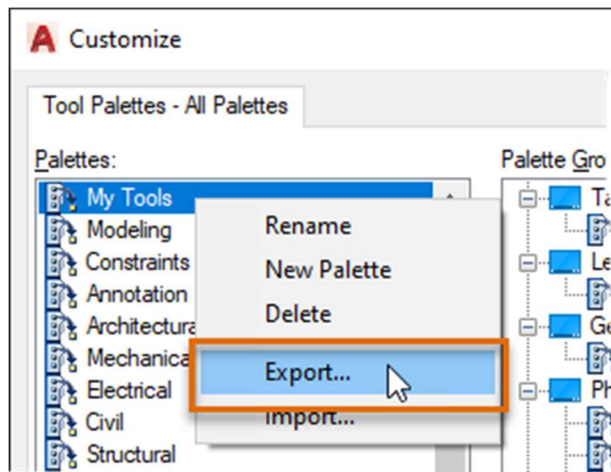
Tool palette groups are used to control which tool palettes are to be displayed on the Tool Palettes window. The All Palettes tool palette group displays all loaded tool palettes on the Tool Palettes window; whereas, for example the Architectural tool palette group shows only a single tool palette with examples of architectural dynamic blocks. Tool palette groups are managed with the Customize dialog box (CUSTOMIZE command).



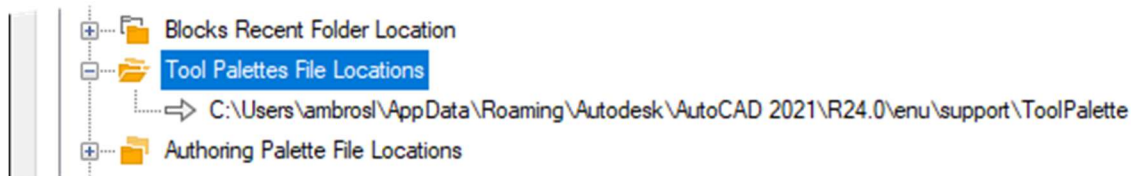
A tool palette group can be set current by right-clicking over the title bar of the Tool Palettes window and choosing the tool palette group name to set current from the bottom of the menu.

Sharing Tool Palettes

Tool Palettes can be shared between multiple users by exporting a tool palette from one workstation and then importing it onto another workstation. The exporting and importing of tool palettes are handled through the Customize dialog box (CUSTOMIZE command).



The location AutoCAD looks for tool palettes is under the Tool Palettes File Locations node on the Files tab of the Options dialog box (OPTIONS command). If your company has a network, you can create and share tool palettes from a network location by adding the path in which the tool palettes are stored to the Tool Palettes File Locations node on each workstation.



Additional Information

You can find additional information on tool palettes with these topics in the AutoCAD Online Help system:

- [About Tool Palettes](#)
- [About Creating Tool Palettes](#)
- [About Changing Tool Palette Settings](#)
- [About Sharing Tool Palettes and Tool Palette Groups](#)

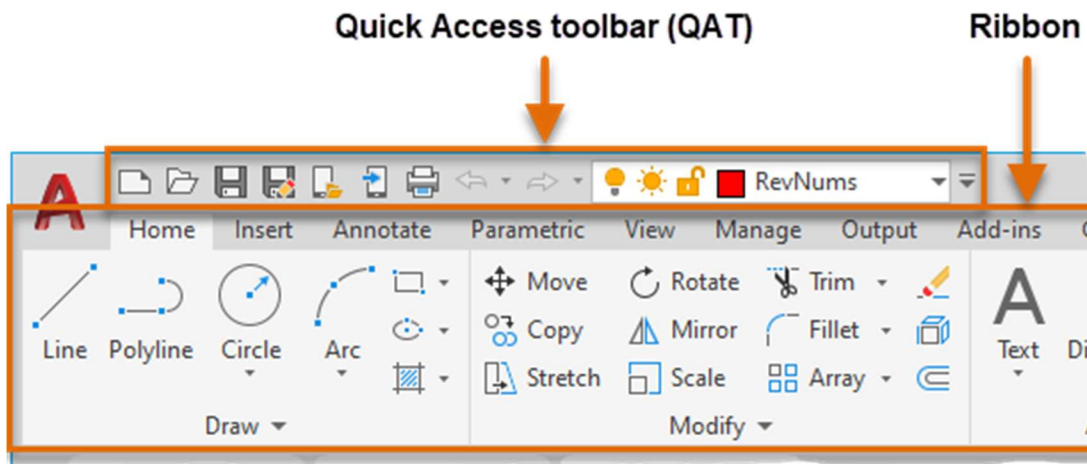
E3 Create a Tool Palette and Tools

This exercise explains how to create a new tool palette with tools based on existing geometry in a drawing, a command tool defined in the *acad.cuix* file along with files stored outside of the AutoCAD program.

See the Exercises section in the handout for this hands-on lab.

7 User Interface (Quick Access toolbar, Ribbon, and Workspaces)

The AutoCAD user interface (UI) contains many features that are used to execute a command or change a current setting and can be customized to better align with your company's workflows. The main features of the AutoCAD UI are the Quick Access toolbar (QAT) and ribbon.



Quick Access Toolbar (QAT)

The QAT is displayed by default in the upper-left corner of the AutoCAD application window above the ribbon. Most of the tools on the QAT are related to file management tasks; creating, opening, saving, and plotting/publishing drawings.

Customizing the QAT can be performed directly from the application window or indirectly with the Customize User Interface (CUI) Editor. From the application window, you can use the Customize button located on the right side of the QAT to:

- Toggle the display of select commands
- Adjust the position of the QAT above or below the ribbon

The Customize User Interface (CUI) Editor can be used to:

- Create a new QAT that can be assigned to different workspaces
- Add and remove commands/controls to/from a QAT
- Adjust the position of the QAT above or below the ribbon

Additional Information

You can find additional information on Quick Access toolbar (QAT) customization with these topics in the AutoCAD Online Help system:

- [About User Interface Customization](#)
- [About Command Customization](#)
- [About Customizing Quick Access Toolbars](#)

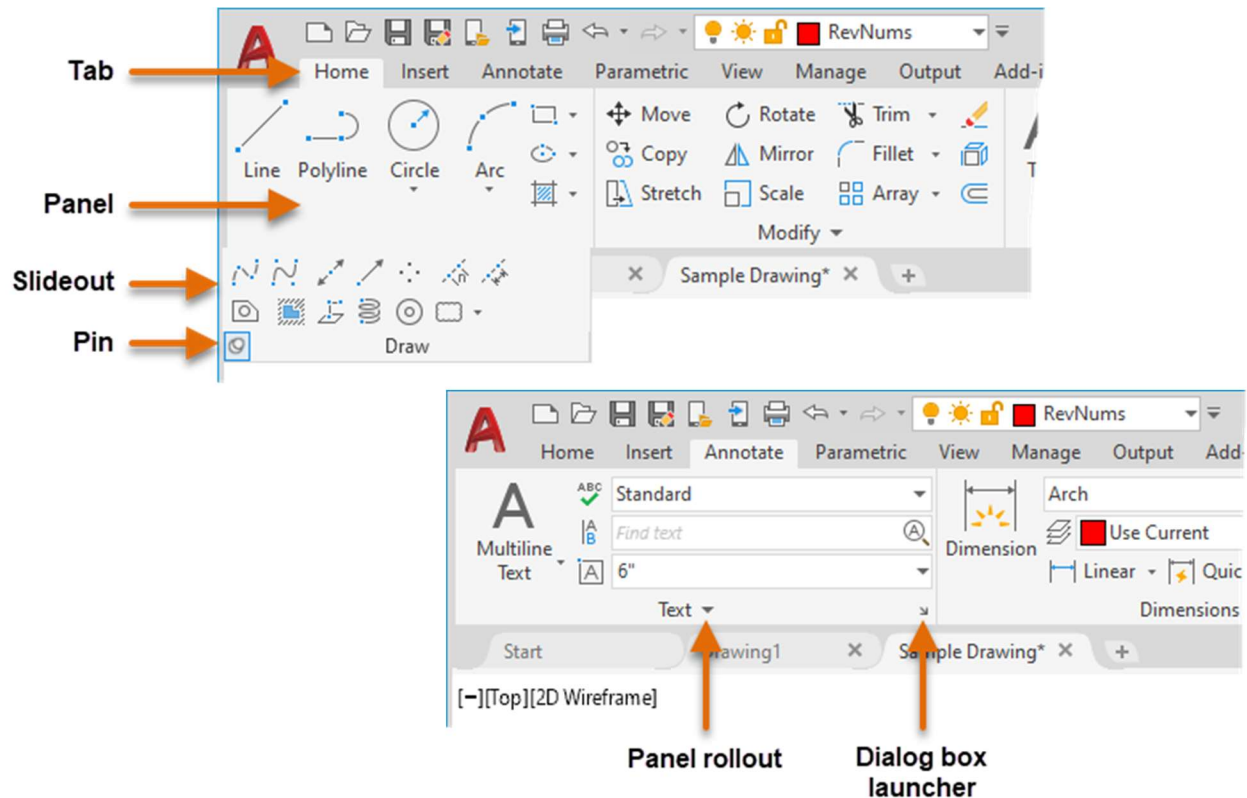
E4 Create a Quick Access Toolbar (QAT)

This exercise explains how to create a new Quick Access toolbar (QAT) and assign it to a workspace.

See the Exercises section in the handout for this hands-on lab.

Ribbon

The ribbon utilizes tabs and panels to group commands and controls by tasks.

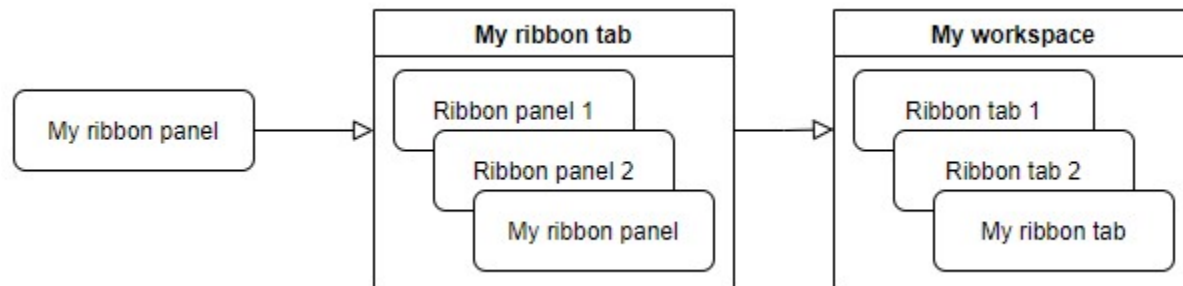


1. **Tab** – Organizes panels by task; draw, annotate, and output among many others.
2. **Panel** – Groups related commands and controls by feature/task; draw, modify, layers, and dimensions among many others.
3. **Control** – Executes a command macro or changes a drawing environment setting.
4. **Panel rollout/Panel title bar** – Displays the name of the panel and indicates whether the panel can be expanded (rolled out) to display additional controls that are below the panel's slideout.
5. **Pin** – Keeps the panel rollout expanded until the panel tab loses focus.
6. **Dialog box launcher** – Starts a command macro, commonly displays a dialog box or palette.

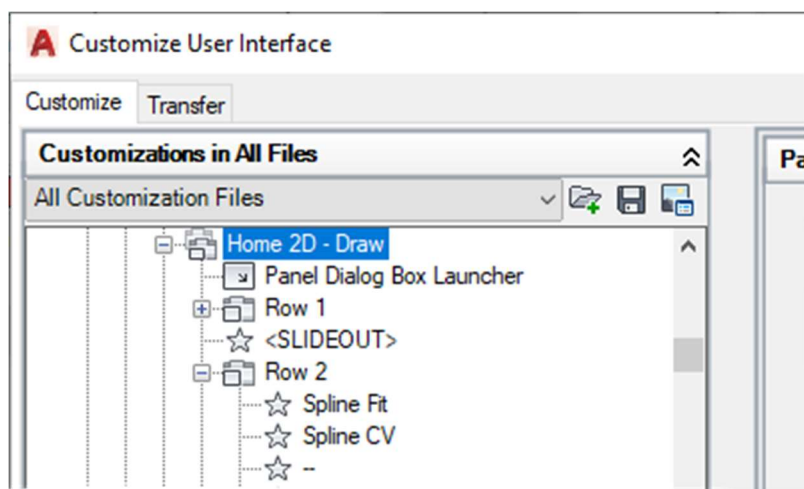
Like the QAT, the ribbon can be customized directly from the application window and the Customize User Interface (CUI) Editor. While limited, the ribbon can be customized from the application window by allowing you to:

- Minimize the display of the ribbon
- Toggle the display of panels and tabs assigned to the current workspace
- Dock and undock panels
- Reorder panels on a tab
- Assign a tool palette group to be displayed when a tab is set current

While some customization of the ribbon can be performed directly in the application window, most customization of the ribbon is handled indirectly with the CUI Editor. The CUI Editor allows you to create ribbon panels and assign those ribbon panels to a ribbon tab which is then displayed using a workspace.



The creation and customization of a ribbon panel is very similar to that of a QAT. However, a ribbon panel can display multiple rows of commands and controls along with it having the ability to define which commands and controls on the panel are displayed by default. The placement of commands and controls in relationship to the <SLIDEOUT> item determine which commands and controls are displayed by default from those that are displayed only when the panel is expanded after the panel's title bar has been clicked.



Additional Information

You can find additional information on ribbon customization with these topics in the AutoCAD Online Help system:

- [About User Interface Customization](#)
- [About Command Customization](#)
- [About Customizing Ribbon Panels](#)
- [About Customizing Ribbon Tabs](#)
- [About Customizing Ribbon Contextual Tab States](#)

E5 Create a Ribbon Panel and Tab

This exercise explains how to create and add commands to a new ribbon panel, create a new ribbon tab to display the new ribbon panel, and control the display of the new ribbon tab with a workspace.

See the Exercises section in the handout for this hands-on lab.

Workspaces

Workspaces play an important role in the AutoCAD UI as they control the display and placement of many user interface elements within the application and drawing windows. AutoCAD comes with several pre-defined workspaces that help you switch between the tools used for 2D drafting and 3D modeling. The Drafting and Annotation workspace is the default workspace that is used when AutoCAD is started the first time.

The user interface elements controlled by a workspace are:

- Quick Access toolbar (QAT)
- “Classic” toolbars
- Pull-down menus
- Palettes
- Ribbon tabs

Workspaces can also be used to:

- Set the Model tab or last used layout current when a drawing is opened
- Control the display of the
 - menu bar,
 - application status bar,
 - navigation bar,
 - drawing window model/layout tabs,
 - and scroll bars in the drawing window.

The display and placement of supported user interface elements can be modified directly in the application window or indirectly with the Customize User Interface (CUI) Editor. After manipulating user interface elements in the application window, changes can be saved to a new or existing workspace with the Save Workspace dialog box (WSSAVE command).



Additional Information

You can find additional information on workspace customization with these topics in the AutoCAD Online Help system:

- [About User Interface Customization](#)
- [About Workspace Customization](#)
- [About Using Task-Based Workspaces](#)

E6 Modify a Workspace

This exercise explains how to modify a workspace directly in the user interface and save the changes to the current workspace.

See the Exercises section in the handout for this hands-on lab.

E7 Reset the AutoCAD Environment

This exercise explains how to reset AutoCAD back to its default settings.

See the Exercises section in the handout for this hands-on lab.

8 Desktop Shortcuts

Shortcuts, often found on the Windows desktop/taskbar or Start Menu, are used to open documents and start applications. Application shortcuts are commonly used to start an executable and pass executable optional parameters, known as *command line switches*, to alter the way the executable runs. For example, when you install an AutoCAD-based toolset, the installer typically creates two shortcuts; one that can be used to start plain AutoCAD and a second that starts the toolset product. These two shortcuts use different command line switches to start the AutoCAD executable in the preferred configuration.

Customizing a shortcut involves changing its Target property and by adding one or more of the 15+ different available command line switches associated with the AutoCAD executable; AutoCAD LT supports less than 15 different command line switches. Command line switches are not case sensitive, and more than one command line switch can be used at a time.

The following is the default value of the Target property for the AutoCAD 2021 shortcut:

```
"C:\Program Files\Autodesk\AutoCAD 2021\acad.exe" /product ACAD
/language "en-US"
```

The following table lists several command line switches available for the AutoCAD executable.

Command Line Switch	Description
/product	Specifies the AutoCAD-based product to launch when AutoCAD is installed side-by-side with an AutoCAD-based toolset. /product ACAD /product ACA
/language	Specifies which installed language pack to use. /language "en-us"
/nologo	Hides the splash screen during startup. /nologo
/p	Sets a named user profile current that is defined in the product or loads a previously exported profile (ARG) file. /p "<<Unnamed Profile>>"
/t	Specifies the drawing template to use for the default drawing. /t "mytemplate.dwt"
/w	Sets a named workspace current if defined in the main CUIx file. /w "2D Drafting"
/safemode	Prevents the loading and execution of all executable files in the current session. Introduced with AutoCAD 2014. /safemode

Note: When specifying a folder or file that contains spaces in the Target property, you must surround the path and file name with double quotation marks. Without the double quotation marks, the AutoCAD program interprets each space as the start of a new parameter. For example, *C:\Program Files\Autodesk\AutoCAD 2021\acad.exe* would need to be entered as *"C:\Program Files\Autodesk\AutoCAD 2021\acad.exe"*.

Multiple command line switches can be used together, simply add a space before each command line switch and its associated value; some switches do not require an associated value.

The following example shows the use of multiple command line switches:

```
"C:\Program Files\Autodesk\AutoCAD 2021\acad.exe" /product ACAD
/language "en-US" /nologo /t "C:\Datasets\AS468502-L\C-size.dwt"
/w "3D Basics"
```

- **/product** controls whether the AutoCAD program or an AutoCAD toolset is started when one or more AutoCAD-based products of the same release are installed on the same workstation; in the previous example, the AutoCAD program is started
- **/language** specifies which language pack to use when one or more language packs are installed; in the previous example, the English language pack is used
- **/nologo** suppress the program's splash screen during startup
- **/t** specifies the drawing template to use for the creation of the default drawing; in the previous example, the drawing template used is named *C-size.dwt*
- **/w** specifies the workspace to set current; in the previous example, the "3D Basic" workspace is set current

Note: In the previous example, replace "C:\Datasets\AS468502-L\C-size.dwt" with the location of the DWT file after extracting the dataset for the hands-on lab.

Additional Information

You can find additional information on command line switches with these topics in the AutoCAD Online Help system:

- [About Customizing Startup](#)
- [Command Line Switch Reference](#)

E8 Create a Desktop Shortcut

This exercise explains how to create a custom desktop shortcut to launch AutoCAD, create the default drawing based on a specific drawing template file, and set the "3D Basic" workspace current.

See the Exercises section in the handout for this hands-on lab.

9 Exercises

See the *ClassHandout-AS468502-L-Ambrosius-AU2020.pdf* file for the exercises associated with this session.

10 Where to Get More Information

When you first start to learn a new skill, there is no doubt that you will have questions and where you go to get answers for those questions might not be clear. The following list of resources can be helpful in locating answers to the questions you might have along with where to go to further learn about AutoCAD customization:

- **AutoCAD Online Help System** – The Customization Guide in the AutoCAD Online Help system contains a lot of information on customizing many of the options mentioned in this session.

To access the online help, go to: <https://help.autodesk.com/view/OARX/2021/ENU/>

- **Autodesk Discussion Forums** – The Autodesk forums provide peer-to-peer networking and interactions with Autodesk moderators. Here, you can ask a question about anything AutoCAD related and get a response from a fellow user or Autodesk employee.

To access the Autodesk discussion forums, go to <https://forums.autodesk.com>, click Browse By Product near the upper-right of the page and then click AutoCAD. Click the appropriate subgroup link.

- **AUGI Forums** – The AUGI forums provide peer-to-peer networking where you can ask questions about virtually anything in AutoCAD or Autodesk software related and get a response from a fellow user. Visit AUGI at <https://www.augi.com>.
- **Industry Events and Classes** – Industry events such as Midwest University and Autodesk University are great places to learn about features and workflows in an Autodesk product. Along with industry events, you might also be able to find classes at your local technical college or an Autodesk Authorized Training Center (ATC).
- **Internet** – There are tutorials and information scattered across the Internet that you can read to learn more about customization and programming options supported in the AutoCAD program. Use your favorite search engine, such as Google or Bing, to find this information.
- **Books** – There are many general and specialized books that cover AutoCAD customization and programming. To find a book, use Amazon ([amazon.com](https://www.amazon.com)) or Barnes & Noble ([bn.com](https://www.bn.com)) to locate a book online or visit your local Barnes and Noble store. My latest book, *AutoCAD Platform Customization: User Interface, AutoLISP, VBA, and Beyond*, covers all the customization options mentioned in this session and many more.

