

AS500221

Down the Rabbit Hole: AutoCAD Customization and Task Automation

Lee Ambrosius
Autodesk, Inc.

Learning Objectives

- Learn the importance of customization and task automation
- Discover which customization and automation features are available
- Learn about which additional tools might be needed
- Learn how to load and use sample programs and files

Description

Have you ever found yourself wondering if there's a more efficient way to work? Maybe you downloaded an AutoLISP or VBA program to improve your productivity, only to discover that it doesn't work quite like you'd hoped. Many of the daily interactions you perform in the AutoCAD program can be customized or automated. Using customization, you can tailor the product to the way you work. And automation enables you to streamline design workflows. This session will provide you with a basic, high-level introduction to features and programming languages you can use to customize and automate workflows in AutoCAD, along with resources that can help get you started.

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

Have you ever found yourself sitting at your desk working in AutoCAD and wondering if there was an easier or faster way to get your work done? Or maybe you found yourself feeling like you are on autopilot constantly repeating commands.



But what do you do, it's just the way the software works, right? Well, no, it doesn't have to be that way. AutoCAD supports many customization and automation features that can help you improve your productivity and allow you to focus on your design rather than on the design tool.



Wait a minute... Time out... Rewind. Customization and automation features? Nope, that sounds like programming and I'm not a programmer.

Before you bow out and exit stage right... what if I told you, you are already a programmer and you didn't even know it. Confused? I'm sure you are, about as much as I was typing this sentence.

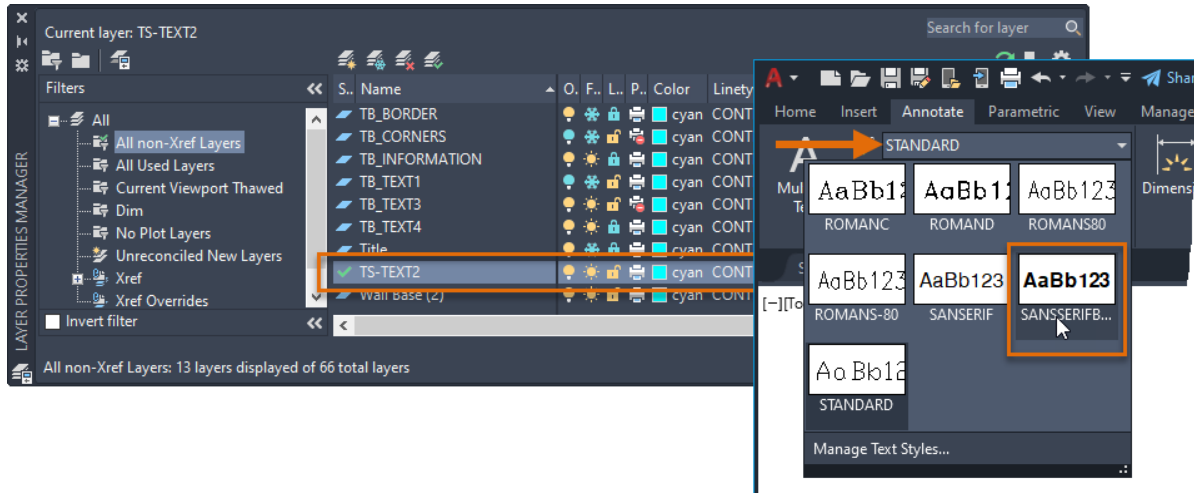
Okay... you might not be a programmer in the traditional sense writing code and compiling programs, but you know how to instruct AutoCAD to do something by performing commands in a specific order to create or modify objects. This is done using the commands and system variables you normally enter at the Command prompt or start from the ribbon (or the AutoCAD user interface).

Leveraging your understanding of commands and system variables, you can create sequences that combine multiple commands and options along with the input you normally provide. This saves you time by performing the same sequence consistently each time.



AUTODESK UNIVERSITY

For example, you might commonly use multiple layers and text styles in a drawing for labels and general notes. Instead of setting a layer and then text style current followed by starting the TEXT or MTEXT command with a default text height, you could combine all those commands into a single sequence. That not only saves you time, but also helps to enforce established CAD standards.



```
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/Xref]: s
Enter layer name to make current or <select object>: ts-text2
Enter an option [?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRANsparency/MATerial/Plot/Freeze/Thaw/LOck/UnloCk/
stAte/Description/rEconcile/Xref]:
Command: TEXTSTYLE
Enter new value for TEXTSTYLE <"STANDARD">: SANSSERIFBOLD
Command: MTEXT
Current text style: "SANSSERIFBOLD" Text height: 61/256" Annotative: No
Specify first corner:
Specify opposite corner or [Height/Justify/Line spacing/Rotation/Style/Width/Columns]: h
Specify height <61/256">: 3/4"
Specify opposite corner or [Height/Justify/Line spacing/Rotation/Style/Width/Columns]:
```

The customization and programming features of AutoCAD help 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 session is a high-level introduction to the customization and automation features available in AutoCAD. Giving you a glimpse behind the curtain and explains how your knowledge of commands and system variables can be used to improve your productivity in new and exciting ways.

2 Customization and Programming Options Available

Not all customization and programming options are created equally. Some options are easy to learn and so well-integrated into AutoCAD that you might not even realize you are customizing the program. The creation of new layers, named text and dimension styles, and blocks are all forms of customization that you might not think twice about.

AutoCAD supports two types of customization and programming:

- Drawing
- Application

The following lists many of the customization and programming options AutoCAD supports:

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 configurations and 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)

- | | |
|---|--|
| <ul style="list-style-type: none">• AutoLISP / Visual LISP• Visual Basic for Applications (VBA)• ActiveX / COM (VBA, VBScript, VB.NET, C#, C++)• Managed .NET (VB.NET, C#)• ObjectARX (C++) | <ul style="list-style-type: none">• JavaScript• Sheet Set Manager API• CAD Standards plug-ins• Transmittal API• Database Connectivity Automation API• Forge Platform APIs |
|---|--|

AUTODESK UNIVERSITY

Where to Begin?

If you feel overwhelmed by all the customization and programming options available, don't worry as I will point out in this section where to focus your efforts at first and then where you might want to go from there.

Start with drawing templates, if you are not already using them, this will make creating new drawings that contain your company's CAD standards much easier. Drawing templates can include commonly used layers and annotation styles along with your title block inserted on a layout tab.

Next, you will want to make sure you configure any plot devices and plot styles you might need to output your drawings as a hardcopy or an electronic file. This will help to ensure your drawing files are correctly plotted or published.

Once that is all done, then you will want to look at these customization and programming options:

- Command aliases
- User interface (CUI Editor)
- Scripts
- Action macros
- Tool palettes
- Workspaces
- User profiles

For information on the previously mentioned customization and programming options, see the next few numbered sections along with the links to learning resources and past AU sessions under Section 6 later in this handout.

Further Down the Road

After you take advantage of some of the basic customization options, you may want to look at AutoLISP programming. AutoLISP programming allows you to leverage your knowledge of commands and system variables and build from there with a basic understanding of the AutoLISP programming language syntax and just five AutoLISP functions.

In addition to learning AutoLISP, if you want to connect to external data sources or programs in Microsoft Office, the VBA programming language can be beneficial. If the AutoLISP and VBA programming languages don't offer enough capabilities in the programs you need, you will want to consider the AutoCAD Managed .NET and then ObjectARX APIs.

3 Customization Examples from Command Aliases to AutoLISP

As I mentioned earlier, you have the knowledge and tools needed to get started with customizing AutoCAD and increase your productivity.

To get started with customizing AutoCAD, you need:

- AutoCAD installed on your workstation
- Basic text editor available; Notepad on Windows or TextEdit on Mac OS
- Understanding of commands and system variables
- Understanding of command sequences

In the next sections, you will learn about command aliases and how they can be used to quickly start commands along with command sequences. Command sequences allow you to take the commands and system variables you use to define:

- Scripts
- Command macros
- Action macros
- AutoLISP programs

Command Aliases

Command aliases are one of the simplest forms of AutoCAD customization. A command alias is a shortened name assigned to a command. You use a command alias by entering it at the Command prompt as an alternative to the full name of a command or clicking the button on the ribbon that is associated with the command. If you commonly start commands from the ribbon, using command aliases can save you time and improve your proficiency as you don't need to frequently move the cursor out of and then back into the drawing area.

Here are a few examples of command aliases in AutoCAD:

```
C, *CIRCLE
CO, *COPY
E, *ERASE
L, *LINE
T, *MTEXT
```

The command alias is before the comma and the associated AutoCAD command is after the asterisks (*).

Command aliases are stored in the *acad.pgp* (or *acadlt.pgp* on AutoCAD LT) file. The *acad.pgp* file is loaded automatically each time AutoCAD starts. For information on how to add or modify command aliases, see [AutoCAD Customization Boot Camp: Basic \(No Experience Required\)](#) from Autodesk University 2020 on the AU website.

A Look at Command Sequences

Almost every action that you perform in AutoCAD starts a command, whether the command is shown in the history of the Command window or not. Commands that are executed together and repeatedly in the same order are known as a *command sequence*. Command sequences make the best candidates to improve your productivity by reducing the number of times you might need to access commands and tools from the AutoCAD user interface.

AUTODESK UNIVERSITY

For example, the following command sequences might represent the order of commands you use to add text to a drawing to ensure it conforms with your CAD standards:

- LAYER, MTEXT
- LAYER, STYLE, MTEXT
- STYLE, LAYER, MTEXT

Commands like LAYER and STYLE normally display a user interface, such as a palette or dialog box, that is user friendly but not customization friendly. In many cases, commands that utilize a user interface for input have an alternative command that starts with a hyphen (-).

For example, instead of the LAYER command, you would use the -LAYER command to automate workflows related to managing layers. The -LAYER command displays the following prompt:

```
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Pl
ot/Freeze/Thaw/LOck/Unlock/stAte/Description/rEconcile/Xref]:
```

Using the -LAYER command, you can perform many of the same tasks that you would normally do from the Layer Properties Manager or Layers drop-down list on the ribbon.

At the Command prompt, the sequence to set a layer named TS-TEXT2 current looks like:

```
Command: -LAYER
Current layer: "0"
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Pl
ot/Freeze/Thaw/LOck/Unlock/stAte/Description/rEconcile/Xref]: S
Enter layer name to make current or <select object>: TS-TEXT2
Enter an option
[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRansparency/MATerial/Pl
ot/Freeze/Thaw/LOck/Unlock/stAte/Description/rEconcile/Xref]:
```

Similarly, instead of using the STYLE command, you use the -STYLE command to manage text styles when automating workflows. At the Command prompt, the sequence to set a text style named SANSSERIFBOLD current looks like:

```
Command: -STYLE
Enter name of text style or [?] <STANDARD>: SANSSERIFBOLD
Existing style. Full font name = SansSerif Bold
Specify font name or font filename (for SHX) <SansSerif>:
Specify height of text or [Annotative] <0'-0">:
Specify width factor <1.00000000>:
Specify obliquing angle <0>:
Display text backwards? [Yes/No] <No>:
Display text upside-down? [Yes/No] <No>:
"SANSSERIFBOLD" is now the current text style.
```

The creation of a multiline (MText) object using automation might look like the following:

```
Command: MTEXT
Current text style: "SANSERIFBOLD" Text height: 61/256" Annotative: No
Specify first corner:
Specify opposite corner or [Height/Justify/Line
spacing/Rotation/Style/Width/Columns]: H
Specify height <61/256">: 0.75
Specify opposite corner or [Height/Justify/Line
spacing/Rotation/Style/Width/Columns]:
```

Along with utilizing commands that start with a hyphen (-), you can also utilize system variables in customizing AutoCAD and automating workflows. Instead of using the -LAYER command to set a layer current, you could use the CLAYER system variable which is short for "Current Layer". A text style can be set current using the TEXTSTYLE system variable.

The usage of these two system variables looks like:

```
Command: CLAYER
Enter new value for CLAYER <"0">: TS-TEXT2
Command: TEXTSTYLE
Enter new value for TEXTSTYLE <"STANDARD">: SANSERIFBOLD
```

Note: The TEXTLAYER system variable can also be used to set the layer in which new text objects are to be placed.

You can get a list of available commands and system variables using the following links:

- [Commands List](#)
- [Hyphenated Commands List](#)
- [System Variables List](#)

Command Sequence to a Script

Scripts execute a sequence of commands, options, and values in a linear order without the ability to request input from the user. They are great for automating tasks like general drawing setup, creating layers and styles in a drawing, or migrating from one set of CAD standards to another.

Here is an example of commands and input entered at the Command prompt that might be used to setup a drawing:

```
Command: LIMITS
Reset Model space limits:
Specify lower left corner or [ON/OFF] <0'-0",0'-0">: 0,0
Specify upper right corner <1'-0",0'-9">: 408,264
Command: -LAYER
Current layer: "0"
```


AUTODESK UNIVERSITY

Enter an option

[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRANsparency/MATERial/Pl
ot/Freeze/Thaw/LOck/Unlock/stAte/Description/rEconcile/Xref]: **M**

Enter name for new layer (becomes the current layer) <0>: **TS-TBLK**

Enter an option

[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRANsparency/MATERial/Pl
ot/Freeze/Thaw/LOck/Unlock/stAte/Description/rEconcile/Xref]: **C**

New color [Truecolor/COLORbook] : **7**

Enter name list of layer(s) for color 7 (white) <TS-VPORT>:

Enter an option

[?/Make/Set/New/Rename/ON/OFF/Color/Ltype/LWeight/TRANsparency/MATERial/Pl
ot/Freeze/Thaw/LOck/Unlock/stAte/Description/rEconcile/Xref]:

Command: **-INSERT**

Enter block name or [?]: **B-TBLK**

Units: Unitless Conversion: 0'-1"

Specify insertion point or [Basepoint/Scale/X/Y/Z/Rotate/Explode/REpeat]:
0,0

Enter X scale factor, specify opposite corner, or [Corner/XYZ] <1>: **24**

Enter Y scale factor <use X scale factor>: **24**

Specify rotation angle <0>: **0**

Command: **ZOOM**

Specify corner of window, enter a scale factor (nX or nXP), or
[All/Center/Dynamic/Extents/Previous/Scale/Window/Object] <real time>:
EXTENTS

Command: **ORTHOMODE**

Enter new value for ORTHOMODE <0>: **1**

Command: **GRIDMODE**

Enter new value for GRIDMODE <1>: **0**

Command: **SNAPMODE**

Enter new value for SNAPMODE <0>: **0**

Command: **OSMODE**

Enter new value for OSMODE <47>: **33**

Command: **WSCURRENT**

Enter new value for WSCURRENT <"Drafting & Annotation">: **Drafting &
Annotation**

AUTODESK UNIVERSITY

When you create a script, the same exact commands, options, and values you enter at the Command prompt are entered into the script file. For example, the previous input, might look like one of these two script files (example on the left has commands and values on different lines; whereas the example on the right has the command and associated values on the same line):

<pre>LIMITS 0,0 408,264 -LAYER M TS-TBLK C 7 -INSERT B-TBLK 0,0 24 24 0 ZOOM EXTENTS ORTHOMODE 1 GRIDMODE 0 SNAPMODE 0 OSMODE 33 WSCURRENT Drafting & Annotation</pre>	<pre>LIMITS 0,0 408,264 -LAYER M TS-TBLK C 7 -INSERT B-TBLK 0,0 24 24 0 ZOOM EXTENTS ORTHOMODE 1 GRIDMODE 0 SNAPMODE 0 OSMODE 33 WSCURRENT Drafting & Annotation</pre>
---	--

When a script file is run in AutoCAD, each space or end of line is interpreted as the pressing of the Spacebar or Enter key. When a space is interpreted as a literal space, such as the string value of the MTEXT command or workspace name, you will want that input to be the last on a line.

Note: A script file must always have a blank line after the last line in the file.

Scripts are stored in plain text files with the SCR extension. For information on how to create a script file, see [AutoCAD Customization Boot Camp: Basic \(No Experience Required\)](#) from Autodesk University 2020 on the AU website.

Command Sequence to a Command Macro

Command macros are executed from the AutoCAD user interface (UI) with the use of buttons/controls on the ribbon and toolbars, and menu items on shortcut menus. A command macro can be used to start a single command and specify no options or values or can be much more complex in that it can start many commands and provide specific options and values. Unlike scripts, command macros can request input from the user.

In the *A Look at Command Sequence* section earlier in this handout, I showed the prompts of the command sequence -LAYER, -STYLE, and MTEXT. Using that same sequence of commands and input provided, here is what the command macro might look like:

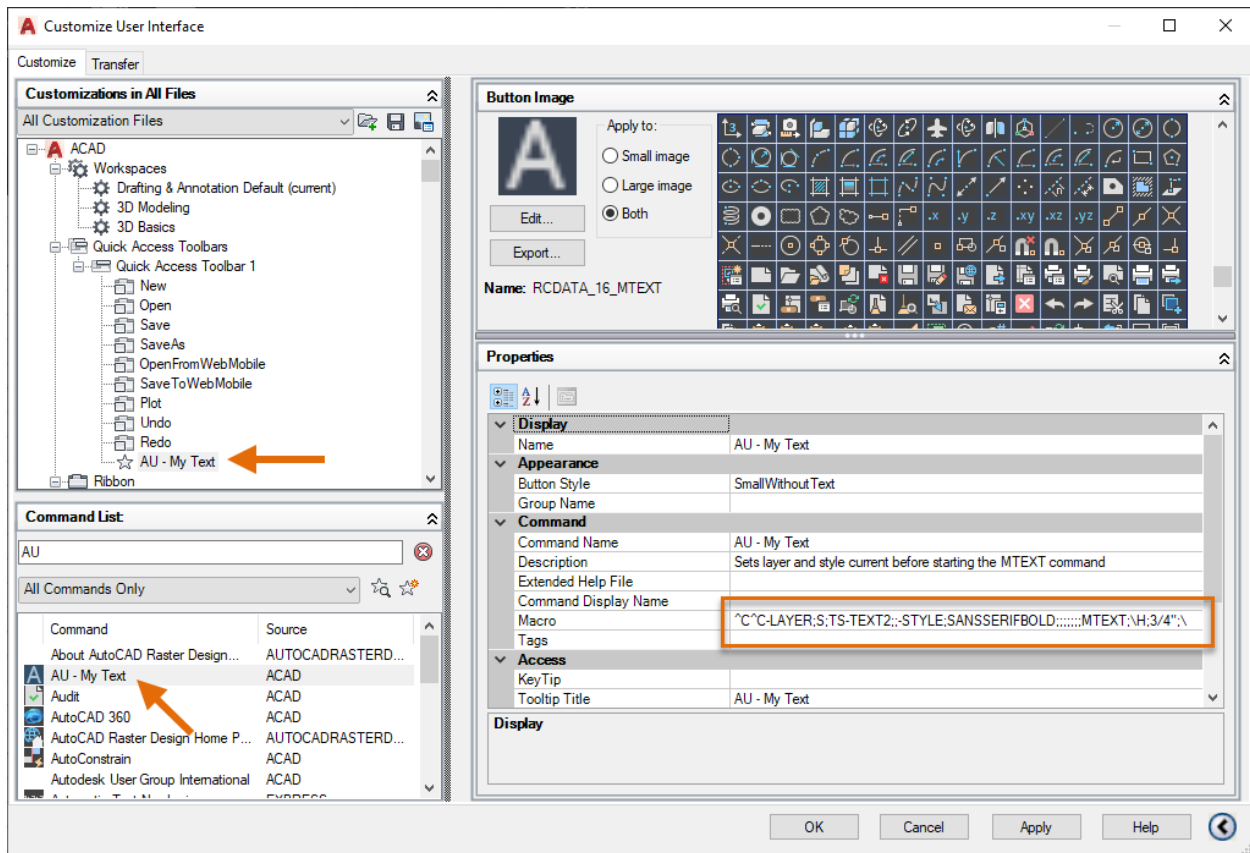
```
^C^C-LAYER;M;TS-TEXT2;C;5;;-STYLE;SANSERIFBOLD;;;;;;MTEXT;\H;0.75;\
```

Now that you have seen a command macro, let's take a closer look at the various parts and syntax of the example command macro:

- A single ^C sequence represents the pressing of the Esc key once. Before the macro starts, you would typically want to press Esc twice to ensure any active command is cancelled.
Note: The cancelling of a command doesn't deselect any previously selected commands.
- Command names, options, and values are indicated by the orange color. These are just as they appeared when entered at the Command prompt.
- A semi-colon represents the pressing of the Enter key. Just like after you type the name of a command, option, or value you must press Enter, the same is true in a command macro.
- The use of the backslash (\) character represents a request (or pause) for user input. Typically, a request for input is used to allow the user to select objects or specify a point in a drawing. A backslash (\) character doesn't always need to be preceded with a semi-colon, though you may need to provide one when prompting the user to select objects.

AUTODESK UNIVERSITY

The following image shows a custom command with the previous described macro in the Customize User Interface (CUI) Editor. The command in the example is added to AutoCAD's Quick Access toolbar (QAT).



The Customize User Interface (CUI) Editor is used to define custom commands and add those commands to the AutoCAD User Interface (UI), such as the ribbon and Quick Access toolbar (QAT). The CUI Editor is displayed with the CUI command. You use the CUI Editor to manage customization (CUIx) files. CUIx files can be loaded into AutoCAD using the CUI Editor or CUILOAD command.

In addition to defining the tools on the ribbon and toolbars along with menu items on the shortcut menu, customization (CUIx) files also contain workspaces. A workspace is used to organize the elements of the AutoCAD user interface (UI).

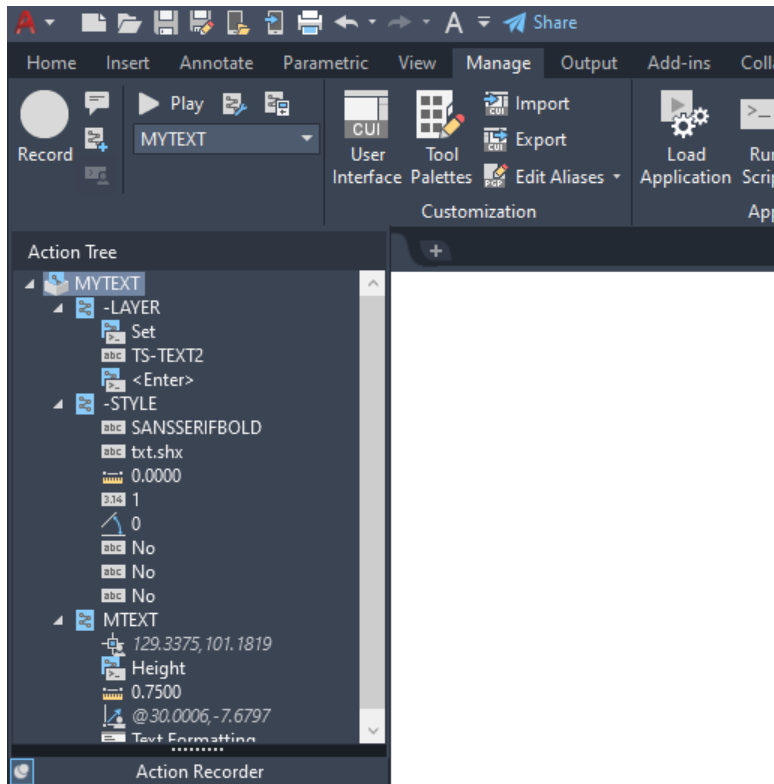
For information on how to create command macros, and customize the ribbon and QAT in AutoCAD along with workspaces, see [AutoCAD Customization Boot Camp: Basic \(No Experience Required\)](#) from Autodesk University 2020 on the AU website.

Command Sequence to an Action Macro

Action macros are a mix of functionality provided by scripts, command macros, and AutoLISP programs, of which I discuss in the next section. Instead of needing to understand specialized syntax, you simply perform the tasks you want to repeat and save them to an action macro (ACTM) file.

Action macros are recorded using the Action Recorder located on the Manage tab of the ribbon in AutoCAD. You start recording by clicking the Record button and then end recording by clicking the Stop button. When done recording and naming an action macro, you can edit recorded values or indicate which values you would like the user to specify during playback.

Note: You can't add commands to an action macro after recording has stopped, but you can remove commands you recorded and no longer want as part of your action macro.



An action macro can be played back from the ribbon or by entering its name at the Command prompt.

Action macros are stored in ACTM files that are automatically loaded each time AutoCAD starts. For information on how to record and modify an action macro, see [AutoCAD Customization Boot Camp: Automate Workflows and Tasks](#) from Autodesk University 2020 on the AU website.

Command Sequence to an AutoLISP program

AutoLISP allows you to execute commands and provide options and values to those commands. Users can also be requested to provide input when a command is being executed or when you need a value from the user. AutoLISP statements are typically stored in a file with the LSP extension, but they can also be part of a command macro or script file and even entered directly at the Command prompt.

While there are hundreds of functions that make up the AutoLISP programming language, there are five functions you need to understand to get started. The five functions you need to know to take advantage of AutoLISP are:

- COMMAND – Executes a command with specified options and values.
- SETQ – Stores a value which can later be passed to a command or AutoLISP function.
- SETVAR – Assigns a value to a system variable.
- GETVAR – Gets the current value of a system variable.
- DEFUN – Organizes multiple AutoLISP statements into a named function. Functions can be defined as commands which can be entered directly at the AutoCAD Command prompt.

In the *A Look at Command Sequence* section earlier in this handout, I showed the prompts of the -LAYER, -STYLE, and MTEXT commands. Here is what that same sequence would look as AutoLISP statements:

```
(command "-LAYER" "M" "TS-TEXT2" "C" "5" "" "")  
(command "-STYLE" "SANSSERIFBOLD" "" "" "" "" "" "")  
(command "MTEXT" PAUSE "H" 0.75 PAUSE)
```

Now that you have seen some AutoLISP statements, let's take a closer look at the various parts and syntax of the example AutoLISP statements:

- An AutoLISP statement start with an open and end with a closing parenthesis.
- Command names, options, and values are indicated by the orange color. These are just as they appeared when entered at the Command prompt except, they are surrounded by double quotation marks in most situations. Double quotation marks are used to denote a string value.
- Double quotation marks without a value represent the pressing of the Enter key.
- The PAUSE variable (or token) is used to indicate when the user should be prompted for user input.
- If a string contains a double quotation mark, put a backslash (\) character before the double quotation mark to interpret it as a literal double quotation mark. For example, if your string contains an inch mark it might look like "1 1/4\".

Note: The previous AutoLISP statements expect the TS-TEXT2 layer and SANSSERIFBOLD text style to already exist in the drawing. Using AutoLISP, you can check to see if a layer or text style already exists in a drawing and make the appropriate changes to the AutoCAD environment to allow the AutoLISP program to execute as expected.

In the *Command Sequence to a Script* section earlier in this handout, I showed the prompts of several commands and system variables that might be used to setup a drawing. Here is what that same sequence from the script file example would look as AutoLISP statements:

```
(command "LIMITS" "0,0" "408,264")
(command "-LAYER" "M" "TS-TBLK" "C" "7" "" "")
(command "-INSERT" "B-TBLK" "0,0" 24 24 0)
(command "ZOOM" "EXTENTS")
(setvar "ORTHOMODE" 1)
(setvar "GRIDMODE" 0)
(setvar "SNAPMODE" 0)
(setvar "OSMODE" 33)
(setvar "WSCURRENT" "Drafting & Annotation")
```

Note: Along with string values, AutoLISP also supports integer and real numbers along with lists that might represent a point (X,Y,Z) value. Numeric values don't need to be surrounded with double quotation marks.

AutoLISP statements can be combined into a named function that can be used from the Command prompt. Here is an example of one of the previous AutoLISP examples defined as a function with the name of MYTEXT_AUTOLISP and can be entered at the Command prompt:

```
(defun c:MYTEXT_AUTOLISP (/)
  (command "-LAYER" "M" "TS-TEXT2" "C" "5" "" "")
  (command "-STYLE" "SANSERIFBOLD" "" "" "" "" "" "")
  (initdia)
  (command "MTEXT" PAUSE "H" 0.75 PAUSE)
)
```

When the example above is loaded in AutoCAD, you can enter MYTEXT_AUTOLISP at the Command prompt to execute the AutoLISP statements defined within the function.

Note: The (initdia) statement before the MTEXT command ensures it displays the MTEXT editor instead of prompting for text at the Command prompt.

AutoLISP programs are stored in plain text files with the LSP extension. For information on how to create AutoLISP programs, see [AutoCAD Customization Boot Camp: Automate Workflows and Tasks](#) from Autodesk University 2020 on the AU website.

4 What Tools Do You Need

Many of the customization and programming options or features available for use within the AutoCAD program are supported through

- Utilities or commands found inside the program
- 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 install additional software which may have an additional cost. If you want to develop web-based applications that utilize the Forge APIs, you will need to know a programming language that can utilize REST APIs and obtain a developer account and pay for the utilized services.

The following table lists the utilities needed to customize AutoCAD and automate workflows:

Drawing Customization/Programming Options

Option/Feature	Utility/Program (Command)
Layer	<ul style="list-style-type: none">• Layer Properties Manager (LAYER)
Annotation styles (text, dimensions, multileaders, and tables)	<ul style="list-style-type: none">• Text Style Manager (STYLE)• Dimension Style Manager (DIMSTYLE)• Multileader Style Manager (MLEADERSTYLE)• Table Style Manager (TABLESTYLE)• Multiline Style Manager (MLSTYLE)
Blocks	<ul style="list-style-type: none">• Create Block (BLOCK)
Dynamic Blocks	<ul style="list-style-type: none">• Block Editor (BEDIT)
Materials and render presets	<ul style="list-style-type: none">• Materials Browser (MATBROWSEROPEN)• Materials Editor (MATEDITOROPEN)• Render Presets Manager (RENDERPRESETS)
Visual styles	<ul style="list-style-type: none">• Visual Styles Manager (VISUALSTYLES)
Drawing templates	<ul style="list-style-type: none">• Save Drawing As (SAVEAS)

Application Customization/Programming Options

Option/Feature	Utility/Program (Command)
Desktop shortcut	<ul style="list-style-type: none">• Windows Desktop
Command aliases	<ul style="list-style-type: none">• Notepad in Windows
Scripts	<ul style="list-style-type: none">• Notepad in Windows
Action macros	<ul style="list-style-type: none">• Action Recorder (ACTRECORD)
User interface (CUI Editor)	<ul style="list-style-type: none">• Customize User Interface Editor (CUI)

Option/Feature	Utility/Program (Command)
Custom linetypes and hatch patterns	<ul style="list-style-type: none"> • Notepad in Windows
Custom shapes and text styles	<ul style="list-style-type: none"> • Notepad in Windows
Tool palettes	<ul style="list-style-type: none"> • Tool Palettes palette (TOOLPALETTES) • Customize dialog box (CUSTOMIZE)
Workspaces	<ul style="list-style-type: none"> • Customize User Interface Editor (CUI) • Save Workspace dialog box (WSSAVE)
User profiles	<ul style="list-style-type: none"> • Options dialog box (OPTIONS)
Plot configurations and styles	<ul style="list-style-type: none"> • Plotter Manager (PLOTTERMANAGER) • Plot Style Manager (STYLESMANAGER)
AutoLISP / Visual LISP	<ul style="list-style-type: none"> • Visual LISP Editor (VLIDE) • Microsoft Visual Studio Code
Visual Basic for Applications (VBA)	<ul style="list-style-type: none"> • Visual Basic for Applications Integrated Development Environment (VBAIDE) • Visual Basic for Applications Manager (VBAMAN)
ActiveX / COM	<ul style="list-style-type: none"> • AutoLISP, VBA, VBScript, VB.NET, C#, or C++ programming languages
Sheet Set Manager API	<ul style="list-style-type: none"> • VBA, VB.NET, C#, or C++ programming languages
CAD Standards plug-ins	<ul style="list-style-type: none"> • VB.NET, C#, or C++ programming languages
Transmittal API	<ul style="list-style-type: none"> • VBA, VBScript, VB.NET, C#, or C++ programming languages
Database Connectivity Automation API	<ul style="list-style-type: none"> • VBA, VB.NET, C#, or C++ programming languages
JavaScript	<ul style="list-style-type: none"> • Notepad in Windows • Atom • Visual Studio Code
Managed .NET (VB.NET, C#)	<ul style="list-style-type: none"> • Visual Studio • Visual Studio Community
ObjectARX (C++)	<ul style="list-style-type: none"> • Visual Studio • Visual Studio Community
Forge Platform APIs	<ul style="list-style-type: none"> • Visual Studio Code • Visual Studio • Visual Studio Community

5 Loading and Using the Sample Applications

If you have made it this far in the handout, I congratulate you on wanting to dig deeper into customizing AutoCAD and automating your workflows. This section focuses on the effort of loading customized files and programs into AutoCAD using the sample files that are part of this session in AutoCAD.

Adding Support Paths for Blocks, Custom Programs, and Action Macros

This exercise explains how to download and organize the sample files for this session, and then add that folder in which the sample files are stored to the Support File Search Path and other settings in the Options dialog box of the AutoCAD program.

These steps explain how to download and extract the contents of this session's dataset:

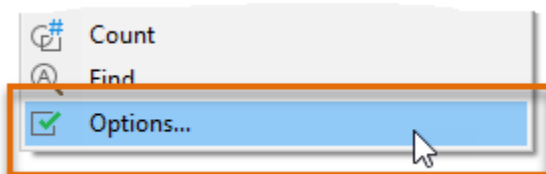
1. Browse to this session's page on the Autodesk University website.
Go to <https://www.autodesk.com/autodesk-university/conference/overview> and search on the session ID AS500221.
2. In the search results, click the entry for this session.
The dataset should start downloading to your local drive.
3. On the session's page, click Downloads and then click Materials.
4. Once the materials have been downloaded, extract the contents of the ZIP file to:

C:\Datasets\AS500221

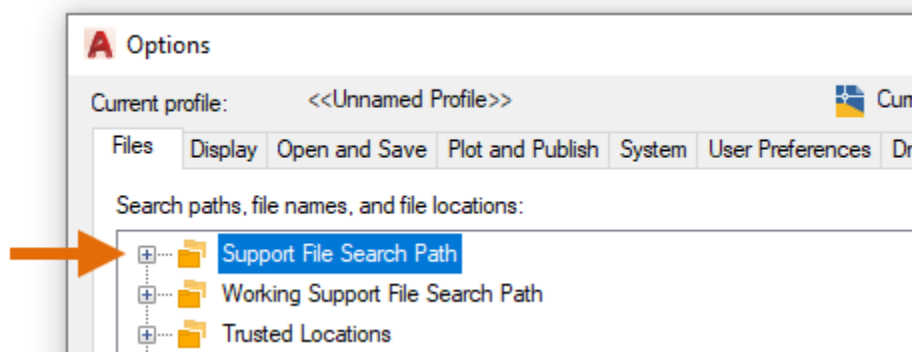
Note: You can extract the files to a different folder on your workstation, if desired. If you do extract the files to a different folder, make sure to use the new folder in place of the folder mentioned above.

These steps explain how to add the folder with this session's sample files to the Support File Search Path and other settings in the Options dialog box of the AutoCAD program:

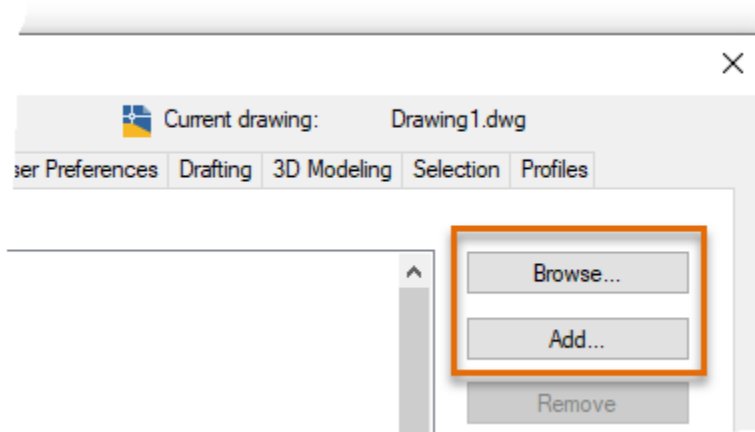
1. Launch AutoCAD and open a drawing.
2. In the drawing area, right-click and choose Options.



3. In the Options dialog box, Files tab, select the Support File Search Path node.

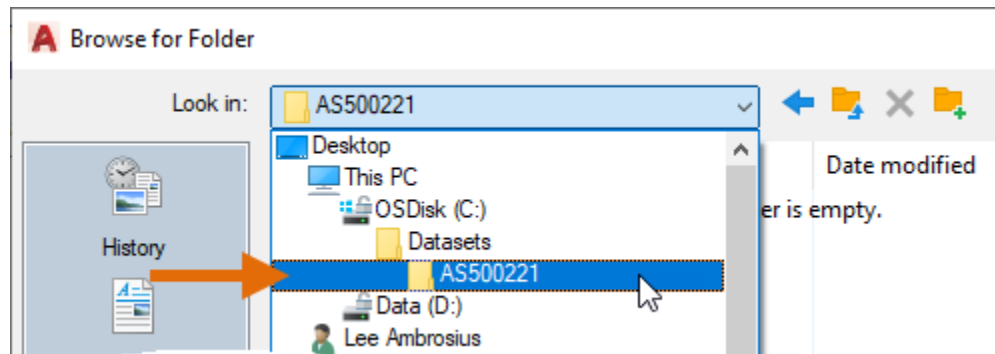


4. On the right side, click Add and then click Browse.

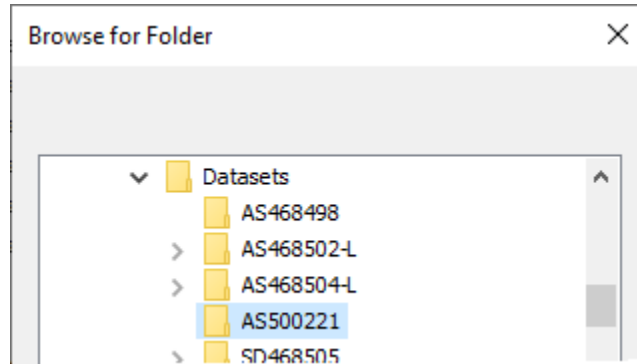


5. Do one of the following:

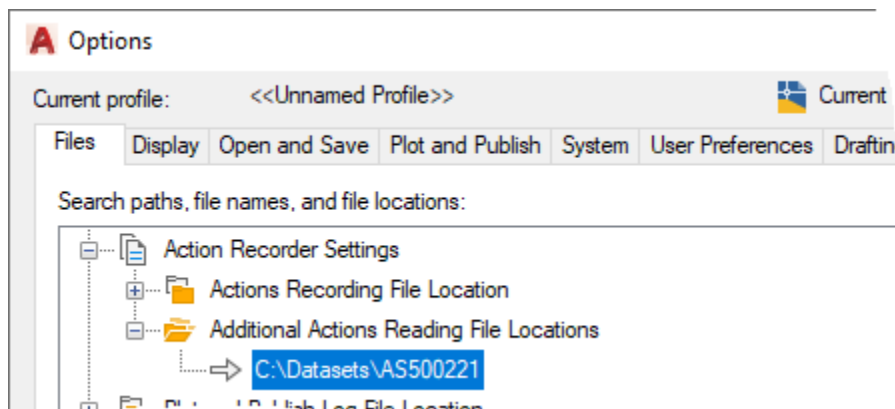
- **AutoCAD 2021 or later:** In the Browse for Folder dialog box, browse to and select the C:\Datasets\AS500221 folder or the folder in which you extracted this session's sample files. Click Open.



- **AutoCAD 2020 and earlier:** In the Browse for Folder dialog box, browse to and select the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click OK.



6. On the Files tab, expand the Action Recorder Settings node and then select the Additional Actions Reading File Locations node.



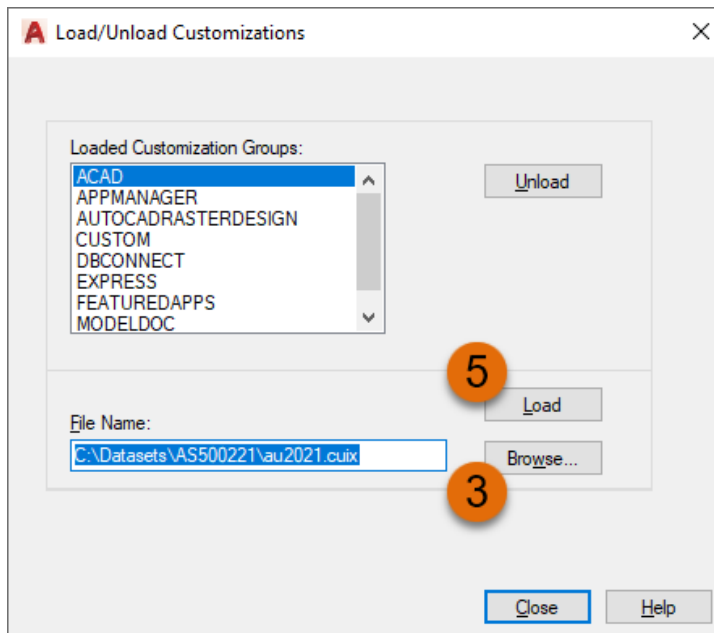
7. On the right side of the dialog box, click Add and then click Browse. Browse to *C:\Datasets\AS500221* or the folder in which you extracted this session's sample files. Click Open (or OK).
8. Click OK to save the setting changes and exit the Options dialog box.
9. Close and discards the changes to the drawing, and then restart AutoCAD.

Load a Partial Customization (CUIx) File

The *au2021.cuix* customization (CUIx) file that comes with the sample files for this session adds a new ribbon tab named "AU – My Tools" to the AutoCAD user interface. This ribbon tab contains several tools that are used to load the sample customization and program files that are included with this session's sample files using custom command macros. The CUIx file also contains the example command macro shown earlier in this handout under the *Command Sequence to a Command Macro* section.

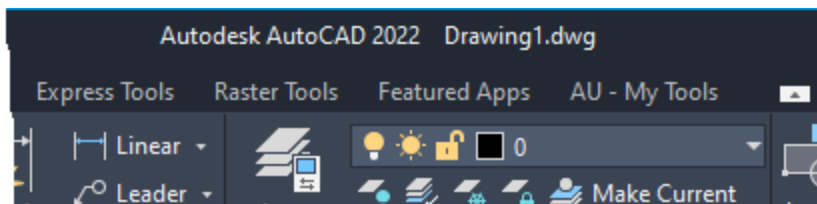
1. Launch AutoCAD and open a drawing.
2. At the Command prompt, enter **cuiload**.

3. In the Load/Unload Customizations dialog box, click Browse.

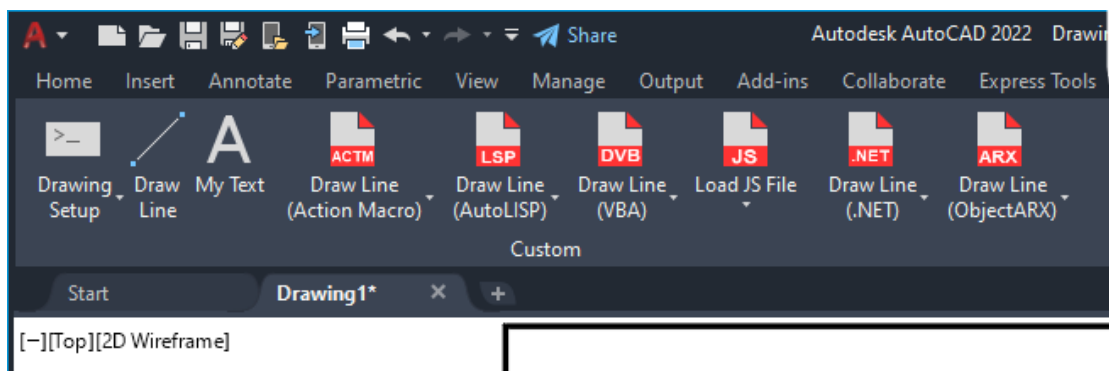


4. In the Select Customization File dialog box, browse to and select the *au2021.cuix* file located in the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click Open.
5. In the Load/Unload Customizations dialog box, click Load and then click Close.

Once the CUIx file has been loaded, you should notice the AU – My Tools tab added to the ribbon.



6. On the ribbon, click the new AU – My Tools tab.



The buttons on the AU – My Tools ribbon tab load and run the various sample files in the following sections.

Note: Drawing Setup must be used before any of the other commands, as it sets up the layers, text styles, and other settings used by the other commands.

- **Drawing Setup drop-down menu** – Allows you to run or open the *startup.scr* file. See *Run a Script (SCR) File* for more information on manually running a script file.
- **Draw Line** – Command macro that draws a line from 0,0 to 150,150 on the current layer. See *Command Sequence to a Command Macro* for more information on command macros.
- **My Text** – Command macro that sets a layer and text style current, and then starts the MText command. See *Command Sequence to a Command Macro* for more information on command macros.
- **Action Macro drop-down menu** – Allows you to playback the DL_ACTMACRO or MYTEXT_ACTMACRO action macro. See *Playback an Action Macro (ACTM) File* for more information on playing back an action macro.
- **AutoLISP drop-down menu** – Loads the *samples.lsp* file and then starts either the DL_AUTOLISP or MYTEXT_AUTOLISP command. See *Load an AutoLISP Source (LSP) File* for more information loading and running custom AutoLISP programs.
- **VBA drop-down menu** – Loads the *samples.dvb* file and then runs either the DrawLine or MyText macro. See *Load a VBA Project (DVB) File* for more information on loading and running VBA macros.
- **JS drop-down menu** – Allows you to load the *samples-<releases>.js* file and then start either the DL_JAVASCRIPT or MYTEXT_JAVASCRIPT command. See *Load a JavaScript (JS) or HTML File* for more information on loading and running custom commands defined with JavaScript.
- **.NET drop-down menu** – Loads the *samples-<releases>.dll* file and then starts either the DL_VBNET or MYTEXT_VBNET command. See *Load a Managed .NET (DLL) File* for more information on loading and running custom commands defined with .NET.
- **ObjectARX drop-down menu** – Loads the *samples-<releases>.arx* file and then starts either the DL_OBJECTARX or MYTEXT_OBJECTARX command. See *Load an ObjectARX (ARX) File* for more information on loading and running commands defined with ObjectARX.

Run a Script (SCR) File

Script (SCR) files are plain text files that execute a sequence of commands. You can run a SCR file using one of the following methods:

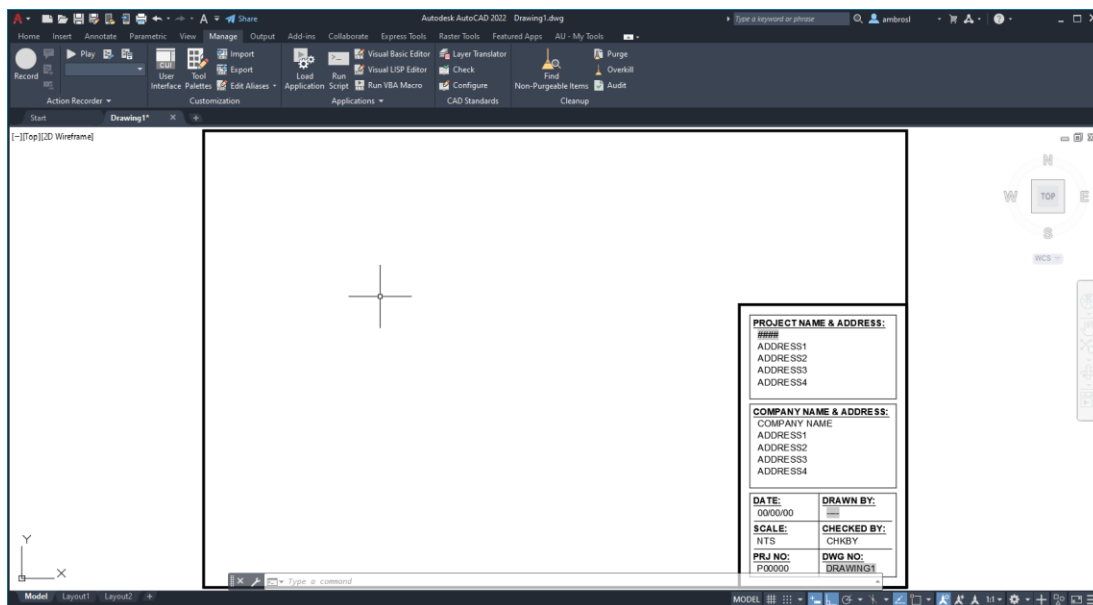
- SCRIPT command from the AutoCAD Command prompt, a command macro in the AutoCAD user interface or even an AutoLISP program
- SCRIPTCALL command from another SCR file
- Drag 'n drop a SCR file into the drawing window in which you want to run the script file
- As part of a custom AutoCAD desktop shortcut using the /b command line switch

Note: An example of running a script file from a command macro can be found in the *au2021.cuix* file that is part of this session's sample files. How to load this CUIx file is explained in the *Load a Partial Customization (CUIx) File* section.

The following steps explain how to manually run the *startup.scr* file that is part of this session's sample files. The *startup.scr* file is nearly identical to the script shown earlier under the *Command Sequence to a Script* section.

1. Launch AutoCAD and create a new drawing.
2. On the ribbon, click Manage tab > Applications panel > Run Script (or enter **script** at the Command prompt).
3. In the Select Script File dialog box, browse to and select the *startup.scr* file located in the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click Open.

A title block should have been inserted on the TS-TBLK layer along with several drafting aids being changed and the Drafting & Annotation workspace being set current.



Playback an Action Macro (ACTM) File

Action macro (ACTM) files are automatically loaded into AutoCAD if they are in one of the folders identified under the Action Recorder Settings node in the Options dialog box (OPTIONS command). You can run a loaded action macro using one of these methods:

- Play the action macro back from the Action Record panel on the Manage tab or from the Action Record > Play submenu when you right-click in the drawing window
- Enter the name of the action macro to playback at the Command prompt

The following steps explain how to playback the sample action macros DL_ACTMACRO and MYTEXT_ACTMACRO. The action macros should already be loaded into AutoCAD after completing the steps under the *Adding Support Paths for Blocks, Custom Programs, and Action Macros* section earlier in this handout.

1. Launch AutoCAD and create a new drawing.
2. Run the *startup.scr* file following the steps mentioned in the *Run a Scrip (SCR) File* section.
3. At the Command prompt, enter **dl_actmacro**.

A line is drawn from 0,0 to 150,150 on the current layer.

Note: If the **dl_actmacro** command is unknown, make sure you setup the action macro related path correctly under the *Adding Support Paths for Blocks, Custom Programs, and Action Macros* section.

4. If the Action Macro – Playback Complete message box is displayed, click Close.
5. At the Command prompt, enter **mytext_actmacro**.
6. Specify two points to define the location and width of the MText object, and then enter some text into the Multiline Text editor. When done, click outside of the editor to create the MText object.

The text is placed on the TS-TEXT2 layer and assigned the SANSSERIFBOLD style.

7. If the Action Macro – Playback Complete message box is displayed, click Close.

Load an AutoLISP Source (LSP) File

AutoLISP programs are commonly stored in LSP files. A LSP file can be loaded into AutoCAD by using the:

- Load option and Startup Suite of the Load/Unload Applications dialog box (APPLOAD command)
- **LOAD** AutoLISP function with the *acad.lsp* and *acaddoc.lsp* files, and command macros in the AutoCAD user interface
- Plug-in bundles

Note: Examples of loading an LSP file and starting the commands defined in the file from a command macro can be found in the *au2021.cuix* file that is part of this session's sample files. How to load this CUIx file is explained in the *Load a Partial Customization (CUIx) File* section.

The following steps explain how to load the *samples.lsp* file that is part of this session's sample files with the Load/Unload Applications dialog box. After loading the LSP file, you will execute the MYTEXT sample that was shown earlier in the *Command Sequence to an AutoLISP program* section.

1. Launch AutoCAD and create a new drawing.
2. Run the *startup.scr* file following the steps mentioned in the *Run a Scrip (SCR) File* section.
3. On the ribbon, click Manage tab > Applications panel > Load Application (or enter **appload** at the Command prompt).
4. In the Load/Unload Applications dialog box, browse to and select the *samples.lsp* file located in the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click Load.
5. If the Security – Unsigned Executable File message box is displayed, click Load Once.
6. In the Load/Unload Applications dialog box, click Close to close the dialog box.
7. Once LSP file has been loaded, at the Command prompt, enter **dl_autolisp**.
A line is drawn from 0,0 to 150,150 on the current layer.
8. At the Command prompt, enter **mytext_autolisp**.
9. Specify two points to define the location and width of the MText object, and then enter some text into the Multiline Text editor. When done, click outside of the editor to create the MText object.

The text is placed on the TS-TEXT2 layer and assigned the SANSERIFBOLD style.

Note: There are two additional commands defined in the *samples.lsp* file. These commands are named DL_AutoLISPCOM and DL_AutoLISPENTMAKE. The commands show alternative ways to create a line object that doesn't rely on the use of the COMMAND function in AutoLISP.

Load a VBA Project (DVB) File

VBA macros are stored in project files with the DVB extension. A DVB file can be loaded with:

- VBALOAD command
- Visual Basic for Applications Manager (VBAMAN command)
- *acad.dvb* file which gets loaded during the startup of AutoCAD
- VL-VBALOAD AutoLISP function

Important: Before you can utilize VBA in AutoCAD, you must have previously downloaded and installed Visual Basic for Applications for AutoCAD. The installer can be found at:
<https://www.autodesk.com/vba-download>

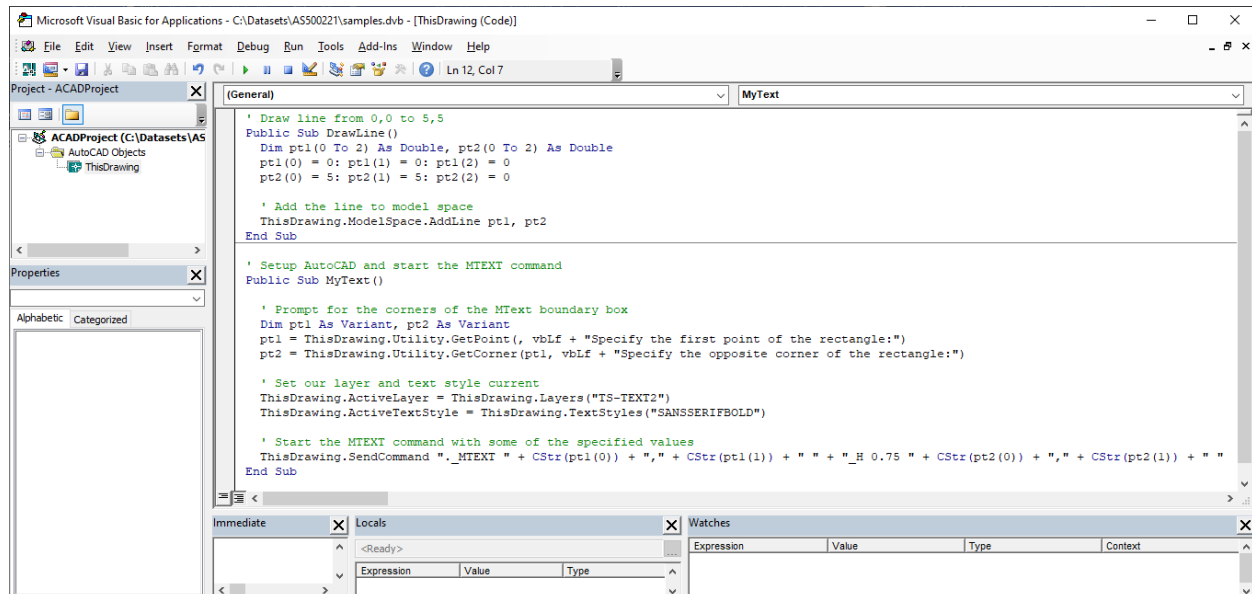
Note: Examples of loading a VBA project file and running macros defined in the project from a command macro can be found in the *au2021.cuix* file that is part of this session's sample files. How to load this CUIx file is explained in the *Load a Partial Customization (CUIx) File* section.

The following steps explain how to manually load the *samples.dvb* file that is part of this session's sample files with the Visual Basic for Applications Manager (VBALOAD command). After loading the DVB file, you will execute the DrawLine and MyText macros.

1. Launch AutoCAD and create a new drawing.
2. Run the *startup.scr* file following the steps mentioned in the *Run a Scrip (SCR) File* section.
3. On the ribbon, click Manage tab > expand the Applications panel > VBA Manager (or at the Command prompt, enter **vbaman**).
4. In the VBA Manager, click Load.
5. In the Open VBA Project dialog box, browse to and select the *samples.dvb* file located in the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click Open.
6. If the Security – Unsigned Executable File message box is displayed, click Load Once.
7. In the AutoCAD message, click Enable Macros.
8. In the VBA Manager, click Macros.
9. In the Macros dialog box, select the DrawLine macro and click Run.
A line is drawn from 0,0 to 150,150 on the current layer.
10. On the ribbon, click Manage tab > Applications panel > Run VBA Macro (or at the Command prompt, enter **vbarun**).
11. In the Macros dialog box, select the MyText macro and click Run.
12. Specify two points to define the location and width of the MText object, and then enter some text into the Multiline Text editor. When done, click outside of the editor to create the MText object.

The text is placed on the TS-TEXT2 layer and assigned the SANSERIFBOLD style.

Note: You can view the code that makes up the VBA macros by displaying the Visual Basic for Applications Integrated Development Environment (VBAIDE command). Within the VBA IDE, you can define new macros and custom dialog boxes with the user of forms.



Load a JavaScript (JS) or HTML File

JavaScript programs are commonly stored in JS files along with HTML files. A JS file can be loaded into AutoCAD with the `WEBLOAD` command. In addition to the `WEBLOAD` command, the `SHOWHTMLMODALWINDOW` AutoLISP function can also be used to load a HTML file that contains JavaScript into a modal dialog box.

Note: Examples of loading a JS file and starting the commands defined in the file from a command macro can be found in the `au2021.cuix` file that is part of this session's sample files. How to load this CUIx file is explained in the *Load a Partial Customization (CUIx) File* section.

The following steps explain how to load the `samples-2018_2020.js` or `samples-2021_2022.js` file that is part of this session's sample files into AutoCAD with the `WEBLOAD` command. After loading the JS file, you will start the `DL_JAVASCRIPT` or `MYTEXT_JAVASCRIPT` commands which are like the commands defined in the AutoLISP samples shown earlier in the *Command Sequence to an AutoLISP program* section.

1. Launch AutoCAD and create a new drawing.
2. Run the `startup.scr` file following the steps mentioned in the *Run a Scrip (SCR) File* section.
3. At the Command prompt, enter **webload**.
4. At the *Enter an option [Load/Unload/?] <Load>*: prompt, press Enter or enter **L** to use the Load option.
5. At the *Enter javascript URL to load*: prompt, enter one of the following based on the release of AutoCAD you are using:
 - AutoCAD 2021 - 2022 - **C:/Datasets/AS500221/samples-2021_2022.js**
 - AutoCAD 2018 - 2020 - **C:/Datasets/AS500221/samples-2018_2020.js**

Note: If you extracted this session's sample files to a different folder, replace the folder mentioned with the one in which you extracted the files.

6. At the Command prompt, enter **dl_javascript**.
A line is drawn from 0,0 to 150,150 on the current layer.
7. At the Command prompt, enter **mytext_javascript**.
8. Specify two points to define the location and width of the MText object, and then enter some text into the Multiline Text editor. When done, click outside of the editor to create the MText object.

The text is placed on the TS-TEXT2 layer and assigned the SANSSERIFBOLD style.

Load a Managed .NET (DLL) File

Managed .NET programs are compiled into a DLL file. A Managed .NET DLL file can be loaded into AutoCAD using the:

- NETLOAD command at the Command prompt, in an AutoLISP program, or a command macro in the AutoCAD user interface
- Plug-in bundles
- Windows Registry loader

Note: Examples of loading a DLL file and starting the commands defined in the file from a command macro can be found in the *au2021.cuix* file that is part of this session's sample files. How to load this CUIx file is explained in the *Load a Partial Customization (CUIx) File* section.

Important: This sample was built for AutoCAD 2021 and AutoCAD 2022. It might run in an earlier AutoCAD release, but you likely will need to recompile the sample project into a new DLL to get it to run in a release prior to AutoCAD 2021.

The following steps explain how to load the *samples-2021_2022.dll* file that is part of this session's sample files into AutoCAD with the NETLOAD command. After loading the DLL file, you will execute the DL_VBNET and MYTEXT_VBNET commands.

1. Launch AutoCAD and create a new drawing.
2. Run the *startup.scr* file following the steps mentioned in the *Run a Scrip (SCR) File* section.
3. At the Command prompt, enter **netload**.
4. In the Choose .NET Assembly dialog box, browse to and select the *samples-2021_2022.dll* file located in the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click Open.
5. If the Security – Unsigned Executable File message box is displayed, click Load Once.
6. At the Command prompt, enter **dl_vbnet**.
A line is drawn from 0,0 to 150,150 on the current layer.
7. At the Command prompt, enter **mytext_vbnet**.

8. Specify two points to define the location and width of the MText object, and then enter some text into the Multiline Text editor. When done, click outside of the editor to create the MText object.

The text is placed on the TS-TEXT2 layer and assigned the SANSERIFBOLD style.

Note: There is a third command defined upon the loading of the DLL file. This command is named MYTEXT_IPE_VBNET. This command utilizes the AutoCAD API to display the MText in-place editor instead of relying on the MTEXT command.

Load an ObjectARX (ARX) File

ObjectARX programs are compiled into to a DLL file that commonly has the extension of ARX or CRX. An ARX or CRX file can be loaded into AutoCAD by using the:

- Load option and Startup Suite of the Load/Unload Applications dialog box (APPLOAD command)
- ARXLOAD AutoLISP function with the *acad.lsp* and *acaddoc.lsp* files, and command macros in the AutoCAD user interface
- ARX command with the *acad.lsp* and *acaddoc.lsp* files, and command macros in the AutoCAD user interface
- Plug-in bundles
- Windows Registry loader

Note: Examples of loading a DLL file and starting the commands defined in the file from a command macro can be found in the *au2021.cuix* file that is part of this session's sample files. How to load this CUIx file is explained in the *Load a Partial Customization (CUIx) File* section.

Important: This sample was built specifically for AutoCAD 2021 and AutoCAD 2022. It cannot be loaded into an earlier AutoCAD release, so you will need to recompile the sample project into a new ARX to get it to load into a release prior to AutoCAD 2021.

The following steps explain how to load the *samples-2021_2022.arx* file that is part of this session's sample files into AutoCAD with the Load/Unload Applications dialog box. After loading the ARX file, you will start the DL_OBJECTARX and MYTEXT_OBJECTARX commands.

1. Launch AutoCAD and create a new drawing.
2. Run the *startup.scr* file following the steps mentioned in the *Run a Scrip (SCR) File* section.
3. On the ribbon, click Manage tab > Applications panel > Load Application (or enter **appload** at the Command prompt).
4. In the Load/Unload Applications dialog box, browse to and select the *samples-2021_2022.arx* file located in the *C:\Datasets\AS500221* folder or the folder in which you extracted this session's sample files. Click Load.
5. If the Security – Unsigned Executable File message box is displayed, click Load Once.
6. In the Load/Unload Applications dialog box, click Close to close the dialog box.

7. Once the ARX file has been loaded, at the Command prompt, enter **dl_objectarx**.
A line is drawn from 0,0 to 150,150 on the current layer.
8. At the Command prompt, enter **mytext_objectarx**.
9. Specify two points to define the location and width of the MText object, and then enter some text into the Multiline Text editor. When done, click outside of the editor to create the MText object.

The text is placed on the TS-TEXT2 layer and assigned the SANSSERIFBOLD style.

6 Where to Get More Information

Autodesk University offers a wealth of information and learning materials for customizing and automating workflows in AutoCAD. Over the years, I have presented on many topics that focus on customizing AutoCAD and automating workflows. Here is a partial list of those sessions that you might find useful based on the customization and programming features mentioned in this session:

- **AutoCAD Customization Boot Camp: Basic (No Experience Required)** –
<https://www.autodesk.com/autodesk-university/class/AutoCAD-Customization-Boot-Camp-Basic-No-Experience-Required-2020>
- **AutoCAD Customization Boot Camp: Automate Workflows and Tasks** –
<https://www.autodesk.com/autodesk-university/class/AutoCAD-Customization-Boot-Camp-Automate-Workflows-and-Tasks-2020>
- **Managing and Migrating Custom Files and Settings** –
<https://www.autodesk.com/autodesk-university/class/Managing-and-Migrating-Custom-Files-and-Settings-2020>
- **Introduction to Visual Basic® for Applications for AutoCAD** –
<https://www.autodesk.com/autodesk-university/class/Introduction-Visual-BasicR-Applications-AutoCAD-2013>
- **Sparring with ObjectARX Round 1: Learning the Ropes** –
<https://www.autodesk.com/autodesk-university/class/Sparring-ObjectARX-Round-1-Learning-Ropes-2014>
- **Sparring with ObjectARX Round 2: Stepping into the Ring** –
<https://www.autodesk.com/autodesk-university/class/Sparring-ObjectARX-Round-2-Stepping-Ring-2014>

Even after reviewing the previously mentioned Autodesk University sessions, you will likely 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 learn more about AutoCAD customization and programming:

- **AutoCAD Online Help System** – The Customization and Developer Guides in the AutoCAD Online Help system contains a lot of information on customizing the program and programming languages mentioned in this session.

To access the online help, go to: <https://help.autodesk.com/view/OARX/2022/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 fellow users and Autodesk employees.

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. From the AutoCAD forums page, click the appropriate subgroup link.

- **AUGI Forums** – The AUGI forums provide peer-to-peer networking where you can ask questions about virtually anything AutoCAD or Autodesk software related and get a response from fellow users. Visit AUGI at <https://www.augi.com>.
- **Industry Events and Classes** – Industry events such as AEC Conference and Autodesk University are great places to learn about industry trends and workflows. Along with industry events, you might also be able to find classes at your local technical college or an Autodesk Authorized Training Center (ATC) to improve your proficiency with AutoCAD.
- **Internet** – There are tutorials and information scattered across the Internet that you can use to learn more about customization and programming options supported in AutoCAD. Use your favorite search engine, such as Google or Bing, to find this information by using the feature and utility names mentioned throughout this handout.
- **Books** – There are some general and specialized books that cover AutoCAD customization and programming. To find which books are available, 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 most recent book, *AutoCAD Platform Customization: User Interface, AutoLISP, VBA, and Beyond*, covers many of all the customization and programming options mentioned in this session and many more.