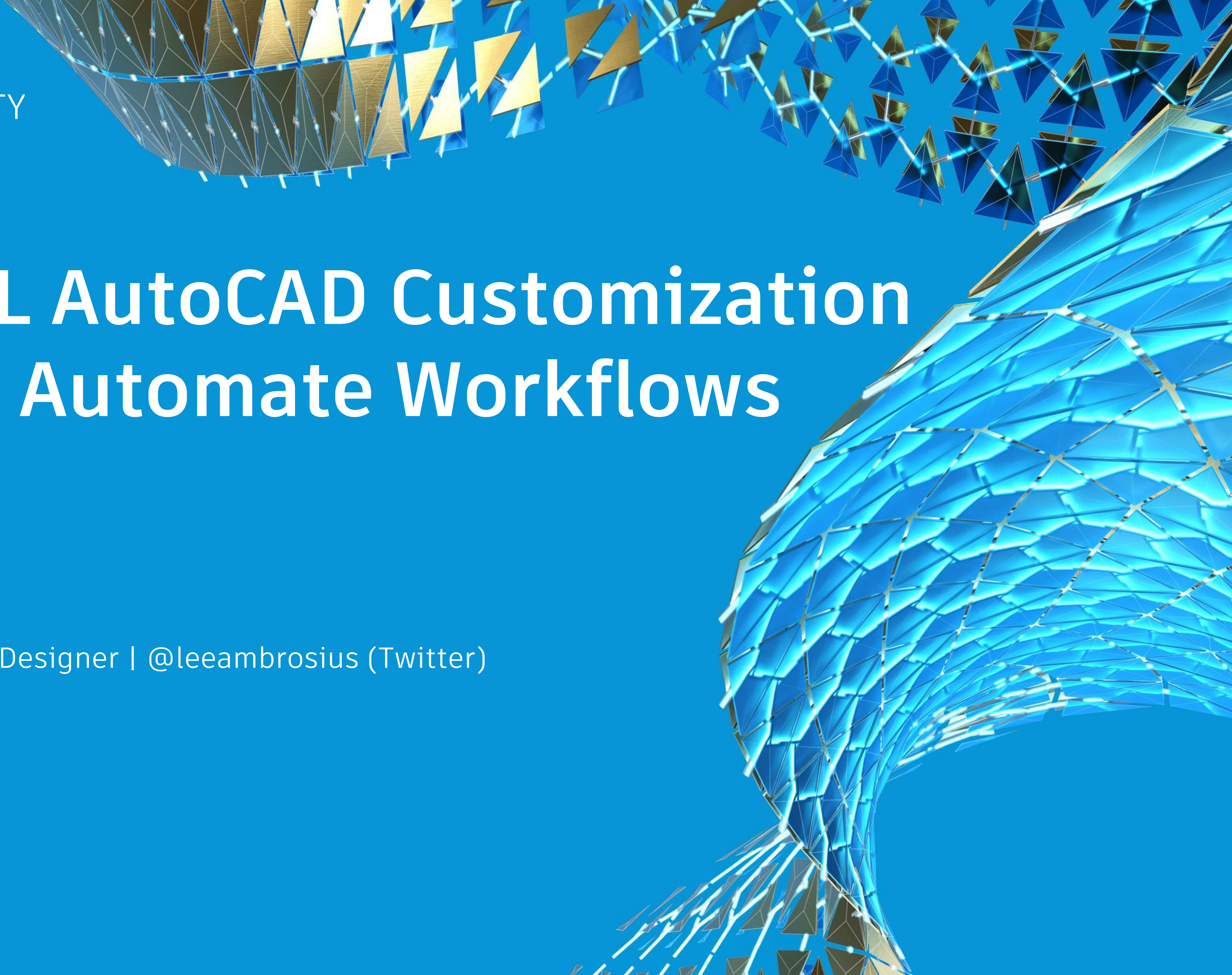
AUTODESK® UNIVERSITY



AS468504-L AutoCAD Customization Boot Camp: Automate Workflows and Tasks

Lee Ambrosius

Principal Learning Experience Designer | @leeambrosius (Twitter)

Who’s this Session For

Those that want to learn how to:

- Automate workflows and tasks in AutoCAD
- Create custom commands with
    - Action macros
    - AutoLISP programs

What you should already know:

- AutoCAD 2021 (or AutoCAD 2016 and later)
- How to use commands and system variables

NO prior programming experience is required

About the Speaker

My name is Lee Ambrosius:

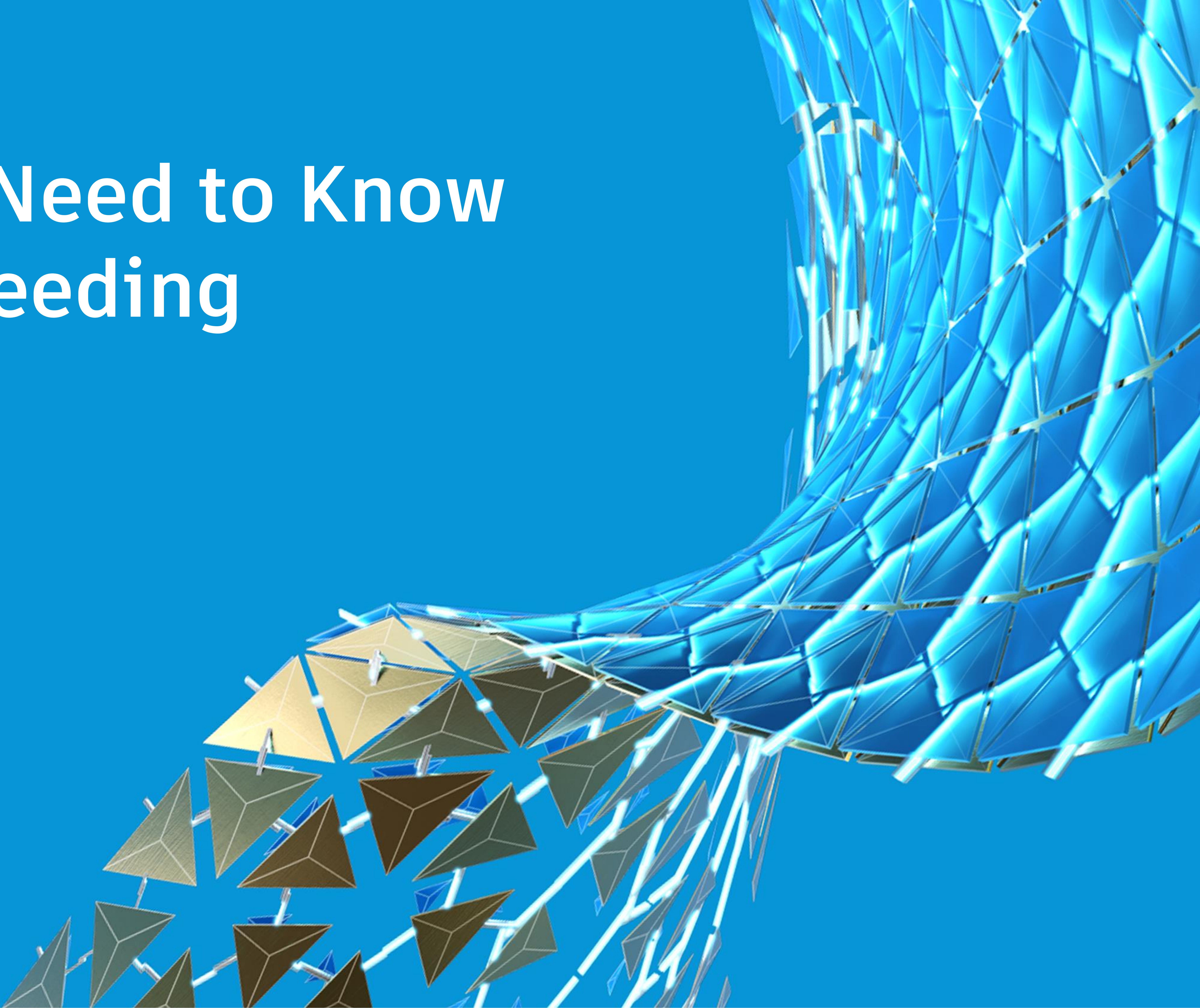
- **Principal Learning Experience Designer at Autodesk, Inc.**
  - Technical writer and data analyst
  - Customization, Developer, and CAD Administration documentation
- **Over 20+ years of AutoCAD customization and programming experience**
- **Authored AutoCAD Customization Platform book series published by Wiley & Sons**

My job in a nutshell:

- **Document the past and present AutoCAD releases for the future**

Yeah, running 48.6 miles in 4 Days is Dopey

Things You Need to Know Before Proceeding



What You Need to Get Started

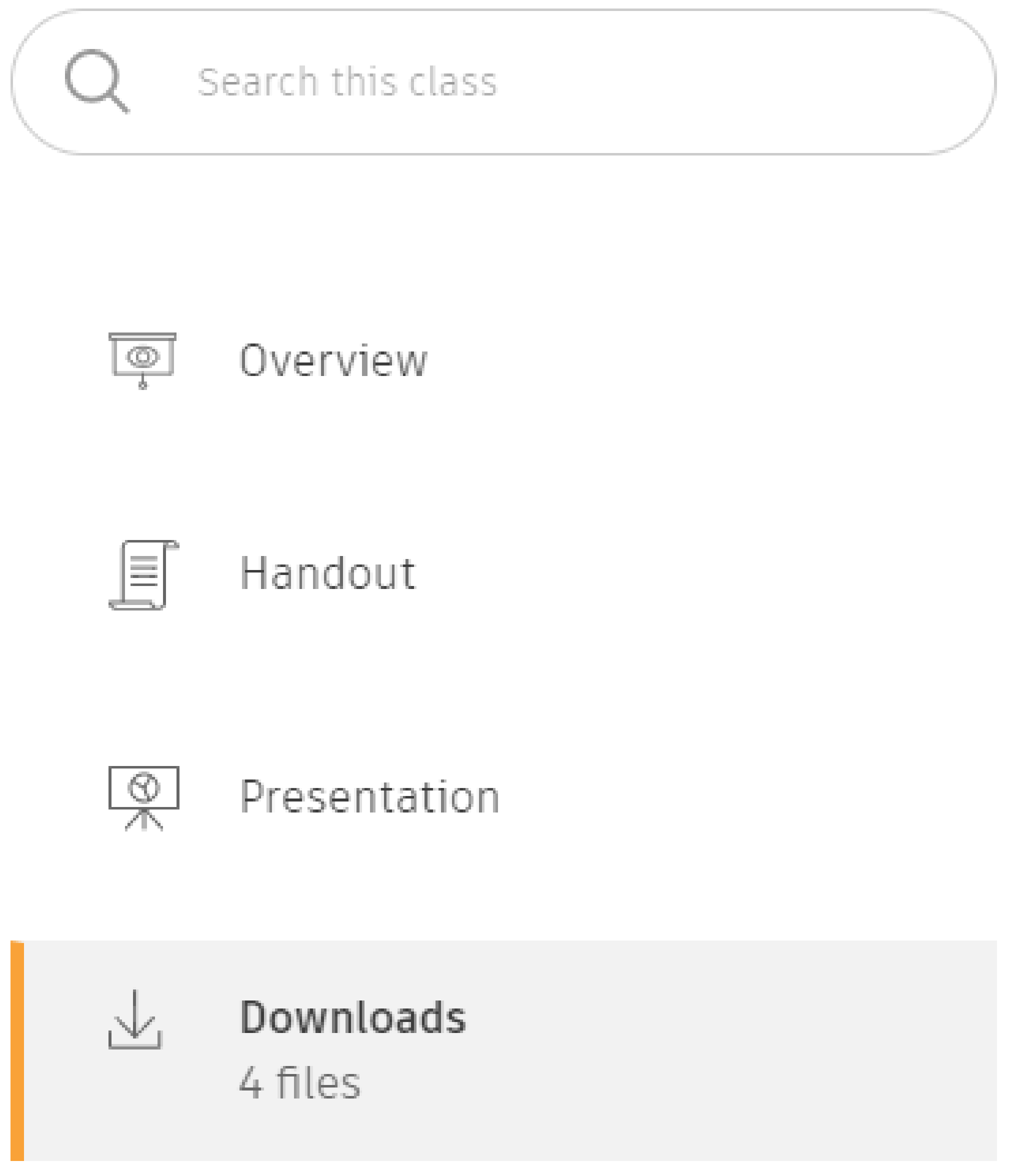
For this session, you will need:

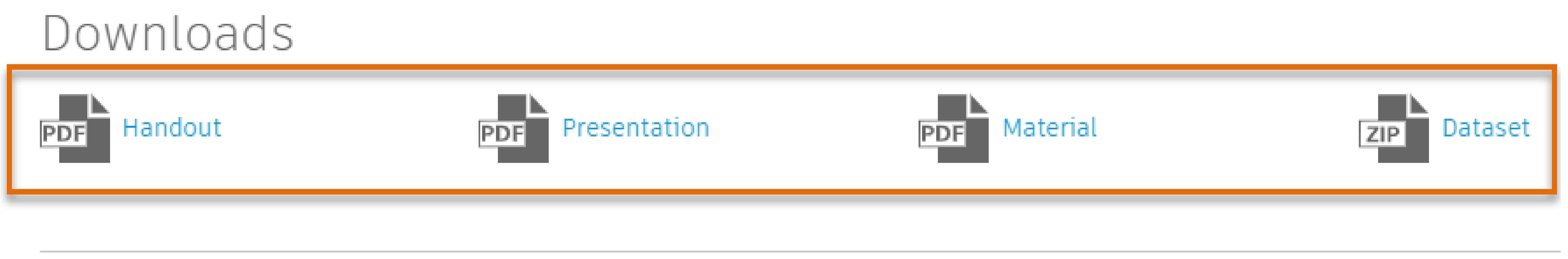
- **AutoCAD 2021 (or AutoCAD 2016 and later)**
- **Action Recorder**
- **Notepad**
- **Materials for this session from the AU website**
  - Dataset
  - Handout
  - Supplemental handout

Setting Up for this Session

Materials for this session can be obtained by:

1. Going to the Autodesk University website and search on this session's ID of AS468504.
2. In the search results, click the entry for this session.
3. On the session page, click Downloads and then download
   a. Dataset
   b. Handout
   c. Material



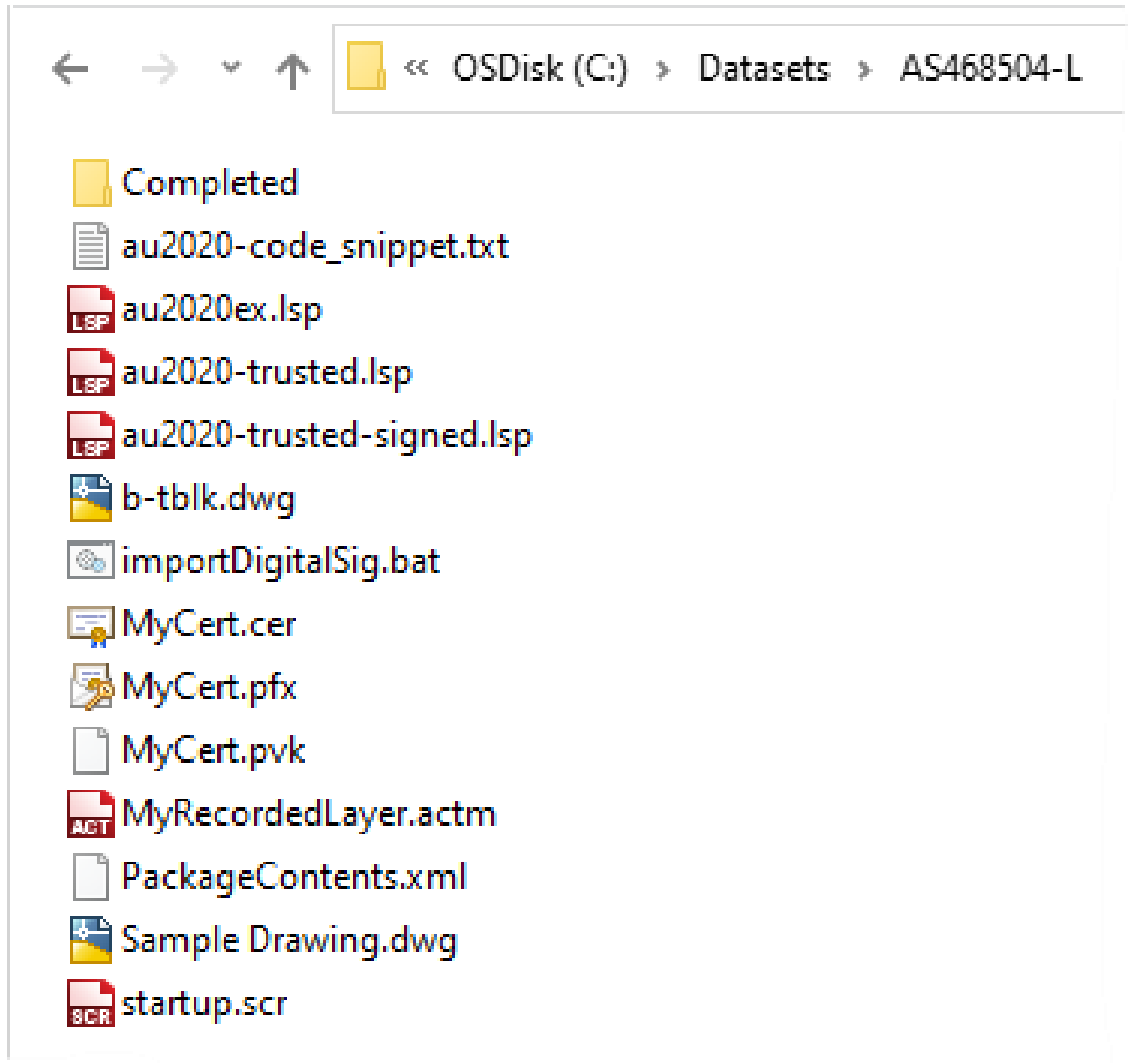


Setting Up for this Session

For this session:

- **Extract the Dataset to**

  C:\Datasets\AS468504-L



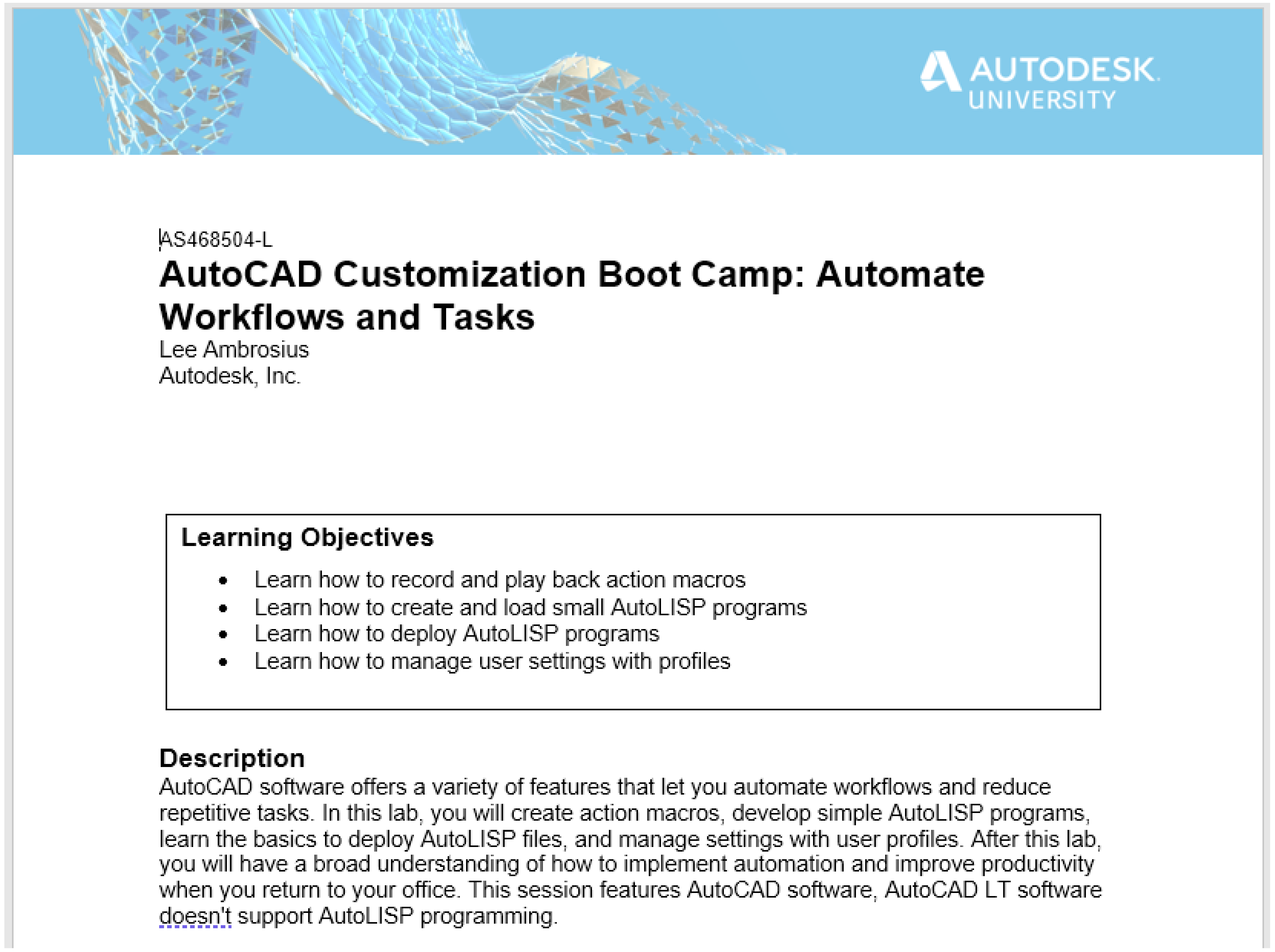
« OSDisk (C:) › Datasets › AS468504-L

Completed
au2020-code_snippet.txt
au2020ex.lsp
au2020-trusted.lsp
au2020-trusted-signed.lsp
b-tblk.dwg
importDigitalSig.bat
MyCert.cer
MyCert.pfx
MyCert.pvk
MyRecordedLayer.actm
PackageContents.xml
Sample Drawing.dwg
startup.scr

Setting Up for this Session

For this session:

- **Extract the Dataset to**
  C:\Datasets\AS468504-L
- **Open the handout**


AUTODESK UNIVERSITY

AS468504-L

AutoCAD Customization Boot Camp: Automate Workflows and Tasks

Lee Ambrosius
Autodesk, Inc.

Learning Objectives

- Learn how to record and play back action macros
- Learn how to create and load small AutoLISP programs
- Learn how to deploy AutoLISP programs
- Learn how to manage user settings with profiles

Description

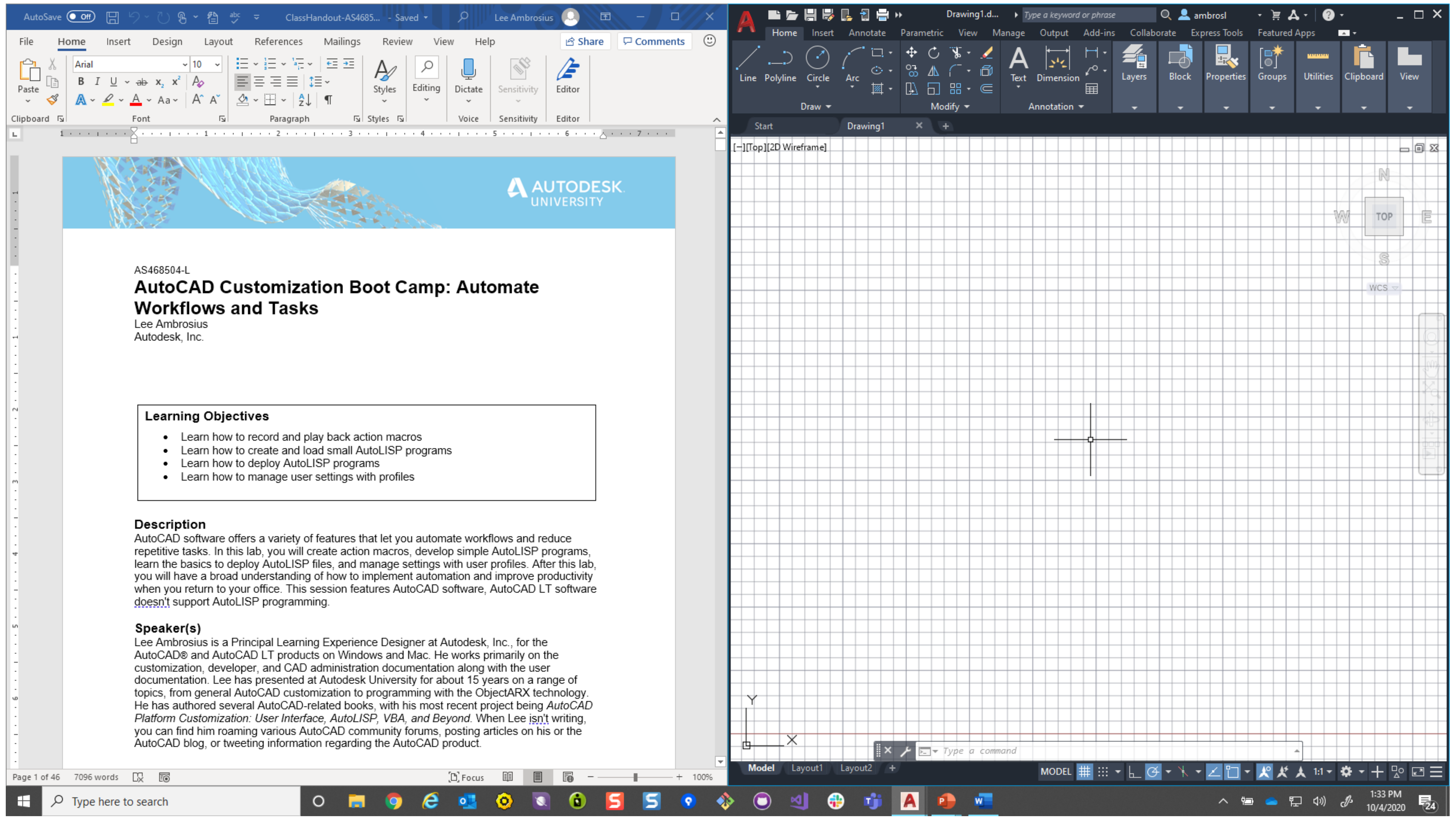
AutoCAD software offers a variety of features that let you automate workflows and reduce repetitive tasks. In this lab, you will create action macros, develop simple AutoLISP programs, learn the basics to deploy AutoLISP files, and manage settings with user profiles. After this lab, you will have a broad understanding of how to implement automation and improve productivity when you return to your office. This session features AutoCAD software, AutoCAD LT software doesn't support AutoLISP programming.

Setting Up for this Session

For this session:

- Extract the Dataset to

  C:\Datasets\AS468504-L
- Open the handout
- Recommend snapping the handout and AutoCAD side by side



AS468504-L

AutoCAD Customization Boot Camp: Automate Workflows and Tasks

Lee Ambrosius

Autodesk, Inc.

Learning Objectives

- Learn how to record and play back action macros
- Learn how to create and load small AutoLISP programs
- Learn how to deploy AutoLISP programs
- Learn how to manage user settings with profiles

Description

AutoCAD software offers a variety of features that let you automate workflows and reduce repetitive tasks. In this lab, you will create action macros, develop simple AutoLISP programs, learn the basics to deploy AutoLISP files, and manage settings with user profiles. After this lab, you will have a broad understanding of how to implement automation and improve productivity when you return to your office. This session features AutoCAD software, AutoCAD LT software doesn't support AutoLISP programming.

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.

E0 - Download and Setup the Dataset Folder and Add it to AutoCAD's Support File Search Path

In this exercise, you will:

- Add the dataset folder to the current AutoCAD profile

Follow along with the video or go to page 3 of the handouts.

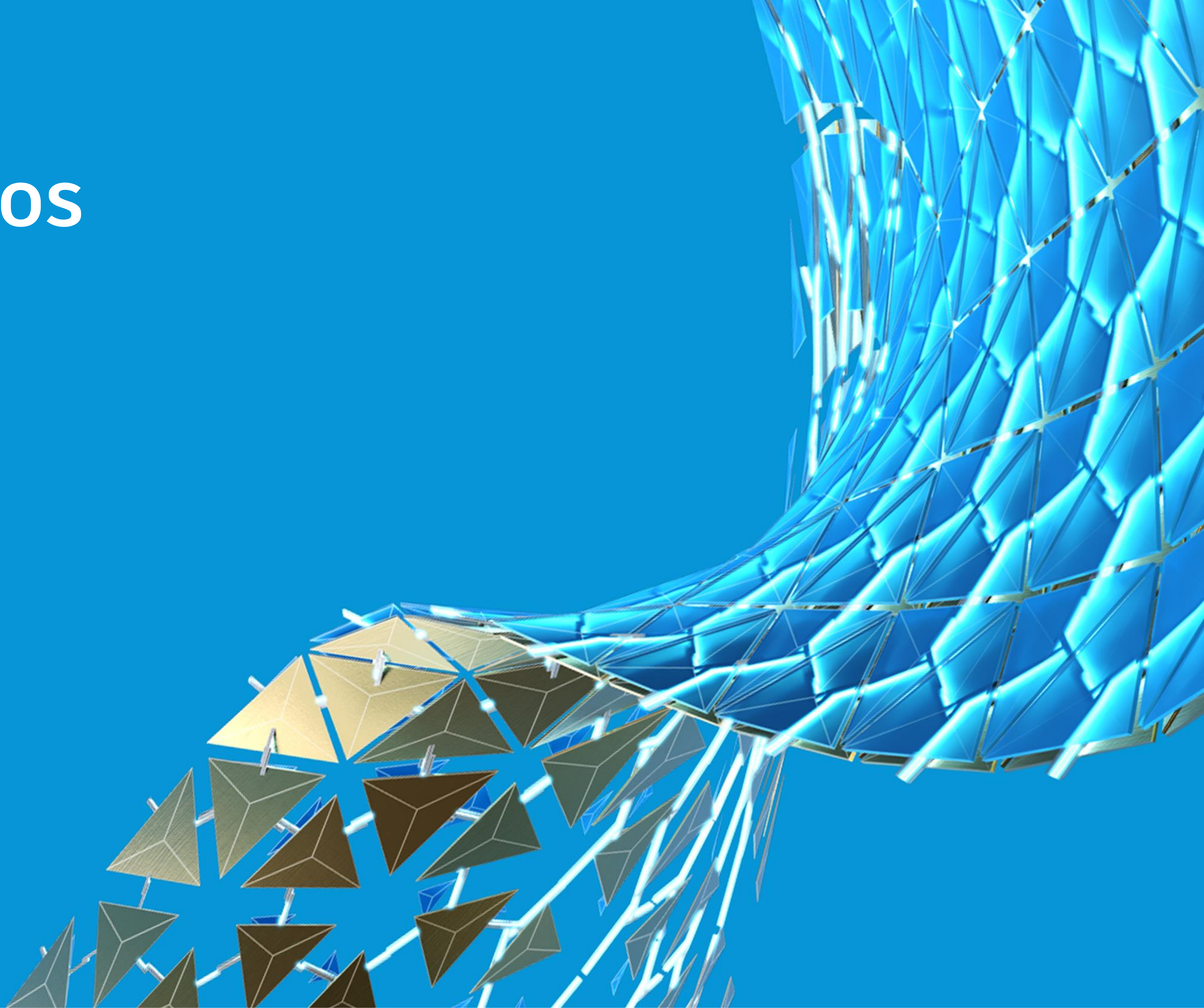
Welcome to Specialist Training

What You Will Learn Today

At the end of this session, you will know how to:

- Record and playback an action macro
- Write basic AutoLISP programs
- Load and deploy AutoLISP programs
- Create and set a user profile current

Action Macros

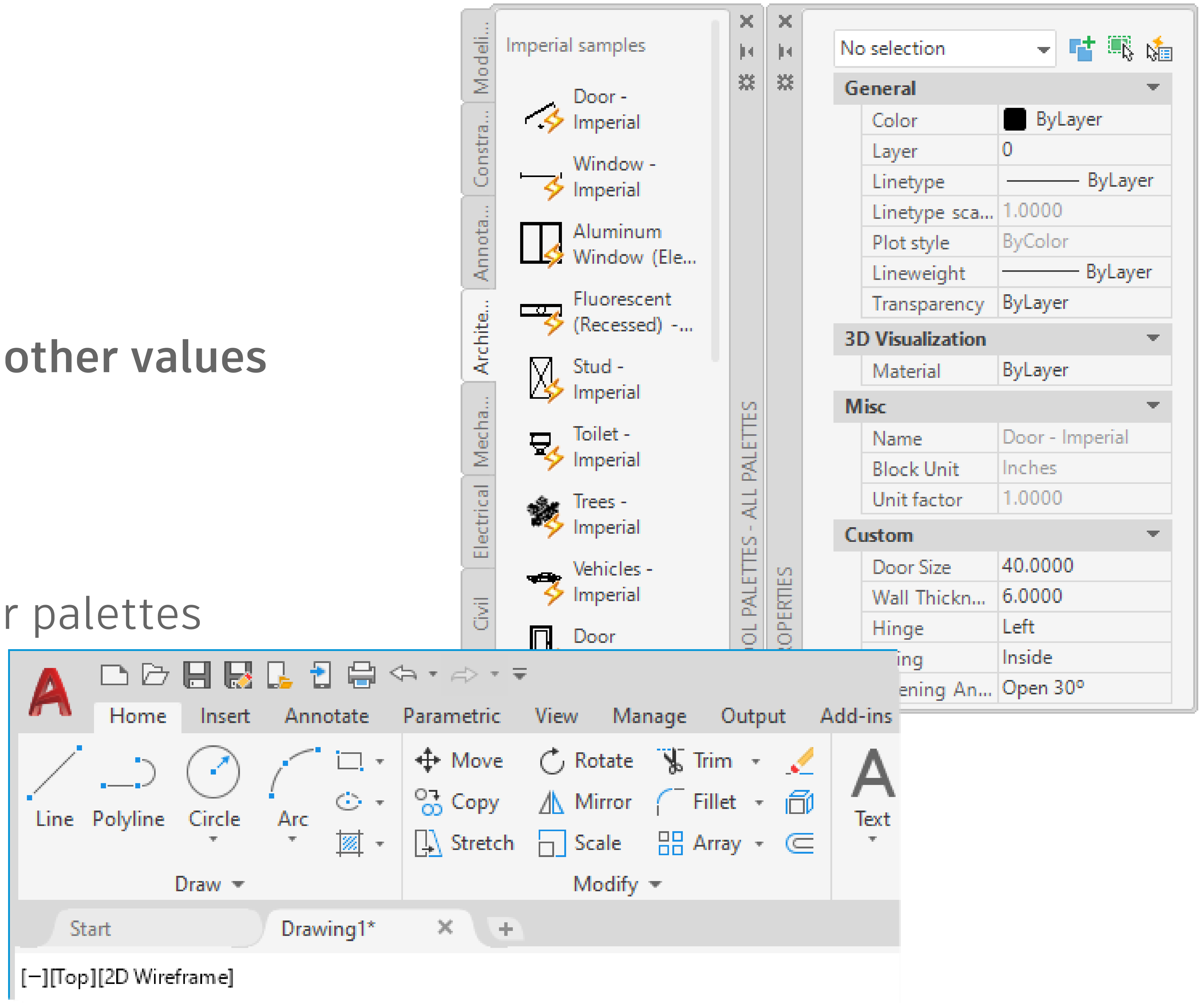


Action Macros – Actions

Smallest interaction that can be recorded

An action can be:

- **Starting of a command**
- **Specifying of coordinate, object selection, or other values**
- **Interactions performed with/on the**
  - Properties and Quick Properties
  - Tool Palettes and Layer Properties Manager palettes
  - Quick Access toolbar and ribbon
  - Status bar



No selection

General	
Color	■ ByLayer
Layer	0
Linetype	ByLayer
Linetype sca...	1.0000
Plot style	ByColor
Lineweight	ByLayer
Transparency	ByLayer
3D Visualization	
Material	ByLayer
Misc	
Name	Door - Imperial
Block Unit	Inches
Unit factor	1.0000
Custom	
Door Size	40.0000
Wall Thickn...	6.0000
Hinge	Left
...ing	Inside
...ening An...	Open 30º

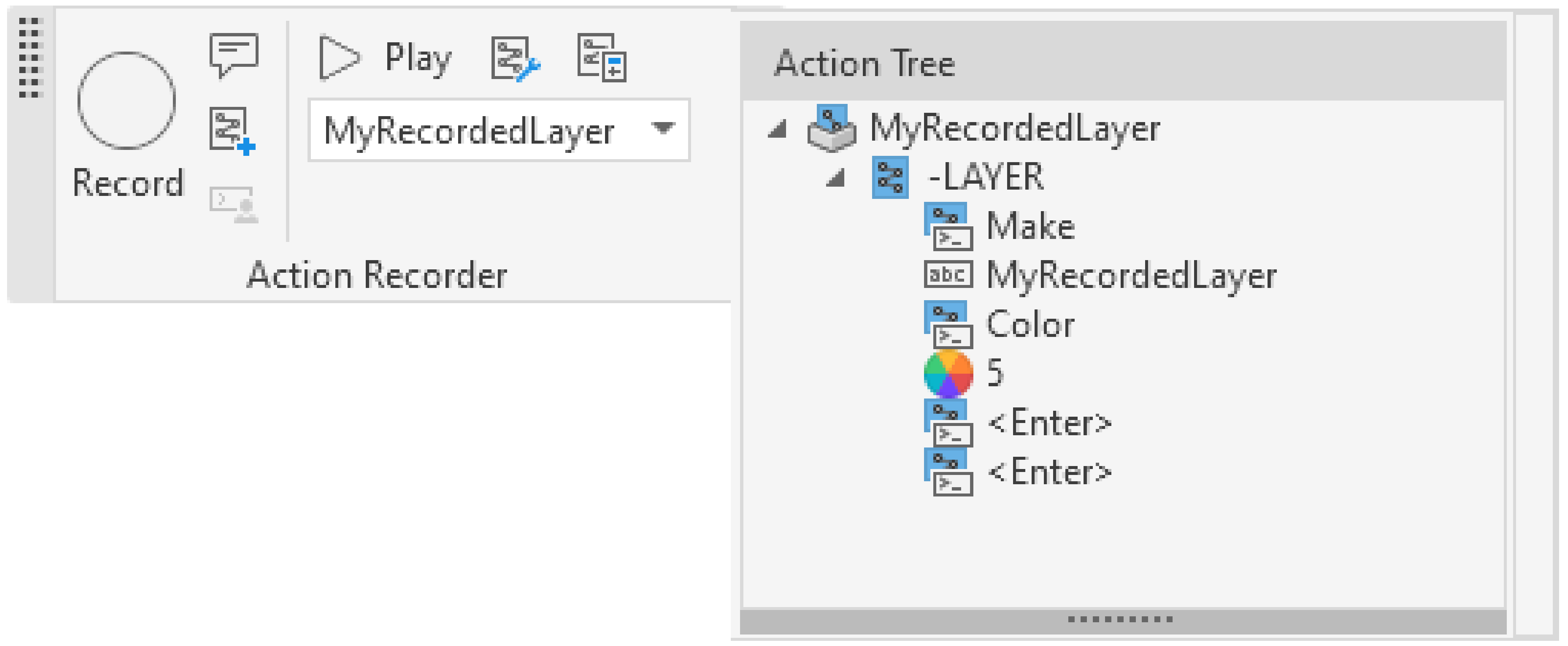
Action Macros – Recording

Actions are recorded with the Action Recorder on the ribbon

Saved to action macro (ACTM) files

Things to know before recording begins:

- Recommended to avoid dialog boxes
- System variable values can be changed

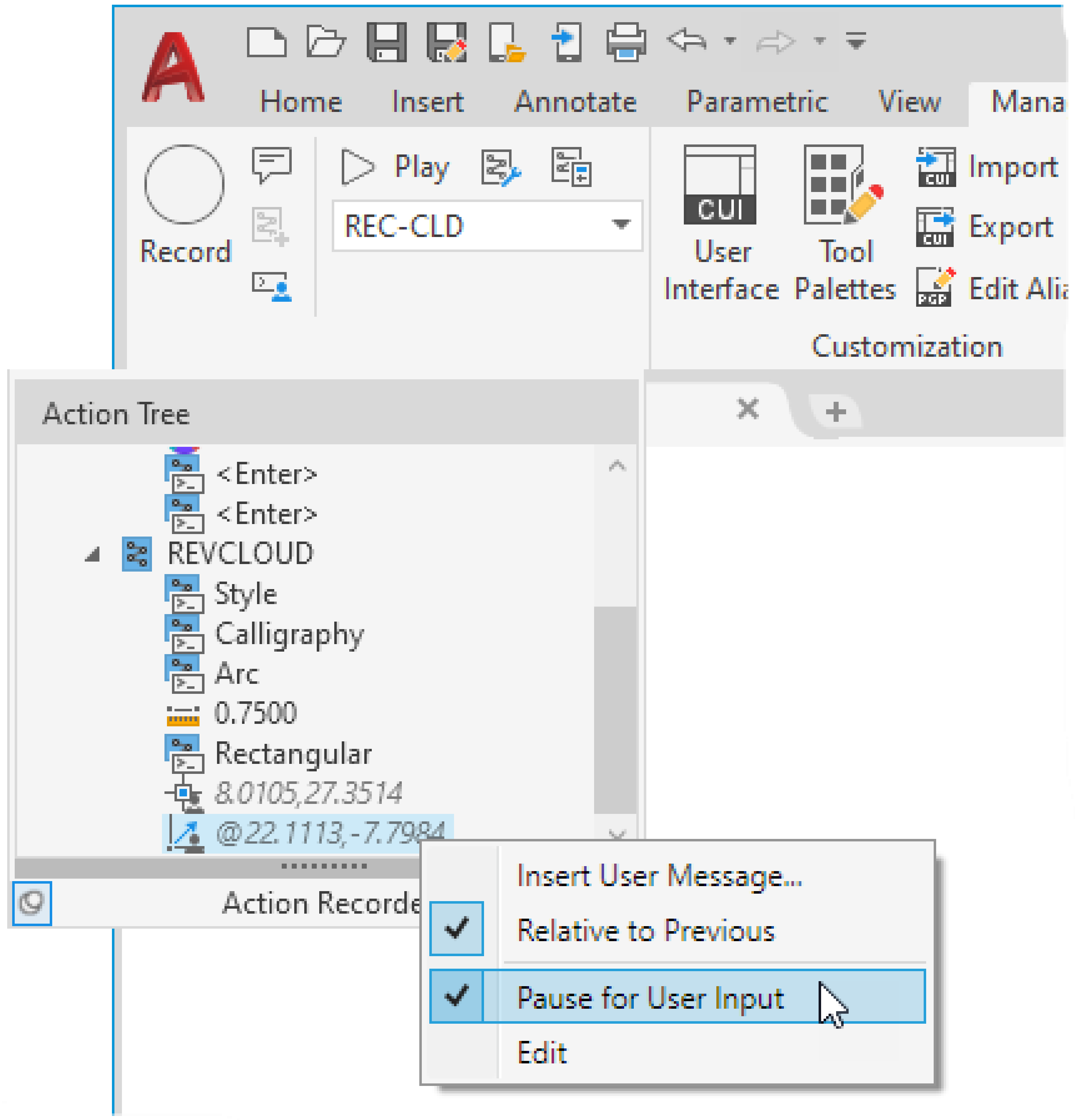


MyRecordedLayer
-LAYER
Make
MyRecordedLayer
Color
5
<Enter>
<Enter>

Action Macros – User Interactions

User interactions can be added to alter the playback of an action macro:

- Display a user message
- Prompt for a value, selection set, or point
- Use the currently selected objects



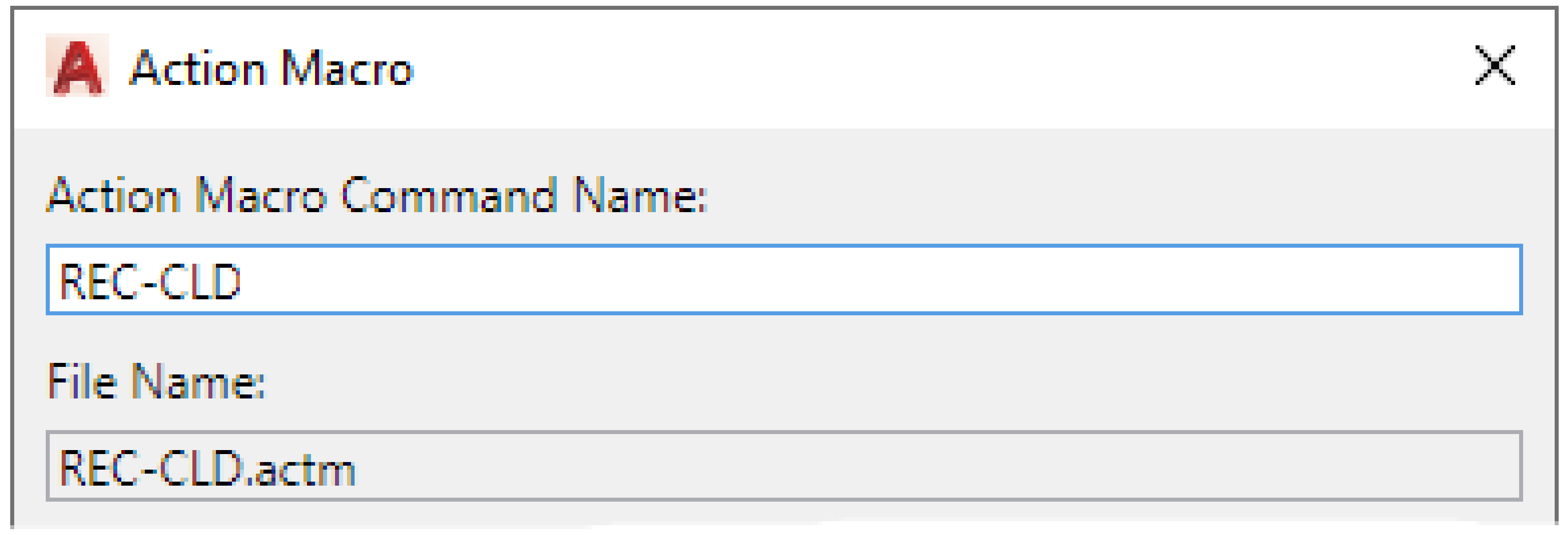
Action Macros – Playing Back

Once saved, recorded action macros can be played back by:

- Entering its name at the Command prompt
- Selecting and playing it from the Action Recorder panel
- Choosing it from the drawing window shortcut menu

Action macros can be shared with others:

- Place them in a common location
- Record only commands that are available to all users



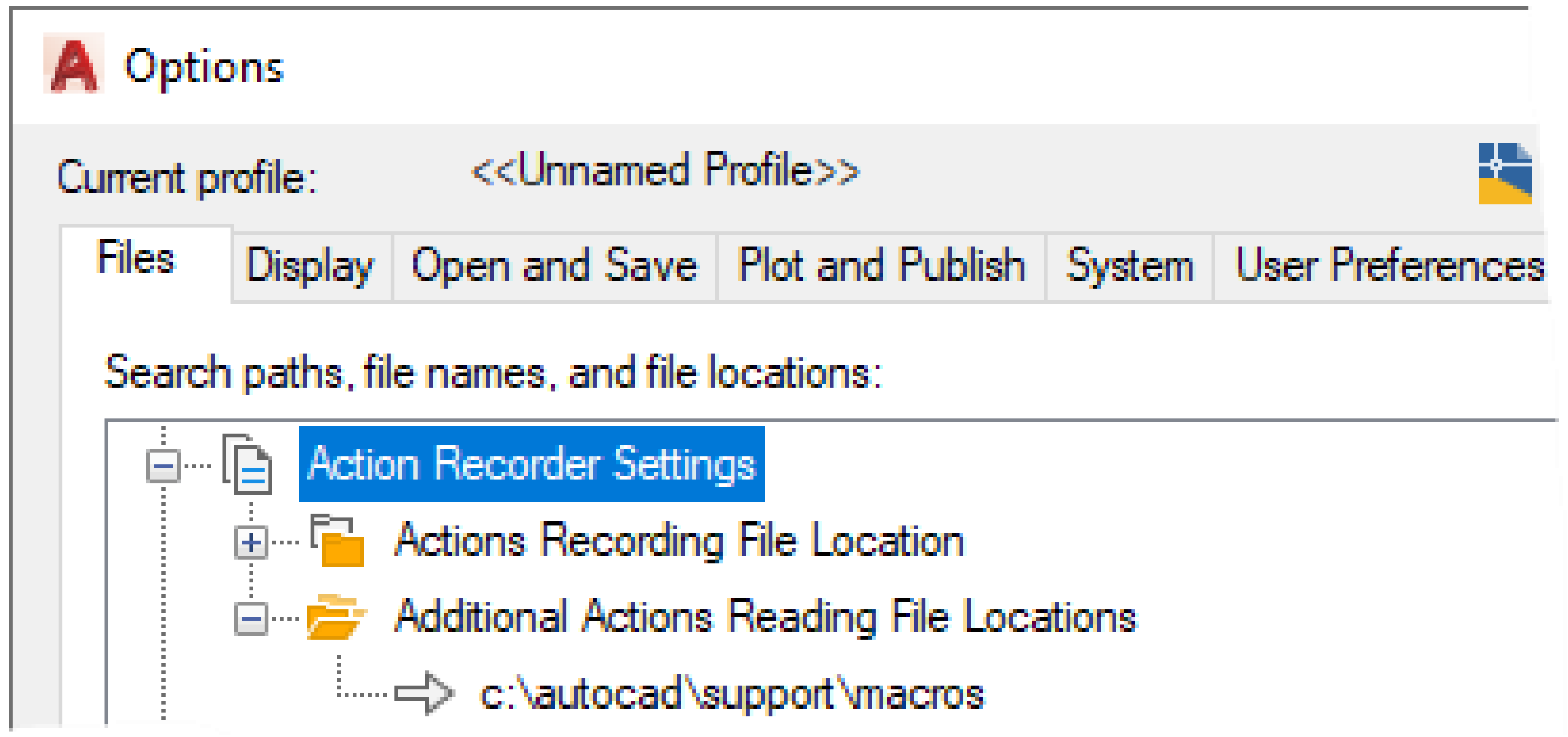
Action Macro

Action Macro Command Name:

REC-CLD

File Name:

REC-CLD.actm



Options

Current profile: <<Unnamed Profile>>

Search paths, file names, and file locations:

- Action Recorder Settings
  - Actions Recording File Location
  - Additional Actions Reading File Locations
    - c:\autocad\support\macros

To Record an Action Macro

1. Start recording from the Action Recorder.
2. Perform the actions in the application and drawing windows you want to record.
3. Stop recording and save the action macro.
4. Edit the actions that were recorded.
5. Optionally, add user interactions to the action macro.
6. Playback and test the action macro.

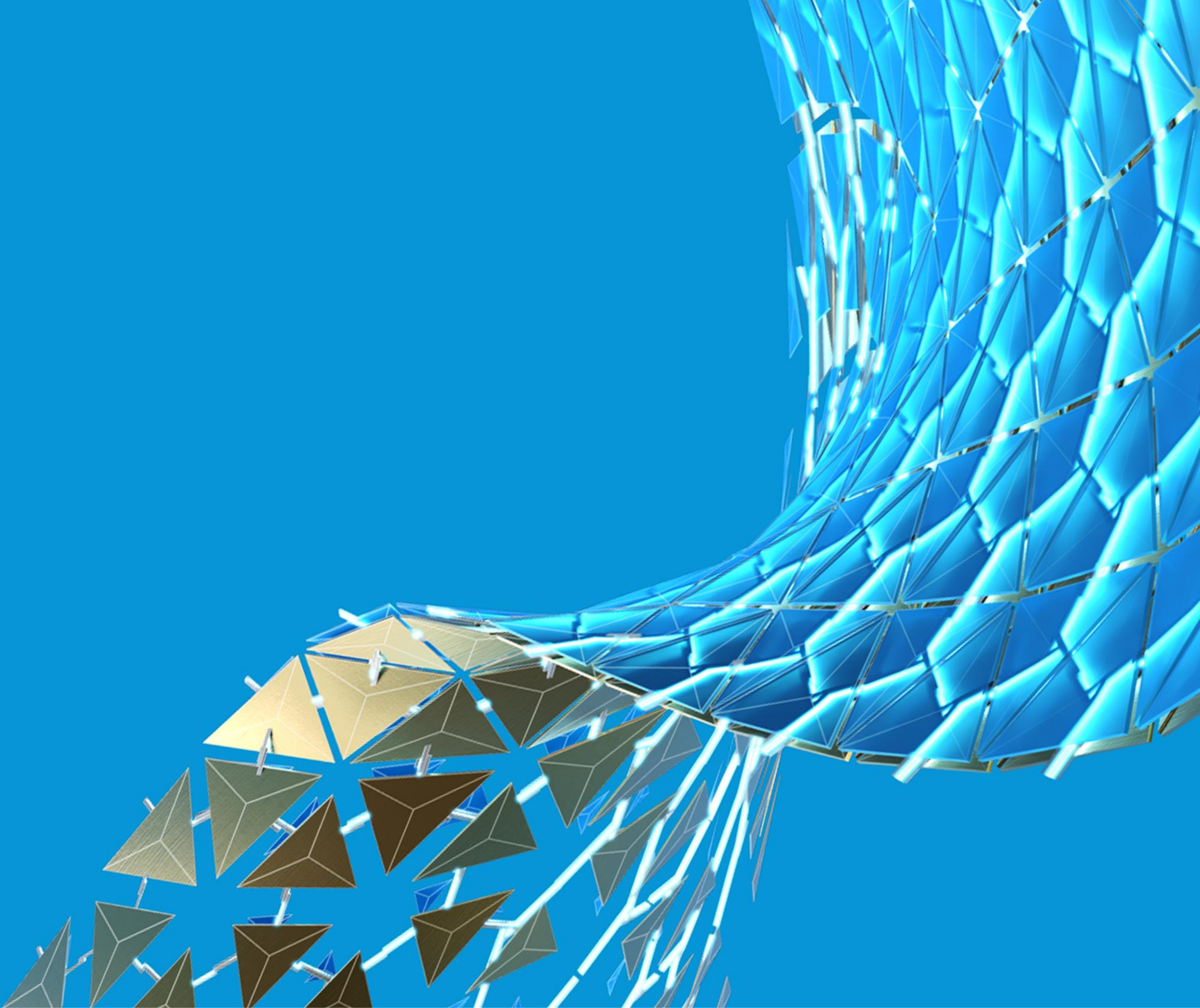
Exercise: E1.A - Record and Playback an Action Macro

In this exercise, you will:

- Record actions performed at the Command prompt
- Save and modify an action macro
- Playback a recorded action macro

Follow along with the video or go to page 6 of the handouts.

AutoLISP



AutoLISP

Programming language

- Based on the LISP (LISt Processing) programming language
- Specific to AutoCAD and AutoCAD-based programs
- 30+ years (January 1986) old, introduced in AutoCAD Version 2.18
- Doesn't require a specialized editor
- Doesn't need to be compiled; interpreted language

AutoLISP – Expressions

AutoLISP expressions can be

- Entered directly at the Command prompt in AutoCAD
- Stored and loaded from a LSP file
- Written using Notepad or the Visual LISP Editor
- Compiled as a FAS or VLX file to protect the source code

AutoLISP – Expressions

AutoLISP expressions must:

- **Start with (**
- **End with)**

Example:

```
    (prompt "\nHello AU 2020!")
```

AutoLISP - Syntax

Syntax of an AutoLISP expression:

```
    (function_name argumentX)
```

- **`function_name`** – Name of the function
- ***`argumentX`*** – Value(s) the function should do something with

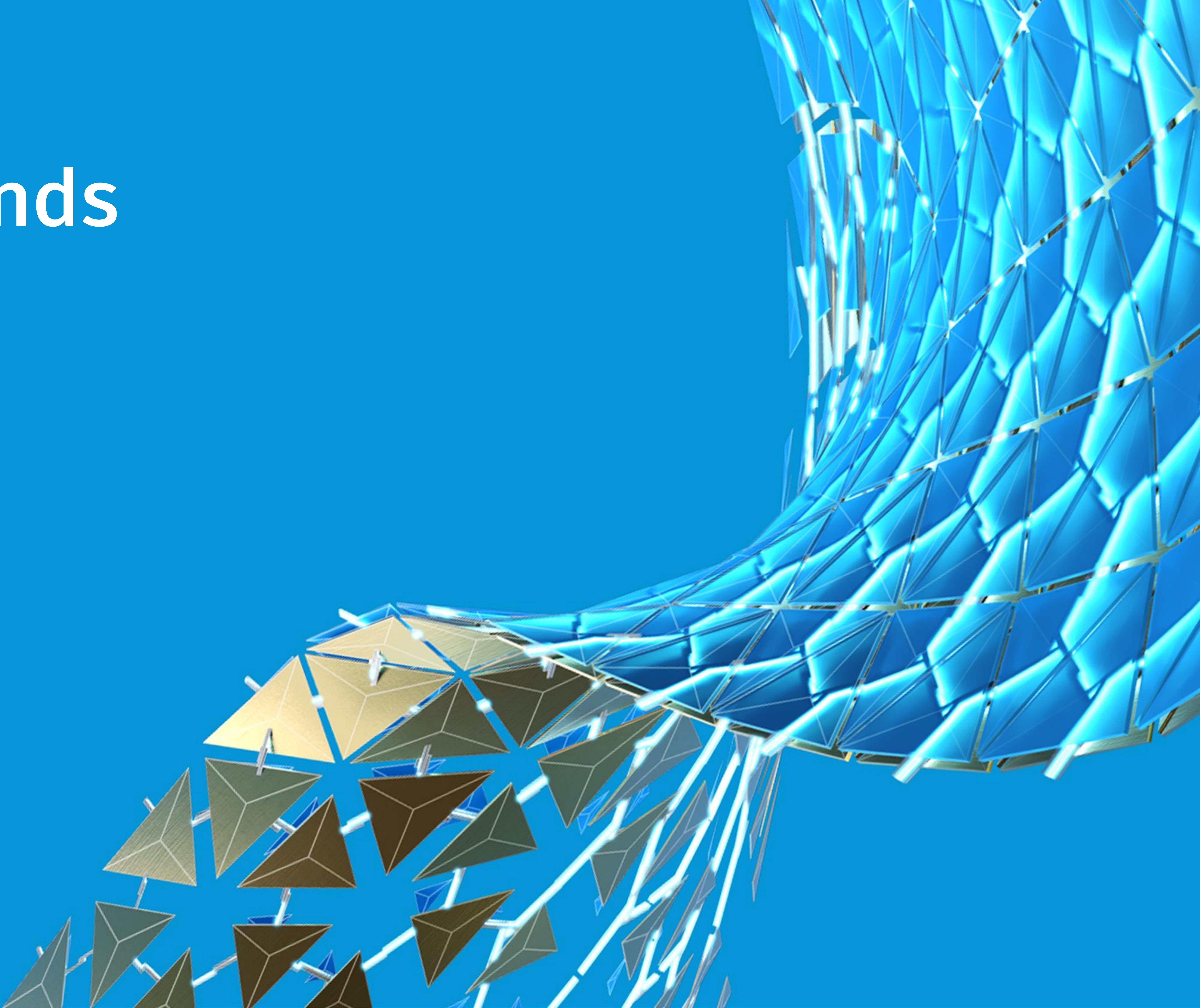
Not all functions except arguments

AutoLISP – Common Functions

Functions you should know when getting started:

- `command` – Executes an AutoCAD command
- `setq` – Assigns a value to user-defined variable
- `setvar` – Assigns a value to a system variable
- `getvar` – Gets a system variable’s current value
- `defun` – Creates a user-defined function

Use Commands



`command` Function – Syntax

Executes a command

Syntax:

```
    (command command_name valueX)
```

- **command_name** – Name of the command to execute
- ***valueX*** – Option(s) and value(s) the command expects

`command` Function – LINE Example

Example of input entered at the Command prompt

```
Command: line
Specify first point: 0,0
Specify next point or [Undo]: 5,5
Specify next point or [Undo]:
```

Same input as an AutoLISP statement

```
(command "line" "0,0" "5,5" "")
```

`command` Function – CIRCLE Example

Example of input entered at the Command prompt

```
Command: circle
Specify center point for circle or [3P/2P/Ttr (tan tan radius)]: 0,0
Specify radius of circle or [Diameter] <0.0000>: d
Specify diameter of circle <0.0000>: 5
```

Same input as an AutoLISP statement

```
(command "circle" "0,0" "d" 5)
```

`command` Function – Special Values

Special values used with the `command` function:

- "" – Represents a press of the Enter key
- PAUSE – Instructs AutoCAD to wait for input

```
(command "circle" PAUSE "d" 5)
```

Store and Work with Data Values

`setq` Function – Syntax

Assigns a value to a user-defined variable

Syntax:

```
    (setq variable_name value)
```

- variable_name – Name of user-defined variable to create or update
- *value* – Value to assign

`setq` Function – Examples

Examples:

Assigns a numeric value of 1.25 to the *dRadius* variable

```
(setq dRadius 1.25)
```

Assigns a text string of AU 2020 to the *strEvent* variable

```
(setq strEvent "AU 2020")
```

Assigns the two previous examples with the same statement

```
(setq dRadius 1.25 strEvent "AU 2020")
```

`setq` Function – Examples

Examples:

Assigns a numeric value of 1.25 to the *dRadius* variable

```
(setq dRadius 1.25)
```

Assigns a text string of AU 2020 to the *strEvent* variable

```
(setq strEvent "AU 2020")
```

Assigns the two previous examples with the same statement

```
(setq dRadius 1.25 strEvent "AU 2020")
```

`setq` Function – Examples

Examples:

Assigns a numeric value of 1.25 to the *dRadius* variable

```
(setq dRadius 1.25)
```

Assigns a text string of AU 2020 to the *strEvent* variable

```
(setq strEvent "AU 2020")
```

Assigns the two previous examples with the same statement

```
(setq dRadius 1.25 strEvent "AU 2020")
```

`setq` Function – Get a Variable’s Value

Prefix a variable name with an ! (Exclamation point) to return its value

Example:

```
Command: (setq dRadius 1.25)

Command: !dRadius

1.25
```

! not needed to use a variable’s value

```
(command "circle" "0,0" "d" dRadius)
```

`setvar/getvar` Function – Syntax

Set or get the value of a system variable

Syntax:

```
    (setvar sysvar_name value)

    (getvar sysvar_name)
```

- **sys_name** – Name of system variable to work with
- ***value*** – Value to assign

`setvar/getvar` Function – Examples

Examples:

Gets the value of OSMODE

```
(setq nOSMODE (getvar "osmode"))
```

Sets the value of OSMODE to END (1) and INT (32)

```
(setvar "osmode" 33)
```

Pause – Mental Break

So far, you have learned you can work with:

- Commands using the `command` function
- Store values with the `setq` function
- Work with system variables using the `setvar` and `getvar` functions

To get this far, you just had to learn a few things:

- Importance of the open and closing parenthesis
- Fundamentally, how values are passed to functions

AutoLISP – Data Types

Functions accept many different types of data:

- **Integer** – Any number without a decimal point

  Examples: `12, 0`

- **Real** – Any number with a decimal point

  Examples: `12.125, 0.0`

- **String** – Any alphanumeric characters enclosed in double quotes

  Examples: `"12.125", "Welcome to AU 2020!"`

AutoLISP Data Types (cont.)

Additional types of data:

- **List** – Any expression in parentheses

  Examples: `(0.0 5.0 0.0)`

  ```
  (command "line" "0,0" "5,5" "")
  ```

- **Symbol** – Internal or user-defined variables

  Examples: `PAUSE`, *`dRadius`*

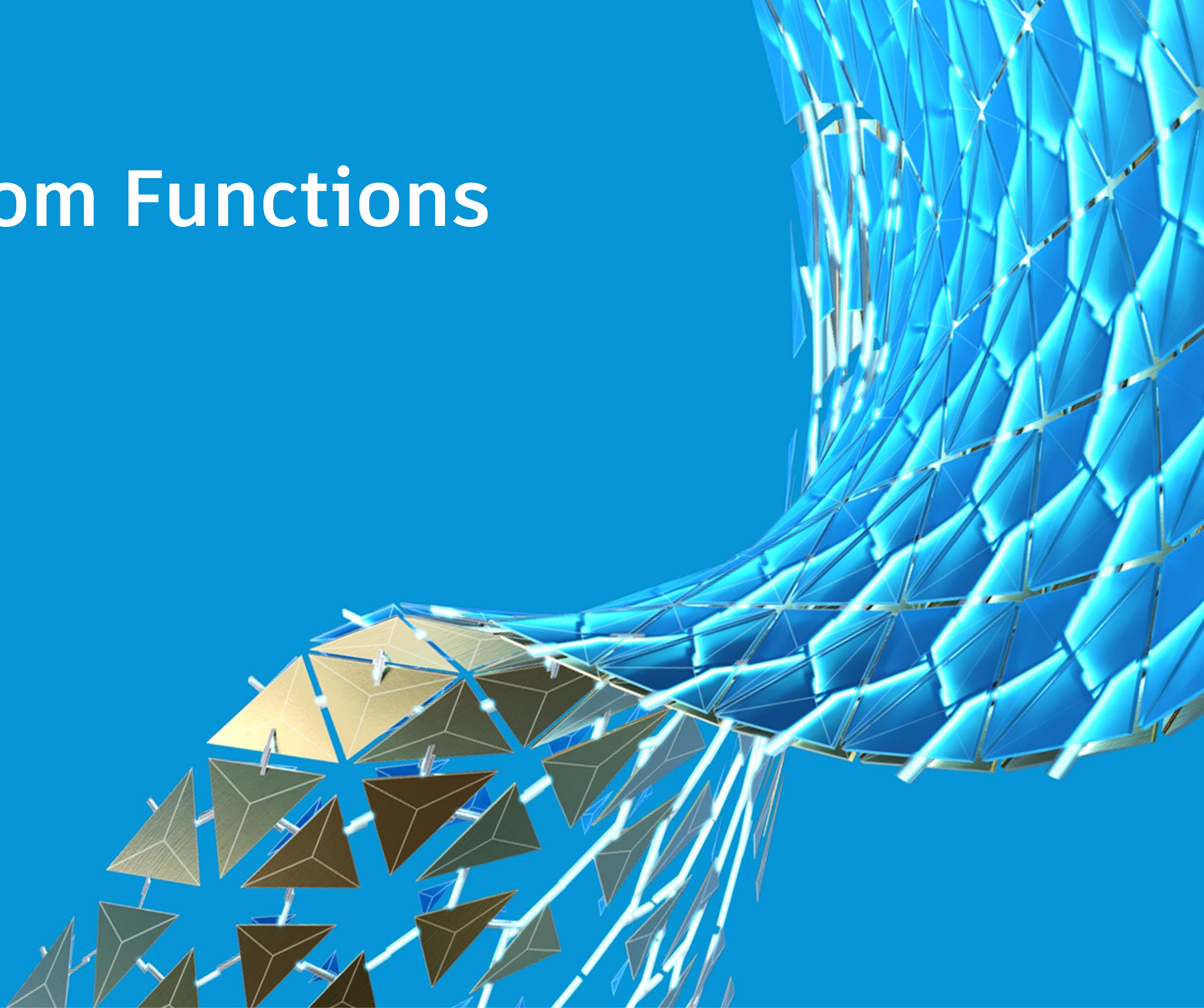
Exercise: E2 - Enter AutoLISP Expressions at the Command Prompt

In this exercise, you will:

- Enter AutoLISP expressions at the Command prompt
- Execute commands
- Store values in user-defined variables

Follow along with the video or go to page 17 of the handouts.

Define Custom Functions



Define Custom Functions

Reusable custom functions can be defined

A custom function is:

- Defined with the `defun` function
- Executed similar to standard AutoCAD commands
- Used to build standardized components for complex programs

`defun` Function – Syntax

Syntax:

```
(defun c:function_name ( / )
          expressionX
 )
```

- **function_name – Name of function to define**
  - Optional, **c:** indicates it can be entered at the Command prompt
- ***expressionX*** **– Expressions to execute**

`defun` Function – Examples

Examples:

Creates a function named HelloWord which displays a message box

```
(defun c:HelloWorld ( / )
    (alert "Hello World!")
)
```

Creates a function named ZP which performs a Zoom Previous

```
(defun c:ZP ( / )
    (command "zoom" "_p")
)
```

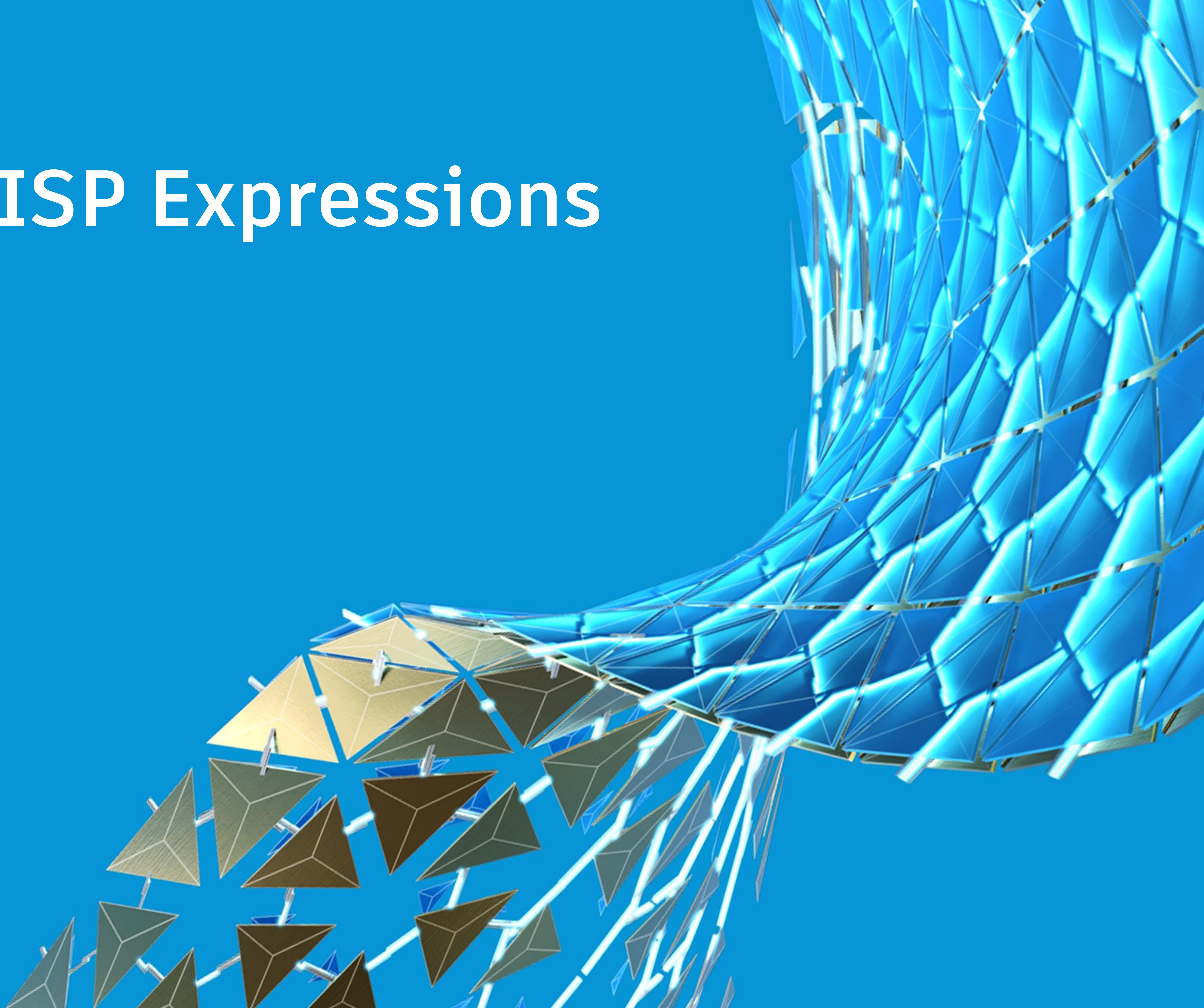
Exercise: E3 - Create Simple Custom Functions

In this exercise, you will:

- Define two custom functions
- Execute the custom functions at the AutoCAD Command prompt

Follow along with the video or go to page 19 of the handouts.

Store AutoLISP Expressions



Store AutoLISP Expressions

Expressions can be stored in a file for re-use:

- **ASCII or Unicode format file with a *.lsp* extension**
  - Unicode support in AutoCAD 2021 only
- **LSP files can be created/modified with**
  - Notepad
  - Visual LISP Editor
  - Visual Studio Code
- **Comments can be added to an LSP file**
- **An LSP file must be loaded into each drawing**

Document AutoLISP Programs

Comments can be added to an AutoLISP file:

- Used to provide information about an LSP file or the expressions in an LSP file
- Indicated by a ; (semi-colon)
- Expressions to the right of a ; are not executed

Examples:

```
    ; Created on: 10/04/2020 by Lee Ambrosius

    (setq dRad 1.25)   ; Default radius value
```

Manually Load a LSP File

These methods can be used to manually load an LSP file:

- APPLOAD command
- AutoLISP `load` function
- Drag and drop an LSP file onto the drawing area (Windows only)

Automatically Load a LSP File

These methods can be used to automatically load an LSP file:

- Startup Suite in the Load/Unload Applications dialog box (APPLOAD command)
- LISP Files node in the Customize User Interface (CUI) Editor (Windows only)
- Menu AutoLISP (MNL) files
- *acad.lsp* and *acaddoc.lsp* files
- Plug-in Bundle

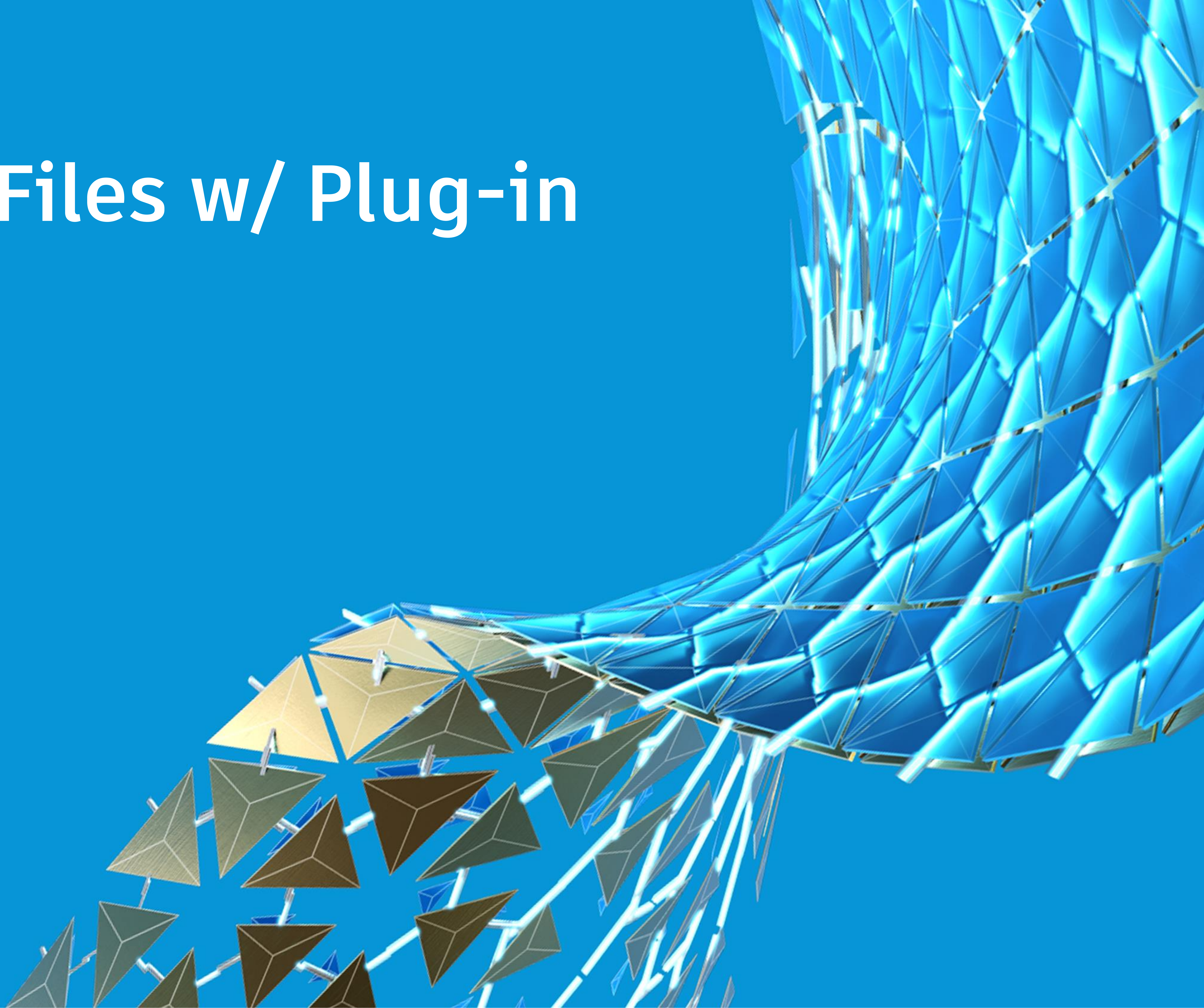
Exercise: E4 - Create and Load a LSP File

In this exercise, you will:

- Create a new LSP file
- Add AutoLISP expressions and comments to an LSP file
- Load an LSP file

Follow along with the video or go to page 20 of the handouts.

Deploy LSP Files w/ Plug-in Bundles



Deploy a LSP File with a Plug-in Bundle

Plug-in bundles:

- Consistent way to deploy and load LSP files
- File and folder structure described by an XML file named *PackageContents.xml*

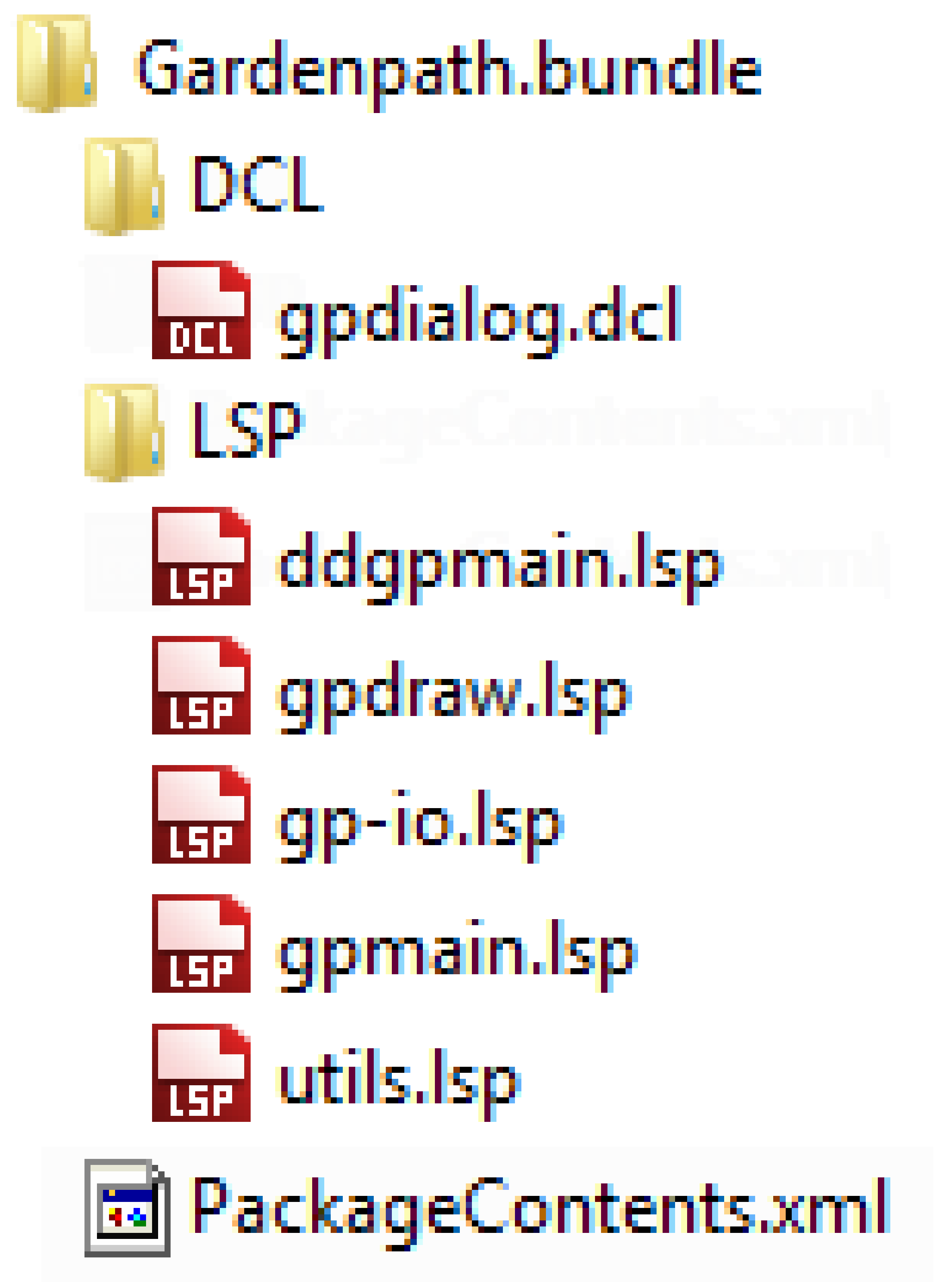
PackageContents.xml:

- Placed in the root folder of each plug-in bundle
- Describes the files in the plug-in bundle

Deploy a LSP File with a Plug-in Bundle

Example structure of a bundle named GardenPath:

```
Gardenpath.bundle
  |-> DCL
       |-> gpdialog.dcl
  |-> LSP
       |-> ddgpmain.lsp
       |-> gpdraw.lsp
       |-> gp-io.lsp
       |-> gpmain.lsp
       |-> utils.lsp
  |-> PackageContents.xml
```



Deploy a LSP File with a Plug-in Bundle

Basic example of a *PackageContents.xml* file:

```
<?xml version="1.0" encoding="utf-8"?>
<ApplicationPackage
  SchemaVersion="1.0"
  AppVersion="1.0"
  Name="AU2020 AS468504-L"
  Description="AU2020 Example for AS468504-L."
  Author="HyperPics, LLC"
  ProductCode="{45F619FE-E286-4C4E-8134-B50E8DFC23E3}"
>
<CompanyDetails
   Name="HyperPics, LLC"
   Url="http://www.hyperpics.com"
/>
```

Deploy a LSP File with a Plug-in Bundle

```
 <Components Description="Windows and Mac OS operating systems">
   <RuntimeRequirements
      OS="Win32|Win64|Mac"
      SeriesMin="R19.0"
      Platform="AutoCAD*"
   />
    <ComponentEntry Description="Your custom file"
       AppName="AU2020Examples"
       Version="1.0"
       ModuleName="./au2020.lsp">
    </ComponentEntry>
  </Components>
</ApplicationPackage>
```

Deploy a LSP File with a Plug-in Bundle

Note: The `ProductCode` value (GUID) must be unique for each bundle.

http://www.guidgenerator.com/

Copy files and folders to one of these locations to deploy:

- Program Files and Applications folder
- All Users Profile folder
- User Profile folder

Deploy a LSP File with a Plug-in Bundle

Trusted and recommended locations:

- **Windows 7 and later**
  - %PROGRAMFILES%\Autodesk\ApplicationPlugins
  - %PROGRAMFILES(x86)%\Autodesk\ApplicationPlugins
- **Mac OS X**
  - ~/Applications/Autodesk/ApplicationAddins

Deploy a LSP File with a Plug-in Bundle

Other supported locations, but they are not trusted by default:

- **Windows 7 and later**
  - %ALLUSERSPROFILE%\Autodesk\ApplicationPlugins
  - %APPDATA%\Autodesk\ApplicationPlugins
- **Mac OS X**
  - ~/Autodesk/ApplicationAddins

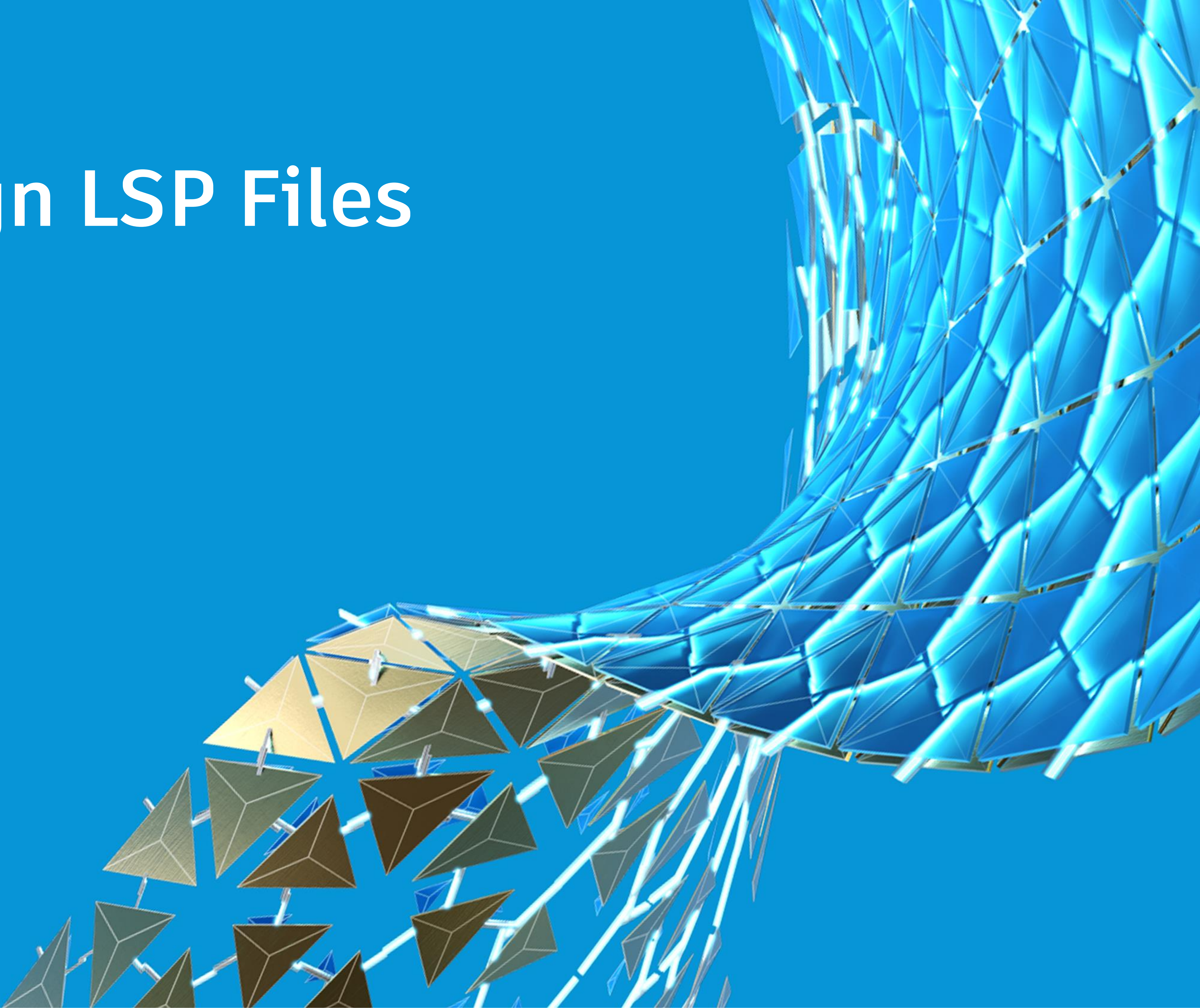
Exercise: E5 - Create a Basic Plug-in Bundle

In this exercise, you will:

- Create the folder structure for a plug-in bundle
- Update the *PackageContents.xml* file in a plug-in bundle
- Deploy a plug-in bundle

Follow along with the video or go to page 26 of the handouts.

Digitally Sign LSP Files



Digitally Sign LSP Files

Helps to protect you from potentially malicious code

Support was added in AutoCAD 2016

- AutoCAD warns when loading an LSP file that isn’t digitally signed
- Digital signatures are stored as comment blocks in an LSP file

LSP files are signed using the Attach Digital Signatures utility

To Digitally Sign an LSP File

1. Obtain a digital certificate from a certificate authority.

   or

   Create your own digital certificate, recommended for in-house development only.
2. Register your digital certificates on our workstation.
3. Start the Attach Digital Signatures utility.
4. Add the LSP files to be signed to the utility.
5. Sign the files and test the files in AutoCAD.

Exercise: E6 - Digitally Sign an LSP File

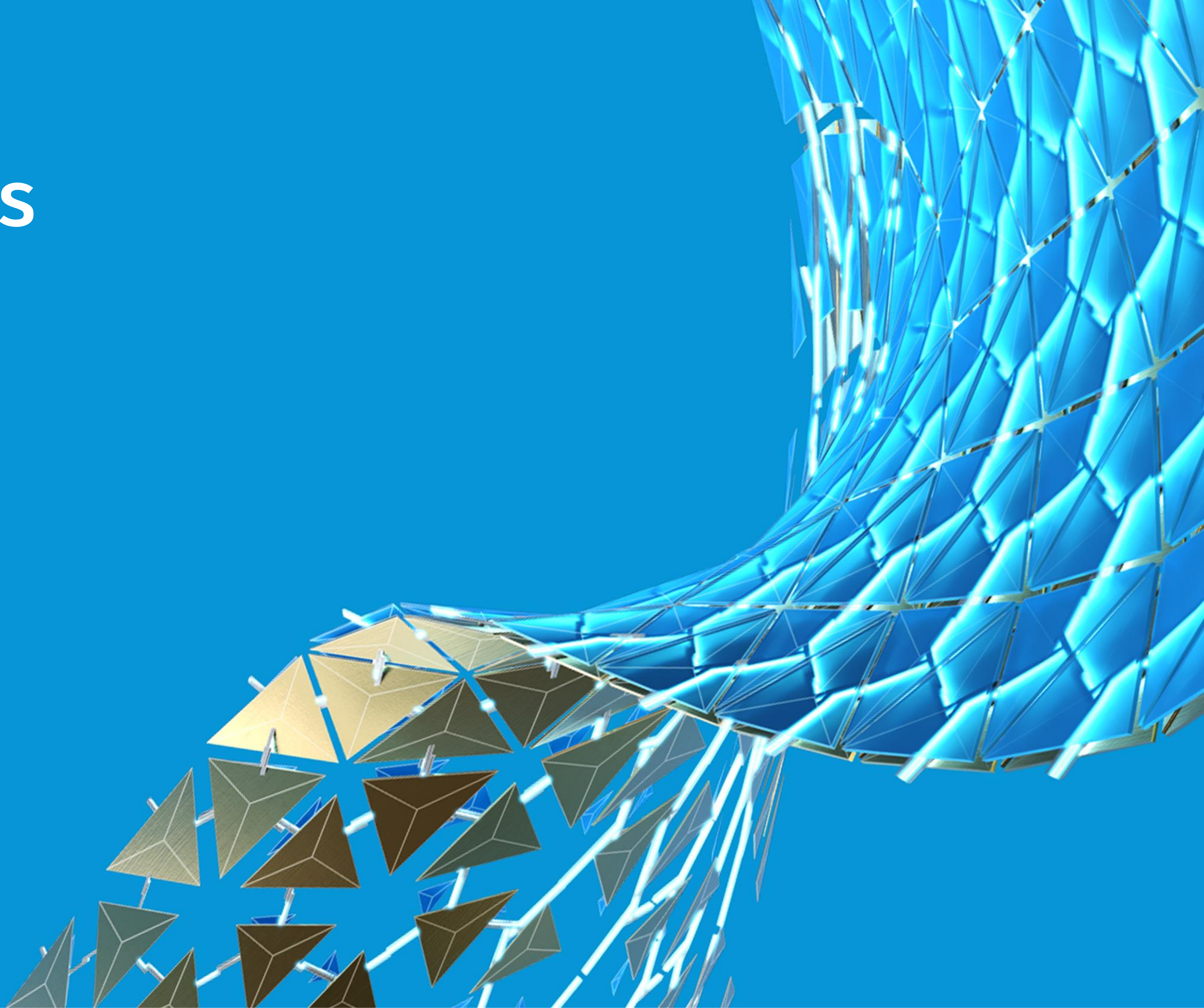
In this exercise, you will:

- Digitally sign an LSP file
- Load and verify a digitally signed LSP file

In the Datasets folder, double-click *importDigitalSig.bat* and click Yes.

Follow along with the video or go to page 30 of the handouts.

User Profiles



User Profiles

Used to control application and user preferences:

- Search paths used to locate support files
- Trusted locations for custom program files
- Colors and fonts used by grips, application, and Command window
- Plot/publish, open and save file options
- And many other settings.

User Profiles

Created using the Options dialog box

Set current using the

- Profiles tab of the Options dialog box
- `/p` command line switch

```
"…\acad.exe" /p "<<Unnamed Profile>>"
```

To Create a User Profile

1. Display the Options dialog box.
2. Set the Profiles tab current.
3. Add a new profile and set it current.
4. Adjust the preferences and settings, as desired, in the Options dialog box.

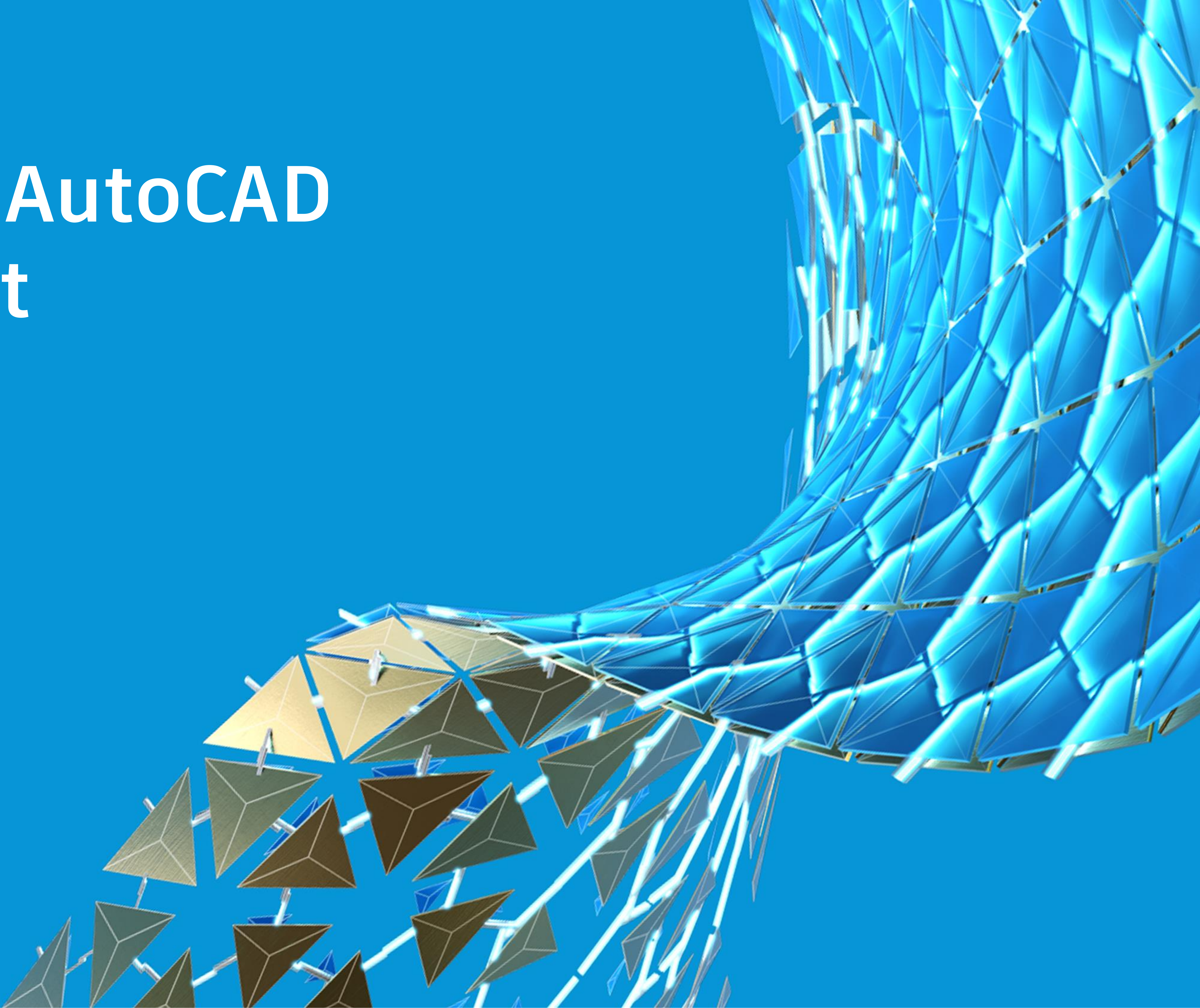
Exercise: E7 - Create and Modify a New Profile

In this exercise, you will:

- Create a new user profile
- Change the settings associated with a user profile
- Set a user profile current

Follow along with the video or go to page 34 of the handouts.

Restore the AutoCAD Environment

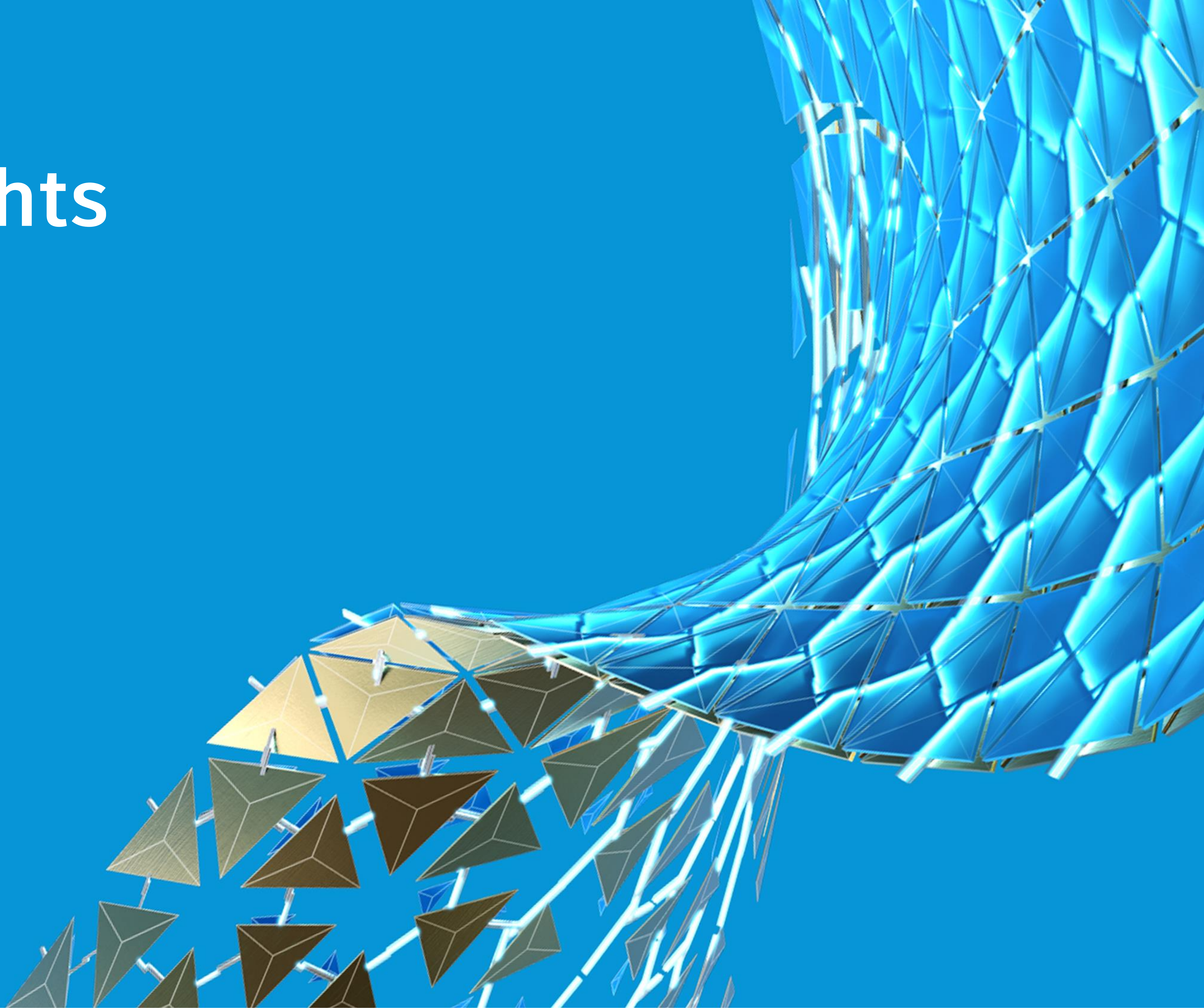


Restore the AutoCAD Environment

Follow the steps under “Restore the AutoCAD Environment”

- Remove the support file search path from the user profile
- Remove the My Profile user profile
- Remove the action macros you recorded
- Remove the AS468504-L plug-in bundle
- Remove the digital certificates to digital sign files

Final Thoughts



Final Thoughts

Customization and programming can:

- Enhance productivity
- Improve or introduce new workflows

Customizing has many similarities to Wonderland in *Lewis Caroll's Alice's Adventures*.

Both

- Are virtually endless
- Hold many mysteries just waiting to be discovered

Final Thoughts

Questions? Questions? Questions?

If you have any further questions, feel free to contact me via

email: lee.ambrosius@autodesk.com

twitter: @leeambrosius

Thanks for watching this session!