AS197432-L - AutoCAD Customization Boot Camp: Beyond the Basics

Lee Ambrosius

Principal Learning Experience Designer

AUTODESK® UNIVERSITY

Where Am I and Who Should Be Here

You are in session:

- AS197432-L - AutoCAD Customization Boot Camp: Beyond the Basics

You should know:

- AutoCAD 2019 (or AutoCAD 2009 and later)

You should want to:

- Automate tasks through scripting and programming
- Get the AutoCAD program to work for you

Who Am I?

My name is Lee Ambrosius:

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

My job in a nutshell:

- Document the past and present AutoCAD releases for the future

Who Are the Lab Assistants?

Lab assistants:

- Scott Hallmark
- Scott Wilcox
- Stephen Sluka

Their roles are to:

- Help out when you get stuck
- Ensure no one gets left behind

Session Rules

- Silence your mobile phone, tablet, and any other device
- If you have to leave, please do so quietly
- Hold all questions to the end
- If you get stuck, raise your hand

Enjoy the Journey

The path isn’t the same for everyone…

but the goal is often shared by many.

To the left and right of you is someone

with a shared interest, talk and learn from them.

Welcome to Specialist Training

What You Will Learn Today

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

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

What You Need to Get Started

For this session, you will be using:

- AutoCAD 2019
- Action Recorder
- Notepad; part of the Windows operating system

What is Going to be Covered

Handouts for this session are divided into two separate parts/files:

- **Exercises** – What we will be doing during this session
- **Supplemental** – Content for the flight back

Things We Need to Know Before Proceeding

Setting Up for the Lab

For this lab:

- If you see a single icon on your desktop labeled “AU 2018 Vegas – lab02 – desktop click it now”
- Open the handouts now if you don’t have them open already:

  C:\DATASETS\AS197432-L - AutoCAD Customization Boot Camp Beyond the Basics\AS197432-L-Ambrosius-AU2018.pdf
- AutoCAD can be found in the Software folder on the Desktop
- Recommend snapping the handouts and AutoCAD application to opposites sides of the Windows Desktop

Script Files

Script Files

What is a script file?

- An ASCII text file with the SCR extension
- Sequence of commands and system variables to be executed in a linear order
- Can include AutoLISP statements

Script Files

Why create or use scripts?

- Low learning curve
- No special editor or programming skills required
- Execute many commands rapidly without user input
- Work across multiple releases and toolsets
- Transparent execution is supported

Script Files

Known Limitations:

- User can’t be prompted for input
- Dialog boxes can’t be displayed
- In AutoCAD 2015 and earlier, only one script can be executed at a time
- Commands are executed as if the FILEDIA and CMDDIA system variables are set 0

Script Files

Example of input entered at the Command prompt:

```
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
Command: GRIDDISPLAY
Enter new value for GRIDDISPLAY <3>: 2
```

Script Files

Examples of scripts with the same input:

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

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

Script Files

Formatting of a script file:

- Commands and options can be lower or uppercase; string values are case sensitive
- A space or new line is equivalent to pressing Enter
- Text values with spaces must be surrounded with double quotation marks; in most cases
- A blank line must always be placed at the end of the file

Script Files

Formatting of a script file (cont.):

- A period in front of a command name ensures the execution of the natively defined command

  `.LINE`

- An underscore in front of a command name forces the use of a global command or option name; global command and option names are always the English name

  `_CIRCLE` or **`_.CIRCLE`**

- A semi-colon denotes a comment in a script, text to the right of a semi-colon isn’t executed

  ```
  ; Created on: 11/5/18
  ```

Script Files

Techniques to running a script file:

- SCRIPT command
- Drag and drop (Windows only)
- /b (Windows) or –b (Mac OS X) command line switch
- ScriptPro (Windows only) - http://autode.sk/2fS0Rml

Script Files

Commands related to script files:

- **DELAY** – Pauses the execution of a script for a specified duration in milliseconds
- **RESUME** – Resumes the execution of a script that was paused by pressing the Backspace key
- **RSCRIPT** – Repeats the previous executed script in the current AutoCAD session
- **SCRIPT** – Runs a SCR file
- **SCRIPTCALL** – Used to run a nested script file; AutoCAD 2016 and later

To Create a Script File

1. At the Command prompt, walkthrough the commands and options to be executed by the script.
2. Create the script (SCR) file with Notepad.
3. Add the commands and options to the SCR file to be executed.
4. Save the SCR file.
5. Create or open a drawing file.
6. Run the SCR file and validate the results.

Exercise: E1 - Create and Run a Script

In this exercise, you will:

- Create a new SCR file that performs some basic drawing setup tasks
- Run a SCR file with the SCRIPT command

Starts on page 4 of the handouts.

Action Macros

Action Macros

Smallest interaction that can be recorded

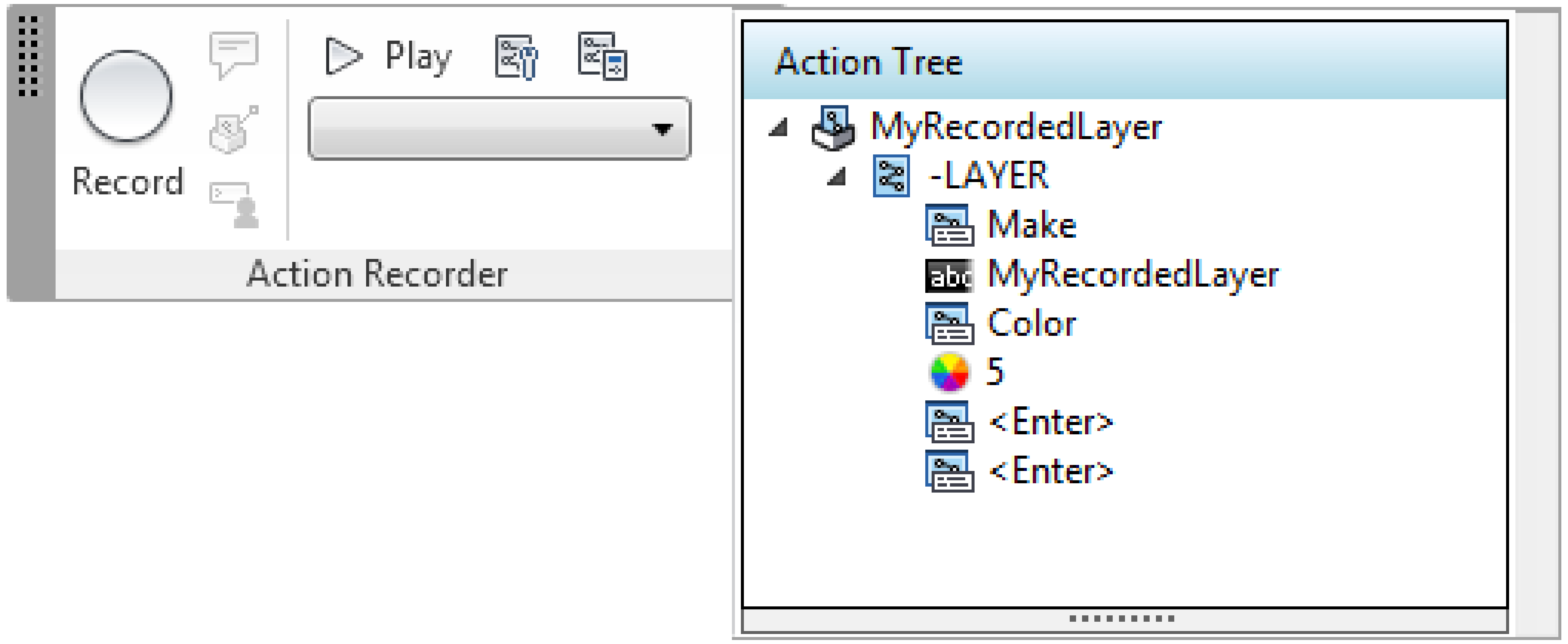
An action can be:

- Starting of a command
- Specified user input; coordinate, object selection, or other values
- Interactions performed with/on the
    - Properties, Quick Properties, Tool Palettes, and Layer Properties Manager palettes
    - Quick Access toolbar, ribbon, and status bar

Action Macros

Actions are recorded with the Action Recorder on the ribbon

Saved to action macro (ACTM) files



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

Action Macros

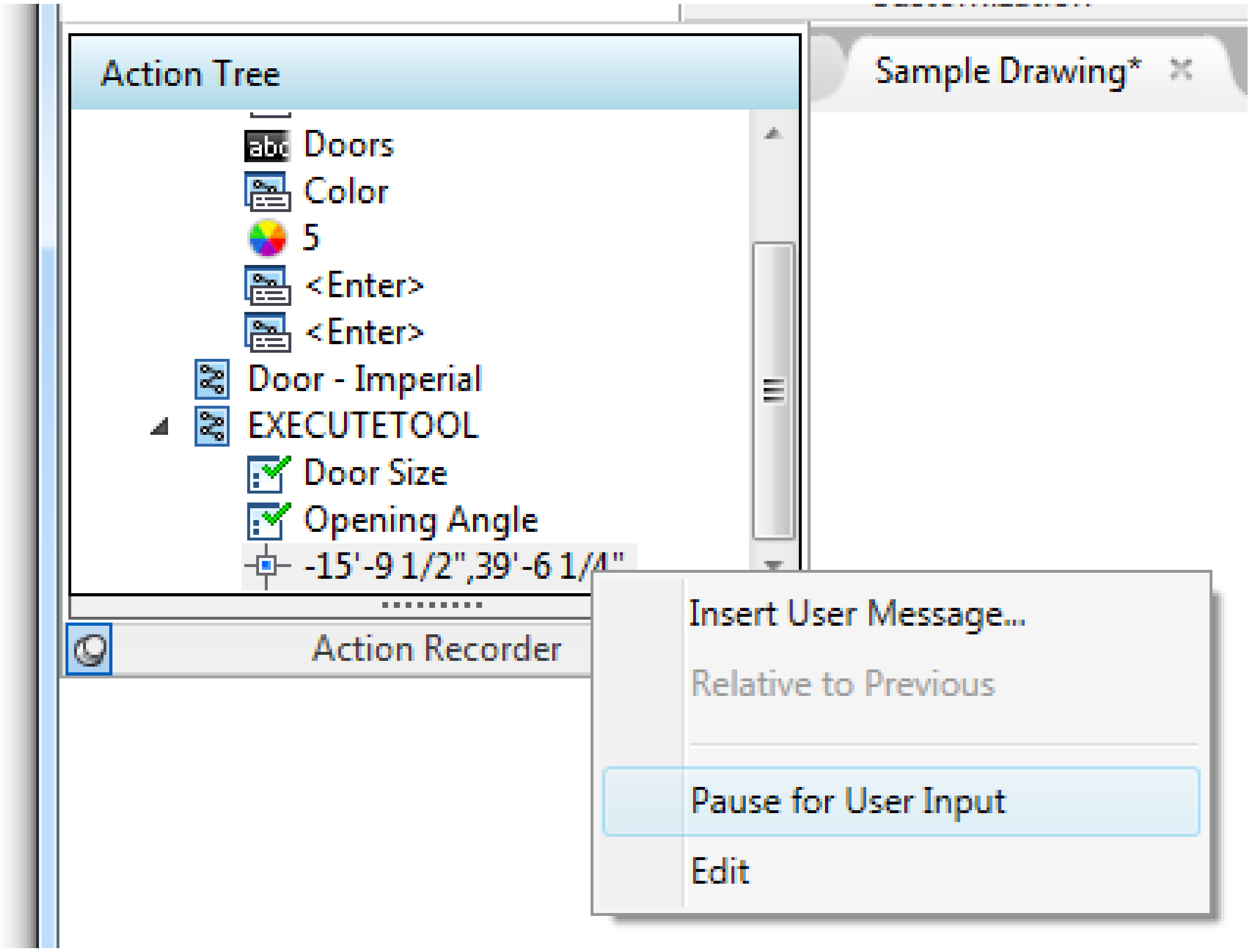
Things to know before recording begins:

- Recommended to avoid commands that display dialog boxes
- System variable values can be changed while recording

Action Macros

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

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 commands that are available to all users; commands specific to AutoCAD Architecture can’t be executed in AutoCAD

To Record an Action Macro

1. Start recording with 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: E2.A - Record and Playback an Action Macro

In this exercise, you will:

- Record the actions performed at the Command prompt that create a new layer and rectangular revision cloud
- Save and modify an action macro
- Playback a recorded action macro

Starts on page 8 of the handouts.

AutoLISP

AutoLISP

Programming language

- based on the LISP (**LIS**t **P**rocessing) programming language
- specific to AutoCAD and AutoCAD-based programs
- has been around for a very long time; 30+ years (January 1986), introduced in AutoCAD Version 2.18
- doesn't require the use of a special editor
- doesn't need to be compiled; interpreted language

AutoLISP

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 2018!")
```

AutoLISP Syntax

Syntax of an AutoLISP expression:

```
(function_name argumentX)
```

- **function_name** – Name of the function to execute
- ***argumentX*** – Value(s) the function should do something with
- Not all functions except arguments, while some functions expect multiple arguments

AutoLISP Functions

There are 5 functions you should know when getting started:

- **`command`** – Executes an AutoCAD command
- **`setq`** – Creates a user-defined variable and assigns it a value
- **`defun`** – Creates a user-defined function that can be executed at the Command prompt
- **`setvar`** – Assigns a value to a system variable
- **`getvar`** – Gets the current value of a system variable

Use Commands

`command` Function

Executes a command

Syntax:

```
(command command_name valueX)
```

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

`command` Function

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]:
```

Input converted to an AutoLISP statement with the `command` function:

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

`command` Function

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
```

Input converted to an AutoLISP statement with the `command` function:

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

`command` Function

Special values used with the `command` function:

- **"" (pair of double quotations)** – Represents a press of the Enter key
- **PAUSE** – Instructs AutoCAD to wait for the user to provide a value, such as a point or object selection

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

Store and Work with Data Values

`setq` Function

Creates a user-defined variable and assigns it a value

Syntax:

```
(setq variable_name value)
```

- **`variable_name`** – Name of the user-defined variable to define or update
- ***`value`*** – Value to be assigned to the user-defined variable

`setq` Function

Examples:

Assigns the numeric value of 1.25 to the variable named *`dRadius`*

```
(setq dRadius 1.25)
```

Assigns the text AU 2018 to the variable named *`strEvent`*

```
(setq strEvent "AU 2018")
```

`setq` Function

Prefix a variable name with an ! (Exclamation point) at the AutoCAD Command prompt to return its value

Example:

```
Command: (setq dRadius 1.25)
Command: !dRadius
1.25
```

setvar/getvar Function

Set or get the value of a system variable

Syntax:

```
(setvar sysvar_name value)
(getvar sysvar_name)
```

- sysvar_name – Name of the system variable to work with
- *value* – Value to be assigned to the system variable

setvar/getvar Function

Examples:

Gets the value of the OSMODE system variable

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

Sets the value of the OSMODE system variable to INT and END

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

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 2018!"`

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: E3 - 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

Starts on page 18 of the handouts.

Define Custom Functions

Define Custom Functions

AutoLISP can be used to create reusable custom functions

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:

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

- `function_name` – Name of the function to be defined
  - Optional, `c:` indicates the function name can be typed at the Command prompt
- *`expressionX`* – Expressions to execute

`defun` Function

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: E4 - Create Simple Custom Functions

In this exercise, you will:

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

Starts on page 20 of the handouts.

Store AutoLISP Expressions

Store AutoLISP Expressions

Expressions can be stored in a file on disk for re-use

- ASCII text file with a *.lsp* extension
- LSP files can be created/modified with Notepad or the Visual LISP Editor
- Comments can be added to a LSP file
- A LSP file must be loaded into each drawing in which it will be used

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: 11/5/18 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: E5 - 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

Starts on page 22 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 that is described by an XML file named *PackageContents.xml*

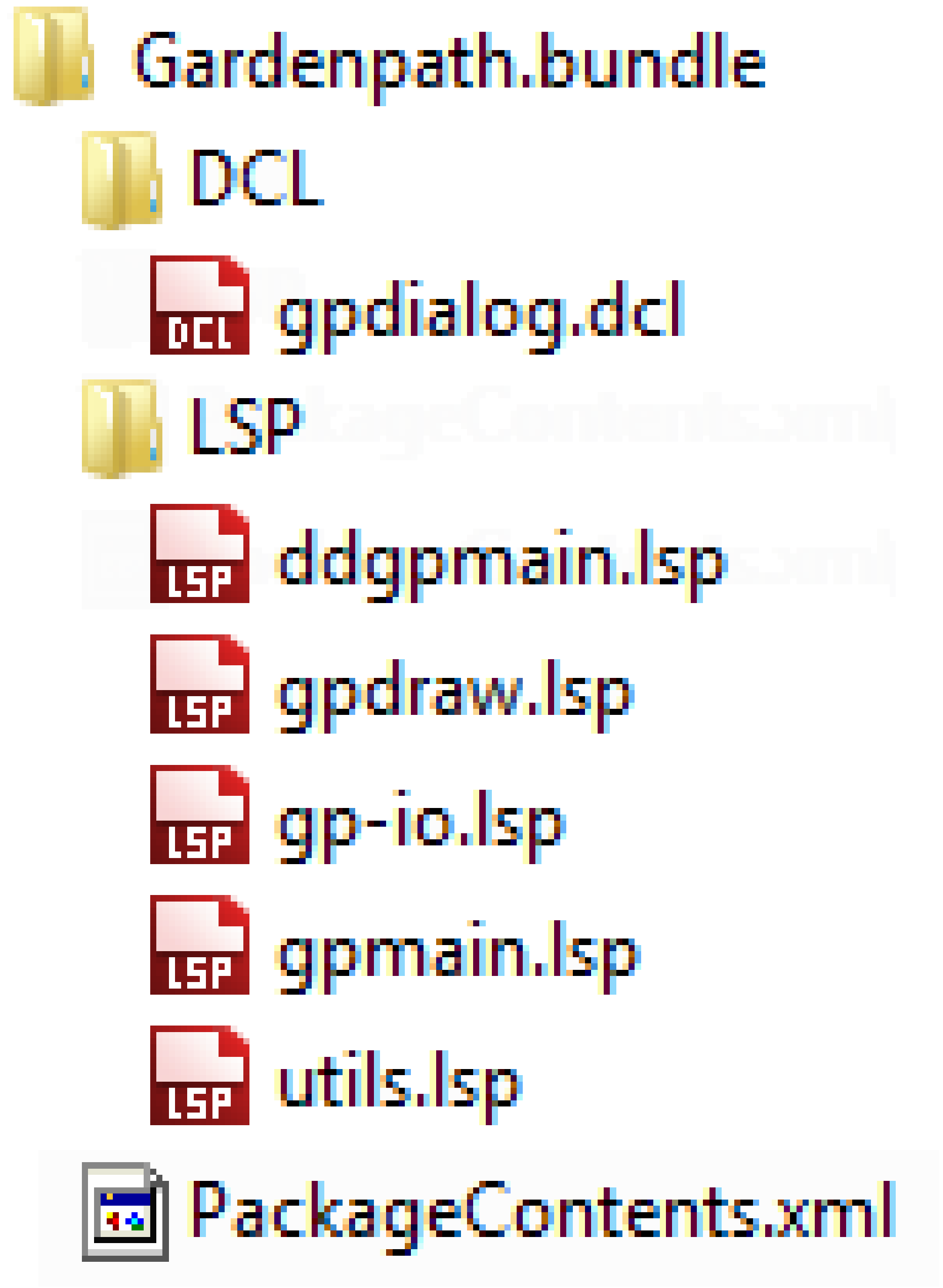
PackageContents.xml is

- Placed in the root folder of a plug-in bundle
- Describes the files in the plug-in bundle and defines how they should be loaded

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
```



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="AU2018 AS197432-L"
  Description="AU2018 Example for AS197432-L."
  Author="HyperPics, LLC"
  ProductCode="{45F619FE-E286-4C4E-8134-B50E8DFC23E3}"
>
```

Deploy a LSP File with a Plug-in Bundle

```
<CompanyDetails
   Name="HyperPics, LLC"
   Url="http://www.hyperpics.com"
/>
<Components Description="Windows and Mac OS operating systems">
   <RuntimeRequirements
      OS="Win32|Win64|Mac"
      SeriesMin="R19.0"
      Platform="AutoCAD*"
   />
```

Deploy a LSP File with a Plug-in Bundle

```
    <ComponentEntry Description="Your custom file"
       AppName="AU2018Examples"
       Version="1.0"
       ModuleName="./au2018.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/

A plug-in bundle is deployed by copying all the files and folders to one of these locations

- 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: E6 - 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

Starts on page 26 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 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

Starts on page 29 of the handouts.

Final Thoughts and Questions

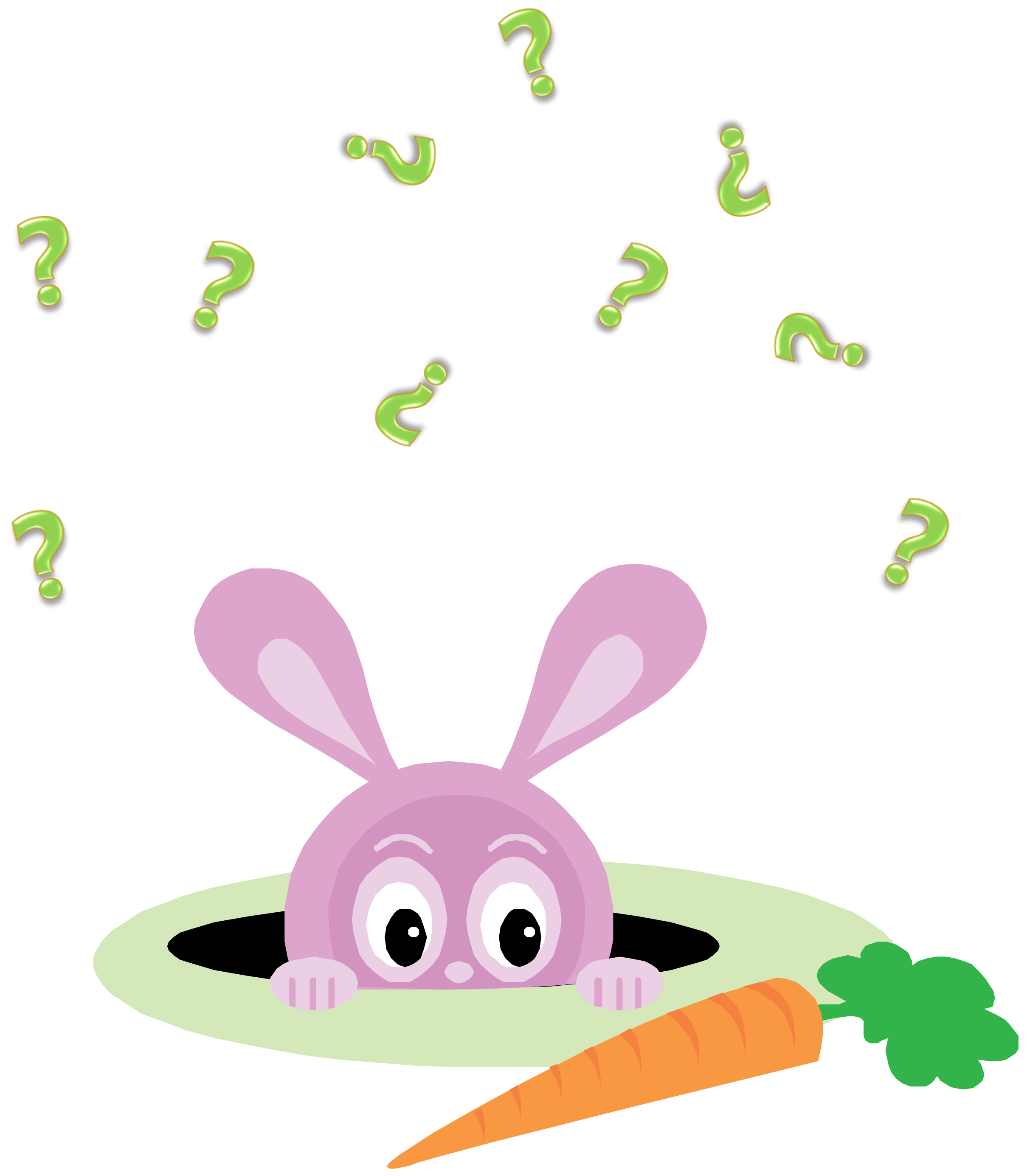
Final Thoughts and Questions

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 waiting to be discovered



Closing Remarks

Thanks for choosing this session.

Don’t forget to complete this session’s survey.

If you have any further questions, contact me via:

email:	lee.ambrosius@autodesk.com
twitter:	@leeambrosius

AUTODESK®

Make anything™