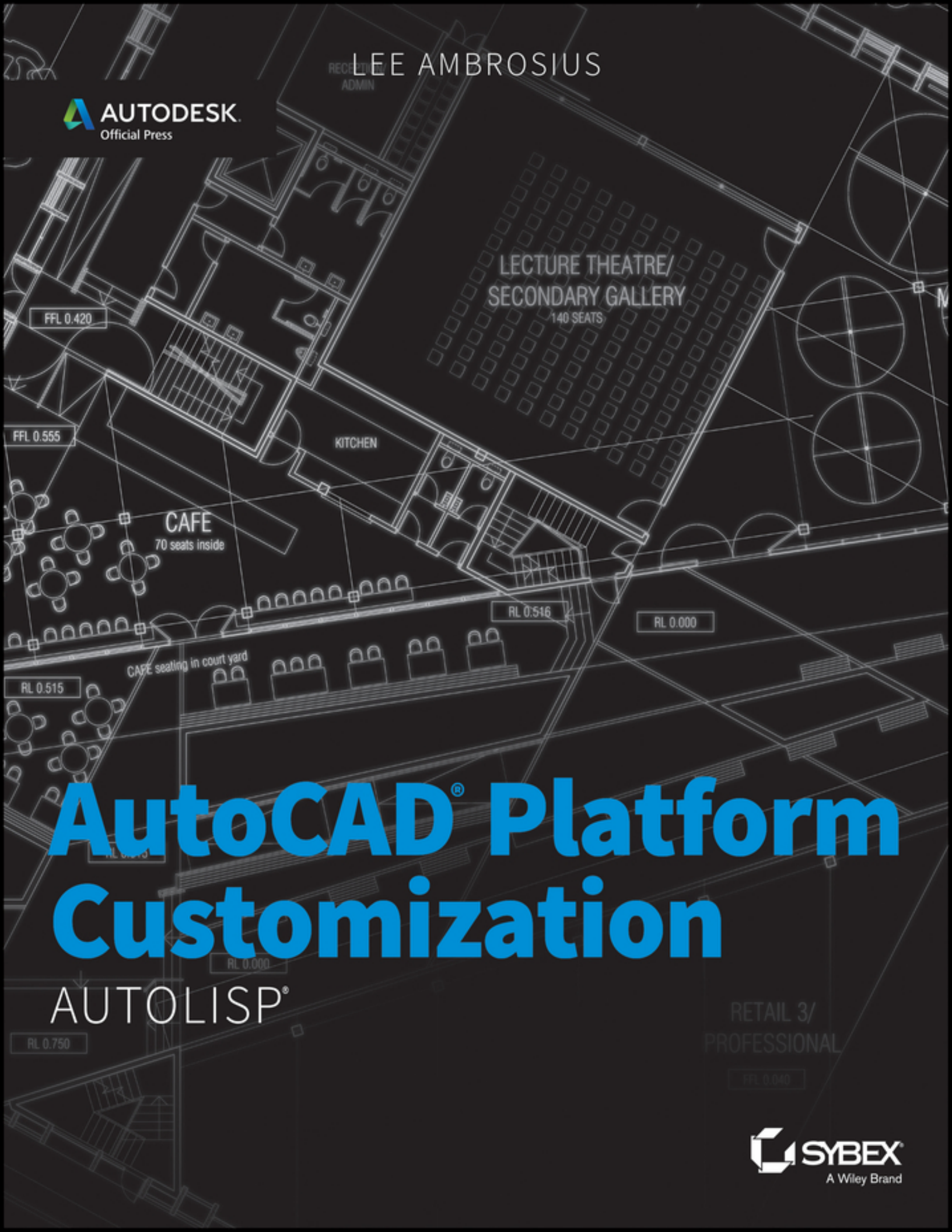
LEE AMBROSIUS

AUTODESK
Official Press



AutoCAD® Platform Customization

AUTOLISP®

SYBEX
A Wiley Brand

Table of Contents

List of Illustrations

List of Tables

AutoCAD® Platform Customization AutoLISP®

Lee Ambrosius

Senior Acquisitions Editor: Willem Knibbe

Development Editor: Mary Ellen Schutz

Technical Editor: Craig Black

Production Editor: Dassi Zeidel

Copy Editor: Liz Welch

Editorial Manager: Pete Gaughan

Vice President and Executive Group Publisher: Richard Swadley

Associate Publisher: Chris Webb

Book Designers: Maureen Forys, Happenstance Type-O-Rama; Judy Fung

Proofreader: Candace Cunningham

Indexer: Ted Laux

Project Coordinator, Cover: Patrick Redmond

Cover Designer: Wiley

Cover Image: © Smileyjoanne/iStockphoto.com

Published simultaneously in Canada

ISBN: 978-1-118-90055-0 (ebk.)

ISBN: 978-1-118-90696-5 (ebk.)

herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

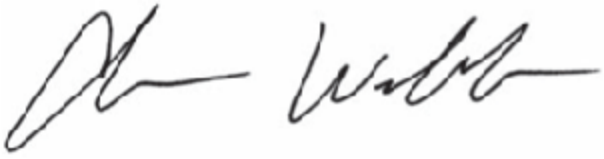
Dear Reader,

Thank you for choosing *AutoCAD Platform Customization: AutoLISP.* This book is part of a family of premium-quality Sybex books, all of which are written by outstanding authors who combine practical experience with a gift for teaching.

Sybex was founded in 1976. More than 30 years later, we're still committed to producing consistently exceptional books. With each of our titles, we're working hard to set a new standard for the industry. From the paper we print on, to the authors we work with, our goal is to bring you the best books available.

I hope you see all that reflected in these pages. I'd be very interested to hear your comments and get your feedback on how we're doing. Feel free to let me know what you think about this or any other Sybex book by sending me an email at contactus@sybex.com. If you think you've found a technical error in this book, please visit http://sybex.custhelp.com. Customer feedback is critical to our efforts at Sybex.

Best regards,



Chris Webb
Associate Publisher
Sybex, an Imprint of Wiley

To my wife, who is also my best friend: It is hard to imagine that I would be writing this book if it were not for you. It was you, all those years ago, who encouraged me to step outside of my comfort zone and share what I knew with others. Thank you for the push I needed and for coming along on this journey with me.

Acknowledgments

I have to give a very special thanks to all the great folks at Sybex for working on and helping to get this project off the ground after a few years of talking about it, especially Willem Knibbe. The next two people I would like to thank are Mary Ellen Schutz and Dassi Zeidel, the development and production editors on this book; you two made sure I stayed on track and delivered a high-quality book. I also want to thank Liz Welch (copyeditor), Candace Cunningham (proofreader), and Ted Laux (indexer) for the work you all did on this book.

Thanks to all the folks at Autodesk, who put in the long hours and are dedicated to the work they do on the Autodesk® AutoCAD® product. Last but not least, to a great friend of mine, Craig Black. I have known Craig for nearly 20 years. I first met him while attending an AutoLISP® session at the local Autodesk Training Center, where he was an instructor. Craig is a great AutoLISP programmer and it was a joy to have him as the technical editor on this book. Being a technical editor is never the easiest job, but it is one of the most important and I appreciate what you have done to make this book better.

About the Author

Lee Ambrosius first started working with AutoCAD R12 for DOS in 1994. As a drafter, he quickly discovered that every project included lots of repetition. Lee, not being one to settle for “this is just the way things are,” set out on a path that would redefine his career. This new path would lead him into the wondrous world of customization and programming—which you might catch him referring to as “the rabbit hole.”

In 1996, Lee began learning the core concepts of customizing the AutoCAD user interface and AutoLISP. The introduction of VBA in AutoCAD R14 would once again redefine how Lee approached programming solutions for AutoCAD. VBA made it much easier to communicate with external databases and other applications that supported VBA, and transformed the way information could be moved between project-management and manufacturing systems.

Not being content with VBA, in 1999 Lee attended his first Autodesk University and began to learn ObjectARX®. Autodesk University made a lasting impression on him. In 2001, he started helping as a lab assistant. He began presenting on customizing and programming AutoCAD at the event in 2004. Along the way he learned how to use the AutoCAD Managed .NET API.

In 2005, Lee decided cubicle life was no longer for him, so he ventured off into the CAD industry as an independent consultant and programmer for his own company, named HyperPics, LLC. After he spent two years as a consultant, Autodesk invited him to work on the AutoCAD team; he has been on the AutoCAD team since 2007. For most of his career at Autodesk, Lee has worked primarily on the

customization and end-user documentation. Recently, he has been working on the AutoLISP, VBA, ObjectARX, .NET, and JavaScript programming documentation.

In addition to working on the AutoCAD documentation, Lee has been involved with the *AutoCAD and AutoCAD LT Bible*, *AutoCAD for Dummies*, *AutoCAD & AutoCAD LT All-in-One Desk Reference for Dummies*, *AutoCAD 3D Modeling Workbook for Dummies*, and *Mastering AutoCAD for Mac* books on different editions as a technical writer or author. He has also written white papers on customization for Autodesk and a variety of articles for *AUGIWorld*, published by AUGI®, on customization and programming.

Introduction

Welcome to *AutoCAD Platform Customization: AutoLISP*! Have you ever thought to yourself, Why doesn't the Autodesk® AutoCAD® program include every feature I need? Why isn't it streamlined for the type of work I perform? If so, you are not alone. AutoCAD at its core is a drafting platform that can be shaped and molded to more efficiently complete the tasks you perform on a daily basis and to enhance your company's workflows with the use of programming. Take a deep breath. I did just mention programming, but programming isn't something to fear. At first, just the idea of programming makes many people want to run in the opposite direction—myself included. The productivity gains are what propelled me forward. Programming isn't all that different from anything else you've tried doing for the first time.

In many ways, learning to program is much like learning a foreign language. For many new to AutoLISP®, the starting place is often a basic understanding of syntax and the `command` function. The `command` function allows you to leverage your knowledge of AutoCAD commands by being able to pass specific values to a command or pausing a command for a value. After you are comfortable with the syntax of AutoLISP and the `command` function, you can begin to learn additional functions that allow for the development of more robust and complex programs.

About This Book

AutoCAD Platform Customization: AutoLISP provides you with an understanding of the AutoLISP programming language and how it can help improve your productivity.

This book is designed to be more than just an introduction to AutoLISP—a resource that can be used time and again when developing AutoLISP programs. As you page through this book, you will notice that it contains sample code and exercises that are based on real-world solutions.

This book is the second in a series of three that focuses on customizing and programming AutoCAD. The three-book series as a whole is known as *AutoCAD Platform Customization: User Interface, AutoLISP, VBA, and Beyond*, which will be available as a printed book in late 2014/early 2015. Book 1 in the series, *AutoCAD Platform Customization: User Interface and Beyond*, was published in early 2014 and focused on CAD standards and general customization of AutoCAD; book 3, *AutoCAD Platform Customization: VBA*, will be available in fall/winter 2014 and covers the VBA programming platform inside AutoCAD.

Is This Book for You?

AutoCAD Platform Customization: AutoLISP covers many aspects of AutoLISP programming for AutoCAD on Windows and Mac OS. If any of the following are true, this book will be useful to you:

- You want to develop and load custom programs using the AutoLISP programming language for use in the AutoCAD drawing environment.
- You want to automate the creation and manipulation of drawing objects.
- You want to automate repetitive tasks.
- You want to help manage and enforce CAD standards for your company.

AutoLISP in AutoCAD

AutoLISP is the most popular and is the original supported programming language for the AutoCAD program. The reason for its popularity with new (and even veteran) programmers is that it is a natural extension of the AutoCAD program. There is no additional software to purchase, and AutoLISP can leverage the commands that Autodesk and third-party developers expose at the Command prompt. For example, with a few simple lines of code you can set a layer as current and insert a title block with a specific insertion point, scale, and rotation. The block is then inserted on the layer you specified. To perform the same tasks manually, the end user would have to set a layer as current, choose the block they want to insert, and specify the properties of the block, which, in the case of a title block, are almost always the same.

The AutoLISP programming language can be used to accomplish the following:

- Create custom functions that can be executed from the AutoCAD Command prompt
- Create and manipulate graphical objects in a drawing, such as lines, circles, and arcs
- Create and manipulate nongraphical objects in a drawing, such as layers, dimension styles, and named views
- Perform mathematical and geometric calculations
- Request input from or display messages to the user at the Command prompt
- Interact with files and directories in the operating system
- Read from and write to external files
- Connect to applications that support ActiveX and COM
- Display dialog boxes and get input from the end user

AutoLISP code can be entered directly at the Command prompt or loaded using a LSP file. Once an AutoLISP program has been loaded, you can execute the custom functions from the Command prompt. Functions executed from the Command prompt can be similar to standard AutoCAD commands, but the programmer determines the prompts that should be displayed. It is also possible to use AutoLISP code with a command macro that is activated from the AutoCAD user interface or a tool on a tool palette.

What to Expect

This book is organized to help you learn AutoLISP fundamentals and how to manage and implement custom AutoLISP programs. Additional resources and files containing the example code found throughout this book can be found on the companion web page, www.sybex.com/go/autocadcustomization.

Chapter 1: Quick Start for New AutoLISP Programmers In this chapter, you'll get an introduction to the AutoLISP programming language. I begin by showing you how to enter AutoLISP expressions at the Command prompt and execute standard AutoCAD commands. After that, you are eased into some basic programming concepts that allow you to perform conditional tests and repeat expressions. The chapter wraps up with creating and loading an AutoLISP file into the AutoCAD program.

Chapter 2: Understanding AutoLISP In this chapter, you'll learn the fundamentals of the AutoLISP programming language. AutoLISP fundamentals include a look at the syntax and structure of an expression, how to use a function, and how to work with variables. Beyond just syntax and variables, you learn to use AutoCAD commands and group multiple AutoLISP expressions into custom functions.

Chapter 3: Calculating and Working with Values In this chapter, you'll learn to work with mathematical and string-manipulation functions. Math functions allow you to perform basic and advanced calculations based on object values or a value that the user might provide, whereas string-manipulation functions allow you to work with text-based values. Both numeric and textual values are used when creating or manipulating objects, adding annotations to a drawing, or displaying a message to the end user. Based on how the values are used, numeric

values can be converted to strings and strings can be converted to numeric values.

Chapter 4: Working with Lists In this chapter, you'll learn to work with the list data type. Lists are used throughout AutoLISP to provide 2D or 3D coordinate values and to define an object stored in a drawing.

Chapter 5: Requesting Input and Using Conditional and Looping Expressions In this chapter, you'll learn to request input from the user, use conditional statements, and repeat expressions. Requesting input allows you to get values from the user and then use those values to determine the end result of the program. Conditional statements enable a program to make choices based on known conditions in a drawing or input from a user. After you understand conditional statements, you will learn to use them in conjunction with looping expressions to execute a set of expressions until a condition is met.

Chapter 6: Creating and Modifying Graphical Objects In this chapter, you'll learn how to create, modify, and attach extended data to graphical objects using AutoCAD commands and AutoLISP functions. Graphical objects represent the drawing objects, such as a line, an arc, or a circle, that are displayed in model space or on a named layout. When modifying objects, you can choose to step through all the objects in a drawing or let the user select the objects to be modified. Extended data allows you to store information with an object that can be used to identify the objects your program creates or link objects to external database records.

Chapter 7: Creating and Modifying Nongraphical Objects In this chapter, you'll learn how to create and modify nongraphical objects using AutoCAD commands

and AutoLISP functions. Nongraphical objects are used to control the appearance of graphical objects and store settings that affect the behavior of features in the AutoCAD program. Drawings support two different types of nongraphical objects: symbol-table objects and dictionaries.

Chapter 8: Working with the Operating System and External Files In this chapter, you will learn how to work with settings and files stored outside of the AutoCAD program. Settings can be stored in the Windows Registry and Plist files on Mac OS, and they can be used to affect the behavior of the AutoCAD program or persist values for your custom programs between AutoCAD sessions. Files and folders stored in the operating system can be accessed and manipulated from the AutoCAD program, which allows you to set up project folders or populate project information in the title block of a drawing from an external file.

Chapter 9: Catching and Handling Errors In this chapter, you will learn how to catch and handle errors that are caused by an AutoLISP function and keep an AutoLISP program from terminating early. AutoLISP provides functions that allow you to trace a function, see arguments as they are passed, catch an error and determine how it should be handled, and group functions together so all the actions performed can be rolled back as a single operation.

Chapter 10: Authoring, Managing, and Loading AutoLISP Programs In this chapter, you will learn how to store AutoLISP code statements in a file, load and manage AutoLISP files, and deploy custom programs with plug-in bundles. Storing AutoLISP code in a file allows for its reuse in multiple drawings. When you load an AutoLISP file, all of the functions defined in the file

are made available while the drawing remains open. Based on how you load or deploy an AutoLISP file, you might need to let the AutoCAD program know where your AutoLISP files are stored.

Chapter 11: Using the Visual LISP Editor (Windows only In this chapter, you will learn how to use the Visual LISP® Editor. The editor provides tools for writing, formatting, validating, and debugging code in an AutoLISP file. Using the Visual LISP Editor, you can group AutoLISP files into project files, which make them easy to manage and compile. Compiling an AutoLISP file secures the source code contained in the file so that it can't be altered by others.

Chapter 12: Working with ActiveX/COM Libraries (Windows only In this chapter, you will learn how to use ActiveX/COM libraries with AutoLISP. ActiveX provides access to additional functions, which allow for the creation and manipulation of drawing objects and AutoCAD application settings that aren't easily accessible with standard AutoLISP functions. External applications, such as Microsoft Word and Excel, can also be accessed from the AutoCAD program when using ActiveX.

Chapter 13: Implementing Dialog Boxes (Windows only In this chapter, you will learn how to create and use dialog boxes with an AutoLISP program. Dialog boxes provide an alternative method of requesting input from the user and are implemented using Dialog Control Language (DCL).

Companion Website

An online counterpart to this book, the companion web page contains the sample files required to complete the exercises found in this book in addition to the sample code

and project files used to demonstrate some of the programming concepts explained in this book. In addition to the sample files and code, the web page contains resources that are not mentioned in this book. The companion web page can be found at www.sybex.com/go/autocadcustomization.

Other Information

This book assumes that you know the basics of your operating system—Windows or Mac OS X—and AutoCAD 2009 or later. When appropriate, I indicate when a feature does not apply to a specific operating system or release of AutoCAD. Most of the images in this book were taken using AutoCAD 2014 in Windows 8 and AutoCAD 2014 in Mac OS X 10.7.

Since AutoCAD LT® doesn't support AutoLISP, none of the content in this book applies to that software package.

Styles and Conventions of This Book

This book uses a number of styles and character formats—bold, italic, monotype face, all uppercase or lowercase letters, and others—to help you distinguish from the text you read, sample code you can try, text that you need to enter at the AutoCAD Command prompt, or the name of an object class or method in one of the programming languages.

As you read through this book, keep the following conventions in mind:

- User-interface selections are represented by one of the following methods:

  Click the Application button ➢ Options.

On the ribbon, click the Manage tab ➤ Customization ➤ User Interface.

On the menu bar, click Tools ➤ Customize ➤ Interface.

In the drawing window, right-click and click Options.

- Keyboard input is shown in bold (for example, type `cui` and press Enter).
- Prompts that are displayed at the AutoCAD Command prompt are displayed as monospace font (for example, `Specify a start point:`).
- AutoCAD command and AutoLISP function names are displayed in all lowercase letters with a monospace font (for example, `line` or `command`).
- Example code and code statements that appear within a paragraph are displayed in monospace font. Code samples might look like one of the following:

```
(command "._circle" PAUSE 3)
```

The `alert` method can be used to display an error message to the user.

```
; Draw a rectangle
```

Contacting the Author

I hope that you enjoy *AutoCAD Platform Customization: AutoLISP* and that it changes the way you think about completing your day-to-day work. If you have any feedback or ideas that could improve this book, you can contact me using the following address:

Lee Ambrosius:lee_ambrosius@hyperpics.com

On my blog and website you'll find additional articles on customization and samples that I have written over the

years. You'll find these resources here:

Beyond the UI:http://hyperpics.blogs.com

HyperPics:www.hyperpics.com

If you encounter any problems with this publication, please report them to the publisher. Visit the book's website, www.sybex.com/go/autocadcustomization, and click the Errata link to open a form and submit the problem you found.

Chapter 1
Quick Start for New AutoLISP Programmers

The AutoLISP® language and programming in general are two subjects that I have enjoyed for over 15 years now, but the same subjects make some people cringe and want to run in the opposite direction. I am not going to claim AutoLISP is easy to learn, but it can be learned by anyone, whether or not they have a programming background. When I first set out to learn AutoLISP, I didn't have any programming experience, but I wanted the benefits that AutoLISP could offer.

I understand if you have some hesitation at the thought of learning AutoLISP, but you don't need to feel that way—I will help you. This chapter will ease you into some core programming concepts and the AutoLISP programming language by exposing you to a variety of functions that are available.

To complete the exercises in this chapter and be able to create and edit LSP files, you must have the following:

For Windows users: Autodesk® AutoCAD® 2006 or later and the Notepad program

For Mac OS users: Autodesk® AutoCAD® 2011 or later and the TextEdit program