Introduction

Technology Overview

AppleScript is a scripting language that provides convenient control of applications and of many parts of the Mac OS. AppleScript, which uses an English-like syntax, is a rich, object-oriented language. You can use it to automate simple tasks or implement complex workflows that combine operations from applications and the Mac OS.

AppleScript uses a type of message called an Apple event for the underlying interprocess communication between scripts and applications. Most applications must be able to respond to Apple events because the Mac OS uses Apple events for certain basic actions, such as launching an application and providing a list of documents for it to open. A scriptable application is one that goes beyond the basics to make its data and operations available to AppleScript scripts or other applications.

Scriptable applications are popular because users can write scripts to combine the capabilities of multiple applications. Making your application scriptable can also provide benefits in areas such as rapid development and automated testing. Both Carbon and Cocoa applications can be made scriptable, and the Cocoa framework contains built-in support that minimizes the amount of code you have to write.

AppleScript provides limited support for displaying user-interface items from scripts. However, with AppleScript Studio, you can create Mac OS X applications that use AppleScript scripts to control complex user interfaces. That makes it a useful tool for system administrators or others who need to quickly automate a process while providing a full user interface. It can also be useful for rapid user-interface prototyping.

The Automator application, available starting in Mac OS X version 10.4, provides a graphical interface for combining individual actions to create an automated workflow. Each action performs a distinct operation, such as copying a file, cropping a photo, or sending an email message. Automator includes actions that work with the Finder, Mail, iTunes, and other Apple applications, and you can write actions for key features of your applications. Starting in Mac OS X version 10.5, you can also use workflows in your applications.

Starting in Mac OS X v10.5 (Leopard), the Scripting Bridge technology provides an automated process for creating an Objective-C interface to scriptable applications. This allows Cocoa applications or other Objective-C code to efficiently access features of scriptable applications, using native Objective-C syntax. Scripting languages such as Ruby and Python can also use the Scripting Bridge. (In addition, they have open source bridges—RubyOSA and py-appscript—to scriptable applications running in Leopard or Tiger.) For more information, see Getting Started With Scripting & Automation.

AppleScript 2.0, which ships starting in Mac OS X v10.5, includes full support for Unicode text, additional support for identifying and working with application objects in scripts, 64-bit support, and more accurate and useful error messages. Mac OS X v10.5 also provides additional scriptability in Apple technologies such as iChat and the Dock.

Start Here

To get a quick introduction to the basics of AppleScript:

- Read AppleScript Overview for a broad overview of AppleScript, Apple events, and related technologies, with links to other documentation.
- From the Finder Help menu, choose Mac Help. In the ensuing window, type “AppleScript” into the query text field and press Return, for introductory topics on working with AppleScript.

To keep up with the latest developments in AppleScript, including news, recent articles, and additional resources, visit the AppleScript website and the AppleScript topic page.

Choose a Learning Path

Whatever your goal with AppleScript, you'll want to read about the AppleScript language and learn how to use it to write scripts. If you're an application developer, you'll need additional information to write scriptable applications and to write Automator actions so that users can take full advantage of your applications in their Automator workflows.

Writing AppleScript Scripts

- See "Script Editor and AppleScript Scripts" in Scripting With AppleScript in AppleScript Overview for information on writing scripts and using the Script Editor application (which is located in `/Applications/AppleScript`).
- Read AppleScript Language Guide or one of several comprehensive third-party books to learn about the basic features of the language. A web search for "AppleScript" will turn up many related websites and documents.
- See Essential Sub-Routines for a large library of AppleScript routines for working with numbers, lists, sorting, text, and more.

Making Your Application Scriptable

You can support scripting in your application whether you develop in procedural C, C++, or Objective-C.

- **If you want to design a scriptable application**, read Technical Note TN2106, Scripting Interface Guidelines, for information on designing a scripting terminology for your application. To learn more about the model-view-controller design pattern, which is useful for scriptable applications, see Cocoa Design Patterns in Cocoa Fundamentals Guide. For design guidelines for a scriptable Cocoa application, see "Designing for Scriptability" in Cocoa Scripting Guide.
- **For both a conceptual and task-oriented introduction to Apple events**, the underlying communication mechanism for AppleScript and scriptable applications, read Apple Events Programming Guide. This information is more important if you are developing Carbon rather than Cocoa applications, but Cocoa developers may still want to skim it.
- **If you are developing in Objective-C**, see Glossary for detailed information on working with scripting in Cocoa applications. For examples of scriptable Cocoa applications, see the Xcode projects for the Sketch and TextEdit applications, in `<Xcode>/Examples/AppKit`.
- **If you are developing in procedural C or C++**, read Coding Your Object Model for Advanced Scriptability to learn more about how to implement the object model for your application. See MoreAppleEvents and MoreOSL for sample code that demonstrates key features of a scriptable application, and includes libraries of routines for working with Apple events. Read Apple Events Programming Guide to learn how you can more easily work with Apple event structures in your application.

Using Automator in Your Application Development

To learn how to work with Automator, see the learning paths in "Development with Automator" in Getting Started With Scripting & Automation.

Creating an AppleScript Studio Application

You can use AppleScript Studio to quickly create applications with complex user interfaces that adhere to the guidelines in Prioritizing Design Decisions.

- **To learn how to write AppleScript Studio applications**, read AppleScript Studio Programming Guide, which includes a number of tutorials. For a complete list of the available AppleScript Studio terminology, read AppleScript Studio Terminology Reference. For sample code, see the applications in `<Xcode>/Examples/AppleScript Studio`.

Next Steps

The AppleScript Reference Library includes the following high-level resource pages, which can be bookmarked for easy access:

- Guides

  Conceptual and how-to information for AppleScript.

- Reference

  Focused, detailed descriptions in reference format for AppleScript.

- Featured Articles

  Articles on working with specific topics in scripting and automation.

- Release Notes

  Late-breaking news and highlights of new or changed features in the latest release.

- Sample Code

  Provides additional examples for writing code that works with Apple events and scripting.

- Technical Notes

  Late-breaking documents on timely technology issues.

- Technical Q&As

  Programming tips, code snippets, & FAQs by Apple's support engineers.

- Mailing Lists

  AppleScript mail lists include applescript-implementors for issues related to scriptable applications, applescript-users for issues related to writing AppleScript scripts, and applescript-studio for issues related to AppleScript Studio.

  Automator mail lists include automator-dev for developers creating new Automator actions and automator-users for users developing workflows and solutions using Automator.