

# Introduction to AppleScript Language Guide

Published: 2016-01-25 · Archived: 2026-04-05 15:49:27 UTC

This document is a guide to the AppleScript language—its lexical conventions, syntax, keywords, and other elements. It is intended primarily for use with AppleScript 2.0 or later and macOS version 10.5 or later.

AppleScript 2.0 can use scripts developed for any version of AppleScript from 1.1 through 1.10.7, any scripting addition created for AppleScript 1.5 or later for macOS, and any scriptable application for Mac OS v7.1 or later. A script created with AppleScript 2.0 can be used by any version of AppleScript back to version 1.1, provided it does not use features of AppleScript, scripting additions, or scriptable applications that are unavailable in that version.

## What Is AppleScript?

AppleScript is a scripting language created by Apple. It allows users to directly control scriptable Macintosh applications, as well as parts of macOS itself. You can create scripts—sets of written instructions—to automate repetitive tasks, combine features from multiple scriptable applications, and create complex workflows.

A scriptable application is one that can be controlled by a script. For AppleScript, that means being responsive to interapplication messages, called *Apple events*, sent when a script command targets the application. (Apple events can also be sent directly from other applications and macOS.)

AppleScript itself provides a very small number of commands, but it provides a framework into which you can plug many task-specific commands—those provided by scriptable applications and scriptable parts of macOS.

Most script samples and script fragments in this guide use scriptable features of the Finder application, scriptable parts of macOS, or scriptable applications distributed with macOS, such as TextEdit (located in `/Applications`).

## Who Should Read This Document?

You should use this document if you write or modify AppleScript scripts, or if you create scriptable applications and need to know how scripts should work.

*AppleScript Language Guide* assumes you are familiar with the high-level information about AppleScript found in [AppleScript Overview](#).

## Organization of This Document

This guide describes the AppleScript language in a series of chapters and appendixes.

The first five chapters introduce components of the language and basic concepts for using it, then provide additional overview on working with script objects and handler routines:

- [AppleScript Lexical Conventions](#) describes the characters, symbols, keywords, and other language elements that make up statements in an AppleScript script.

- [AppleScript Fundamentals](#) describes basic concepts that underly the terminology and rules covered in the rest of this guide.
- [Variables and Properties](#) describes common issues in working with variables and properties, including how to declare them and how AppleScript interprets their scope.
- [Script Objects](#) describes how to define, initialize, send commands to, and use inheritance with script objects.
- [About Handlers](#) provides information on using handlers (a type of function available in AppleScript) to factor and reuse code.

The following chapters provide reference for the AppleScript Language:

- [Class Reference](#) describes the classes AppleScript defines for common objects used in scripts.
- [Commands Reference](#) describes the commands that are available to any script.
- [Reference Forms](#) describes the syntax for specifying an object or group of objects in an application or other container.
- [Operators Reference](#) provides a list of the operators AppleScript supports and the rules for using them, along with sections that provide additional detail for commonly used operators.
- [Control Statements Reference](#) describes statements that control when and how other statements are executed. It covers standard conditional statements, as well as statements used in error handling and other operations.
- [Handler Reference](#) shows the syntax for defining and calling handlers and describes other statements you use with handlers.

The following chapter describes an AppleScript-related feature of macOS:

- [Folder Actions Reference](#) describes how you can write and attach script handlers to specific folders, such that the handlers are invoked when the folders are modified.

The following appendixes provide additional information about the AppleScript language and how to work with errors in scripts:

- [AppleScript Keywords](#) lists the keywords of the AppleScript language, provides a brief description for each, and points to related information.
- [Error Numbers and Error Messages](#) describes error numbers and error messages you may see in working with AppleScript scripts.
- [Working with Errors](#) provides detailed examples of handling errors with [try Statements](#) and [error Statements](#).

- [Double Angle Brackets](#) describes when you are likely to see double angle brackets (or chevrons— « » ) in scripts and how you can work with them.
- [Libraries using Load Script](#) describes how to save libraries of handlers and access them from other scripts.
- [Unsupported Terms](#) lists terms that are no longer supported in AppleScript.

## Conventions Used in This Guide

Glossary terms are shown in *boldface* where they are defined.

The following conventions are used in syntax descriptions:

language element	Plain computer font indicates an element that you type exactly as shown. If there are special symbols (for example, + or & ), you also type them exactly as shown.
<i>placeholder</i>	Italic text indicates a placeholder that you replace with an appropriate value.
[optional]	Brackets indicate that the enclosed language element or elements are optional.
(a group)	<p>Parentheses group elements together.</p> <p>However, the parentheses shown in <a href="#">Handler Syntax (Positional Parameters)</a> are part of the syntax.</p>
[optional]...	Three ellipsis points (...) after a group defined by brackets indicate that you can repeat the group of elements within brackets 0 or more times.
a   b   c	Vertical bars separate elements in a group from which you must choose a single element. The elements are often grouped within parentheses or brackets.
Filenames shown in scripts	<p>Most filenames shown in examples in this document include extensions, such as rtf for a TextEdit document. Use of extensions in scripts is generally dependent on the “Show all file extensions” setting in the Advanced pane of Finder Preferences.</p> <p>To work with the examples on your computer, you may need to modify either that setting or the filenames.</p>

## See Also

These Apple documents provide additional information for working with AppleScript:

- See *Getting Started with AppleScript* for a guided quick start, useful to both scripters and developers.
- See [AppleScript Overview](#), including the chapter [Scripting with AppleScript](#), for a high-level overview of AppleScript and its related technologies.
- See *Getting Started With Scripting & Automation* for information on the universe of scripting technologies available in macOS.
- See [AppleScript Terminology and Apple Event Codes](#) for a list of many of the scripting terms defined by Apple.

For additional information on working with the AppleScript language and creating scripts, see one of the comprehensive third-party documents available in bookstores and online.

---

Source: [https://developer.apple.com/library/archive/documentation/AppleScript/Conceptual/AppleScriptLangGuide/introduction/ASLR\\_intro.html](https://developer.apple.com/library/archive/documentation/AppleScript/Conceptual/AppleScriptLangGuide/introduction/ASLR_intro.html)