



Tech Info Library

AppleScript: Description (11/95)

Article Created: 20 April 1993

Article Reviewed/Updated: 27 November 1995

TOPIC -----

AppleScript is a full-featured, object-oriented, programming language built into the Macintosh operating system. It can access internal software functionality of applications supporting the Open Scripting Architecture (OSA) of System 7.

DISCUSSION -----

AppleScript extends the Macintosh graphical user interface to offer scripting for automation, customization, and application integration. It allows tailoring applications for developers. The AppleScript product consists of a language, a system software extension, a scripting environment, and add-on utilities.

FEATURES

- A natural, English-like syntax that can be localized and/or replaced.
- Integration with the Open Scripting Architecture for seamless communication with multiple applications both within a computer and across networks.
- Use of OSA standards that enable a consistent scripting experience.
- A Script Editor utility that records scripts from user actions.

BENEFITS

- Rapid custom solution development for business problems.
- Automation of complex tasks and work flows within organizations.
- Simplification of desktop and network administration within organizations.
- Automation of routine tasks that can increase personal productivity.

CONFIGURATION REQUIREMENTS

AppleScript requires a Macintosh configured with:

- A 68000 or greater processor
- Minimum 4 megabytes of RAM
- System 7.0 or later

RAM IMPACT

With no scripts running, AppleScript takes up less than 5K of RAM. A running script occupies less than 250K of RAM. Editing and compiling a script consumes about 750K RAM.

The AppleScript Extension requires 250K of disk space when installed uncompressed whether on the hard disk or on floppy disks.

The AppleScript Script Editor requires 150K of disk space. The Scriptable Text Editor requires 100K of disk space.

LANGUAGE

The AppleScript language takes an "English-like" approach. In addition, it's flexible enough to adapt easily to the needs of languages other than English. The base AppleScript language includes language control structures (repeat, if...then, and so on) and a library of built-in commands.

The default syntax of AppleScript is similar to HyperTalk in its natural language approach, but is more regular and predictable. You can replace the default AppleScript syntax with localized language dialects or programming languages (BASIC-like, C-like).

SYSTEM SOFTWARE

AppleScript is implemented as a component, and makes use of the QuickTime Component Manager. This makes it accessible to any developer who wants to take advantage of it to give an application intuitive, consistent scripting capabilities. Implementing the scripting system at the system software level makes scripting an integral part of the Macintosh computing environment.

SCRIPTING ENVIRONMENTS

AppleScript is delivered with a simple script generation and editing facility known as the Script Editor. The Script Editor generates scripts by "watching" the user's actions. Then you can edit and run the scripts.

The End User and Professional versions of the Script Editor differ in particular features, but the basic interface is the same.

ADD-ON EXTENSIONS AND UTILITIES

Several add-on components complete the scripting product for different users. Sample Scripts help teach in-house developers how to use scripting to solve pertinent business problems. They can modify sample scripts to meet particular needs.

End Users can use sample scripts to make using the Macintosh more convenient and effective. A script launching utility launches scripts while the computer's owner is absent.

Finally, Open Scripting Architecture Extensions (OSACS) can extend the reach of AppleScript into other development arenas.

USAGE SCENARIOS

The following categories are the major classes of functionality that AppleScripts enable. AppleScripts create new, custom solutions from existing applications. These could be internal or commercial solution sets. A custom front-end might be built that uses attached scripts to communicate with different applications while keeping this activity transparent to the user.

Work Flow

You can use scripts to automate the process of working: to move data, documents, and information from person to person, computer to computer, department to department, and application to application.

Work flow scripts will automate some aspect of a group's work process such as tracking and reporting the status of project elements, managing versions, routing documents, and archiving files.

They are likely to afford a low-cost minimum-feature alternative for those who don't require extensive document management systems. Work flow scripts access, manage, and report the "data about data" in their working environment and drive applications for communication, including electronic mail.

Administration

Scripts perform administrative functions including software installation and upgrades, backups, remote user support, and troubleshooting across local area networks.

Unattended Computing

Unattended scripts take the place of the user interaction with the Macintosh and provide for the execution of predetermined instructions when specified conditions are met.

Scripts let you "set-and-forget" a series of things for an application to do, or several applications together. You can write scripts that test for certain conditions in your environment, in some document, or in another Macintosh across the network, and execute when those conditions are met.

It's possible to impose time conditions on unattended scripts to allow delayed execution, often overnight.

This article is one of many available through the Apple Fax center. For a complete list of available Fax documents, search the Tech Info Library for Apple

Fax Document Index or call the Apple Fax line at 1-800-505-0171 and select document number 20000 (Apple Fax - Document Index - Product Support Literature).

The Apple Fax center is available free of charge 24 hours a day, 7 days a week.

Article Change History:

27 Nov 1995 - Added keyword and review for accuracy.

12 Jan 1995 - Revised title slightly.

14 Oct 1994 - Reviewed for consistency and updated format.

Support Information Services

Copyright 1993-95, Apple Computer, Inc.

Keywords: specsht, kfax

=====

This information is from the Apple Technical Information Library.

19960627 16:33:15.00

Tech Info Library Article Number: 11893