
Required Apple Events

This section describes the required Apple events-the Apple events your application must support to be 7.0-friendly-and the descriptor types for all parameters of the required Apple events. It also describes how to write the handlers for these events, and it provides sample code.

To support the required Apple events, you must set the necessary flags in the 'SIZE' resource of your application, install entries into your application's Apple event dispatch table, add code to the event loop of your application to recognize high-level events, and call the **AEProcessAppleEvent** function, as described in the preceding two sections. You must also write handlers to handle each Apple event; this section describes how to write these handlers.

When a user opens or prints a file from the Finder, the Finder sets up the information your application can use to determine which files to open or print. In version 7.0, if your application supports high-level events, the Finder communicates this information to your application through the required Apple events. See **Getting Files Selected From the Finder** for a code example showing how to handle these events.

The Finder sends one of the required Apple events to your application to request that it open or print a list of documents, inform it that the Finder has just opened your application, or inform it that the Finder is about to terminate your application.

These are the required Apple events:

Apple event	Requested action
<u>Open Application event</u>	Perform tasks associated with opening your application
<u>Open Documents event</u>	Open the specified documents
<u>Print Documents event</u>	Print the specified documents
<u>Quit Application event</u>	Perform tasks-such as releasing memory, requesting the user to save documents, and so on-associated with quitting; when appropriate, the Finder sends this event to an application immediately after sending it a <u>Print Documents event</u> or if the user chooses Restart or Shut Down from the Finder's Special menu.

The Finder uses the required Apple events as part of the new mechanisms in system 7.0 for launching and terminating applications. This new method of communicating Finder information to your application replaces the mechanisms used in earlier versions of system software.

Applications that do not support high-level events can still use the **CountAppFiles**, **GetAppFiles**, and **ClrAppFiles** procedures (or the

GetAppParms procedure) to get the Finder information. See the **Segment Loader** description for information on these routines. To make your application 7.0-friendly and compatible with earlier versions of system software, it must support both the old and new mechanisms.

Use the **Gestalt** function to determine whether the **Apple Event Manager** is present. If it is and the **isHighLevelEventAware** flag is set in your application's **'SIZE' resource**, your application receives the Finder information through the required Apple events.

If your application accepts **high-level events**, the Finder sends it an **Open Application event**, **Open Documents event**, or **Print Documents event** immediately after launching your application. Upon receiving any of these events, your application should perform the action requested by the event.

Note: This section describes the required Apple events as they are sent by the Finder. When sent by other applications or processes, these same Apple events-which are among the **core Apple events** described in the *Apple Event Registry*-can include optional parameters not listed here. To be 7.0-friendly, your application only needs to handle the required parameters that are described in this section.

<u>Open Application event</u>	perform tasks associated with opening an application
<u>Event class</u>	<u>kCoreEventClass</u>
<u>Event ID</u>	<u>kAEOpenApplication</u>
<u>Parameters</u>	None
<u>Requested action</u>	Perform any tasks-such as opening an untitled document window-that you would normally perform when a user opens your application.

<u>Open Documents event</u>	open the specified documents
<u>Event class</u>	<u>kCoreEventClass</u>
<u>Event ID</u>	<u>kAEOpenDocuments</u>
<u>Keyword</u>	<u>keyDirectObject</u>
<u>Descriptor type</u>	<u>typeAEList</u>
<u>Data</u>	A list of alias records for the documents to be opened
<u>Requested action</u>	Open the documents specified in the <u>keyDirectObject</u> parameter.

<u>Print Documents event</u>	print the specified documents
<u>Event class</u>	<u>kCoreEventClass</u>
<u>Event ID</u>	<u>kAEPrintDocuments</u>
<u>Keyword</u>	<u>keyDirectObject</u>
<u>Descriptor type</u>	<u>typeAEList</u>

Data	A list of <u>alias records</u> for the documents to be printed
Requested action	Print the documents specified in the <u>keyDirectObject</u> parameter without opening windows for the documents.

<u>Quit Application event</u>	perform tasks associated with quitting
<u>Event class</u>	<u>kCoreEventClass</u>
<u>Event ID</u>	<u>kAEQuitApplication</u>
Parameters	None
Requested action	Perform any tasks that your application would normally perform when the user chooses Quit. Such tasks typically include asking the user if he or she wants to save documents that have been changed. When appropriate, the Finder sends this event to an application immediately after sending it a <u>Print Documents event</u> or if the user chooses Restart or Shut Down from the Finder's Special menu..

Your application needs to recognize only two descriptor types to handle the required Apple events: descriptor lists and alias records. The Open Documents event and Print Documents event use descriptor lists to store a list of documents to open. Each document is specified as an alias record in the descriptor list.

You can retrieve the data that specifies the document to open as an alias record, or you can request that the **Apple Event Manager** coerce the alias record to a file system specification (FSSpec) record. The file system specification record provides a standard method of identifying files in version 7.0. See the **File Manager** description for a complete description of how to specify files using file system specification records.