

Creating and Managing the Special Handler Dispatch Tables

The Apple Event Manager provides three functions that allow you to create and manage the special handler dispatch tables. The `AEInstallSpecialHandler` function installs an entry for a special handler in either the application or system special handler dispatch table. The `AEGetSpecialHandler` function returns the handler for a specified special handler. The `AERemoveSpecialHandler` function removes a special handler from either the application or system special handler dispatch table.

Responding to Apple Events

You can also use the `AEInstallSpecialHandler`, `AEGetSpecialHandler`, and `AERemoveSpecialHandler` functions to install, get, and remove object callback functions—including system object callback functions, which cannot be installed with the `AESetObjectCallbacks` function. When calling any of these three functions, use one of the following constants as the value of the `functionClass` parameter to specify the object callback function:

Object callback function	Constant
Object-counting function	<code>keyAECCountProc</code>
Object-comparison function	<code>keyAECompareProc</code>
Token disposal function	<code>keyDiposeTokenProc</code>
Error callback function	<code>keyAEGetErrDescProc</code>
Mark token function	<code>keyAEMarkTokenProc</code>
Object-marking function	<code>keyAEMarkProc</code>
Mark-adjusting function	<code>keyAEAdjustMarksProc</code>

You can also use the `AERemoveSpecialHandler` function to disable all the Apple Event Manager routines that support object specifier records. To do this, specify the constant `keySelectProc` in the `functionClass` parameter as described on page 4-102.

AEInstallSpecialHandler

You can use the `AEInstallSpecialHandler` function to install a special handler in either the application or system special handler dispatch table.

```
FUNCTION AEInstallSpecialHandler (functionClass: AEKeyword;
                                handler: ProcPtr;
                                isSysHandler: Boolean): OSErr;
```

`functionClass`

The keyword for the special handler that is installed. The `keyPreDispatch` constant identifies a handler with the same parameters as an Apple event handler called immediately before the Apple Event Manager dispatches an Apple event. Any of the constants for object callback functions listed above can also be specified in this parameter. If there was already an entry in the specified special handler dispatch table for the same value of `functionClass`, it is replaced.

`handler`

A pointer to the special handler. Note that a handler in the system special handler dispatch table must reside in the system heap; thus, if the value of the `isSysHandler` parameter is `TRUE`, the `handler` parameter should point to a location in the system heap. Otherwise, if you put your system handler code in your application heap, use `AERemoveSpecialHandler` to remove the handler when your application quits.

Responding to Apple Events

`isSysHandler`

The special handler dispatch table to which to add the handler. If the value of this parameter is `TRUE`, the handler is added to the system handler dispatch table and made available to all applications. If the value is `FALSE`, the handler is added to the application handler table.

DESCRIPTION

The `AEInstallSpecialHandler` function creates an entry in either your application's special handler dispatch table or the system special handler dispatch table. You must supply parameters that specify the keyword for the special handler that is installed, the handler routine, and whether the handler is to be added to the system special handler dispatch table or your application's special handler dispatch table.

SPECIAL CONSIDERATIONS

Before an application calls a system special handler, system software has set up the A5 register for the calling application. For this reason, a system special handler should never use A5 global variables or anything that depends on a particular context; otherwise, the application that calls the system handler may crash.

RESULT CODES

<code>noErr</code>	0	No error
<code>paramErr</code>	-50	Parameter error (handler pointer is NIL or odd)
<code>memFullErr</code>	-108	Not enough room in heap zone
<code>errAENotASpecialFunction</code>	-1714	Wrong keyword for a special function

AEGetSpecialHandler

You can use the `AEGetSpecialHandler` function to get a specified special handler.

```
FUNCTION AEGetSpecialHandler (functionClass: AEKeyword;
                             VAR handler: ProcPtr;
                             isSysHandler: Boolean): OSErr;
```

`functionClass`

The keyword for the special handler that is installed. The `keyPreDispatch` constant identifies a handler with the same parameters as an Apple event handler that is called immediately before the Apple Event Manager dispatches an Apple event. Any of the constants for object callback functions listed on page 4-100 can also be specified in this parameter.

`handler` A pointer to the special handler.

Responding to Apple Events

`isSysHandler`

Specifies the special handler dispatch table from which to get the handler. If the value of this parameter is `TRUE`, the handler is taken from the system special handler dispatch table. If the value is `FALSE`, the handler is taken from the application's special handler dispatch table.

RESULT CODES

<code>noErr</code>	0	No error
<code>memFullErr</code>	-108	Not enough room in heap zone
<code>errAENotASpecialFunction</code>	-1714	Wrong keyword for a special handler

AERemoveSpecialHandler

You can use the `AERemoveSpecialHandler` function to remove a handler from a special handler table.

```
FUNCTION AERemoveSpecialHandler (functionClass: AEKeyword;
                                handler: ProcPtr;
                                isSysHandler: Boolean): OSErr;
```

`functionClass`

The keyword for the special handler to be removed. In addition to the constants for object callback functions listed on page 4-100, two other values are allowed for the `functionClass` parameter: `keyPreDispatch` and `keySelectProc`. The `keyPreDispatch` constant identifies a handler with the same parameters as an Apple event handler that is called immediately before the Apple Event Manager dispatches an Apple event. The `keySelectProc` constant indicates that you want to disable the Object Support Library—that is, all the routines described in the chapter “Resolving and Creating Object Specifier Records” in this book (see the description that follows for more information).

`handler`

A pointer to the special handler to be removed. Although the `functionClass` parameter would be sufficient to identify the handler to be removed, providing the `handler` parameter is a safeguard that you remove the correct handler.

`isSysHandler`

Specifies the special handler dispatch table from which to remove the handler. If the value of this parameter is `TRUE`, the handler is taken from the system special handler dispatch table. If the value is `FALSE`, the handler is removed from the application special handler dispatch table.

Responding to Apple Events

DESCRIPTION

In addition to using the `AERemoveSpecialHandler` function to remove specific special handlers, you can use the function to disable, within your application only, all Apple Event Manager routines that support Apple event objects—that is, all the routines available to your application as a result of linking the Object Support Library (OSL) and calling the `AEObjectInit` function.

An application that expects its copy of the OSL to move after it is installed—for example, an application that keeps it in a stand-alone code resource—would need to do this. When an application calls `AEObjectInit` to initialize the OSL, the OSL installs the addresses of its routines as extensions to the pack. If those routines move, the addresses become invalid.

To disable the OSL, you should pass the keyword `keySelectProc` in the `functionClass` parameter, `NIL` in the `handler` parameter, and `FALSE` in the `isSysHandler` parameter. Once you have called the `AERemoveSpecialHandler` function with these parameters, subsequent calls by your application to any of the Apple Event Manager routines that support Apple event objects will return errors. To initialize the OSL after disabling it with the `AERemoveSpecialHandler` function, your application must call `AEObjectInit` again.

If you expect to initialize the OSL and disable it several times, you should call `AERemoveObjectAccessor` to remove your application's object accessor functions from your application's object accessor dispatch table before you call `AERemoveSpecialHandler`.

RESULT CODES

<code>noErr</code>	0	No error
<code>memFullErr</code>	-108	Not enough room in heap zone
<code>errAENotASpecialFunction</code>	-1714	Wrong keyword for a special function
