Chapter X **The AppleScript Suite**

Introduction to the AppleScript suite

The AppleScript suite defines additional classes and events used for communicating with Apple events.

Using object specifiers in place of other parameters

In all of the suites except the Finder suite, you can substitute an object specifier for any parameter of an Apple event that is not already defined to be an object specifier. This object specifier must specify a single object. When you substitute an object specifier for another parameter, the actual value of the parameter is the value of the default descriptor record for the specified object (that is, the value you get when you send a Get Data Apple event for the object and do not specify a particular descriptor type for the result).

Replices - Special fields in the reply event record

OSA mentions a number of additional pieces of information that may be retrieved by calling OSAScriptError. AppleScript returns this information as fields in an Apple event reply record in addition to the usual keyAEResult, keyErrorNumber and keyErrorString fields. This allows, for instance, script applications to communicate their error state fully, without need for additional events to extract script error information.

Reply Parameters

keyAEResult

Description: The result of the subroutine call. If an error occurred (i.e. there is a keyErrorNumber field also present in the reply) then the result corresponds with the kOSAErrorPartialResult selector for OSAScriptError.

Descriptor Type: typeWildCard

Required or Optional? Optional

keyErrorNumber

Description: The result code for the event. This is the same as the kOSAErrorNumber selector for OSAScriptError.

Descriptor Type: typeShortInteger

Required or Optional? Optional

keyErrorString

Description: A character string that describes the error, if any, that occurred when the event was handled. This is the same as the kOSAErrorBriefMessage selector for OSAScriptError.

Descriptor Type: typeIntlText

Required or Optional? Optional

kOSAErrorApp

Description: The address of the actual application that caused the error. Sometimes the error returned from one application is the result of a remote send to another application–this field corresponds to that other application. This is the same as the kOSAErrorApp selector for OSAScriptError.

Descriptor Type: cApplication

Required or Optional? Optional

kOSAErrorOffendingObject

Description: The object that actually caused the error–usually an object specifier. This is the same as the kOSAErrorOffendingObject selector for OSAScriptError.

Descriptor Type: typeWildCard

Required or Optional? Optional

kOSAErrorExpectedType

Description: The type expected whenever a type or coercion error occurs. This is the same as the kOSAErrorExpectedType selector for OSAScriptError.

Descriptor Type: typeType

Required or Optional? Optional

kOSAErrorRange

Description: The error source range of the script that caused the error–used for editing. This is the same as the kOSAErrorRange selector for OSAScriptError.

Descriptor Type:	typeOSAErrorRange
Required or Optional?	Optional

Apple events defined in the AppleScript suite

The Apple events defined in the AppleScript suite are described in the following sections. Table X-1 lists these Apple events.

n **Table X-1** Apple events defined in the AppleScript suite

Name	**Requested action**
Comment	Send a comment to be recorded
No Operation	Do nothing on launch
Subroutine Call	Call a user-defined subroutine
Operator Events	Send overloaded operator events
Get AETE	Get the application terminology
Get AEUT	Get the system terminologies

Comment—Send a comment to be recorded

The comment event is used to add comments to a recorded script. Handling a comment event should have no effect.

Event Class kASAppleScriptSuite

Event ID kASCommentEvent

Parameters

keyDirectObject

Description:	A comment string
Descriptor Type:	typeChar
Required or Optional?	Required

No Operation —Do nothing on launch

The No Operation event can be used as an alternative to the standard launch events sent by the Finder: Open Application, Open Documents, and Print Documents. It indicates that the application is being launched by a script, which will subsequently send more events. There are two reasons for the No Operation event: 1) From a scripting point of view, most applications perform undesirable actions when they receive the standard Open Application event. For example, they open a new document, or display a modal dialog waiting for user response. 2) The event an application is launched with cannot have a reply, thus no errors are returned to the script. No Operation event, when used as for launching, will avoid the standard double-click behavior, and allow subsequent events to return error replies.

Event Class kASAppleScriptSuite

Event ID kASLaunchEvent

Notes AppleScript always sends the launch event NoReply. No errors are returned.

Subroutine Call—Call a user-defined subroutine

The subroutine event is sent for calls to user-defined subroutines. They contain the name of the event as a text string, and any parameters for the event. A standard set of prepositional parameters are available.

Event Class kASAppleScriptSuite

Event ID kASSubroutineEvent

Parameters

keyDirectObject		
	Description:	The target, or list of arguments
	Descriptor Type:	typeObjectSpecifier or typeAEList
	Required or Optional?	Optional
keyASSubroutineName		
	Description:	The subroutine name
	Descriptor Type:	typeChar

Required or Optional?	Required

keyASPrepositionAt
keyASPrepositionAt
keyASPrepositionIn
keyASPrepositionFrom
keyASPrepositionFor
keyASPrepositionTo
keyASPrepositionThru
keyASPrepositionThrough
keyASPrepositionBy
keyASPrepositionOn
keyASPrepositionInto
keyASPrepositionOnto
keyASPrepositionBetween
keyASPrepositionAgainst
keyASPrepositionOutOf
keyASPrepositionInsteadOf
keyASPrepositionAsideFrom
keyASPrepositionAround
keyASPrepositionBeside
keyASPrepositionBeneath
keyASPrepositionUnder
keyASPrepositionOver
keyASPrepositionAbove
keyASPrepositionBelow
keyASPrepositionApartFrom
keyASPrepositionAbout
keyASPrepositionSince
keyASPrepositionUntil

Description:	Named prepositional parameters
Descriptor Type:	typeWildCard
Required or Optional?	Optional

keyASUserRecordFields

Description:	Parameters with user-defined names
Descriptor Type:	typeUserRecordFields
Required or Optional?	Optional

Reply Parameters

keyAEResult

Description:	The result of the subroutine call
Descriptor Type:	typeWildCard
Required or Optional?	Optional

(see Replies section for additional error reply parameters)

Operator Events—Send overloaded operator events

These events allow applications to handle some binary operators directly. The events will be sent when one of the arguments of the binary operator is an AppleScript reference (object specifier). If the application returns errAEEventNotHandled, then the binary operator will be handled by AppleScript, and the event will not be sent to that application again. The binary operator events are currently the same as the Object Support Library comparisons.

Event Class	kASAppleScriptSuite
Event ID	kASEqual
Event ID	kASNotEqual
Event ID	kASGreaterThan
Event ID	kASGreaterThanOrEqual
Event ID	kASLessThan
Event ID	kASLessThanOrEqual
Event ID	kASStartsWith
Event ID	kASEndsWith
Event ID	kASContains

Parameters

keyDirectObject

Description:	The left argument of the operator
Descriptor Type:	typeWildCard
Required or Optional?	Required

keyASArg

Description:	The right argument of the operator
Descriptor Type:	typeWildCard
Required or Optional?	Required

Reply Parameters

keyAEResult

Description:	The result of the operator
Descriptor Type:	typeWildCard
Required or Optional?	Required

(see Replies section for additional error reply parameters)

Get AETE Event—Get the application terminology

Returns the terminology resource of an application. AppleScript installs a system handler for this event at system init time, so that remote applications can access application terminology. For complete information on the get AETE event, see Chapter 8, "Apple Event Terminology Resources," in *Inside Macintosh, Interapplication Communication*.

Event Class kASAppleScriptSuite

Event ID kGetAETE

Reply Parameters

keyAEResult

Description:	The AETE resource of the application, or a list of AETE resources.
Descriptor Type:	typeWildCard
Required or Optional?	Optional

Get AEUT Event—Get the system terminologies

Returns the list of system terminologies. AppleScript installs a system handler for this event at system init time, and manages the terminology for Scripting Additions. Applications should not override or handle this event.

Event Class kASAppleScriptSuite

Event ID kGetAEUT

Reply Parameters

keyAEResult

Description:	A list of AEUT resources.
Descriptor Type:	typeAEList
Required or Optional?	Optional

Object classes defined in the AppleScript suite

The Apple event object classes defined in the AppleScript suite are described in the following sections. Table X-2 lists these object classes.

cList—A list of elements with other properties

The cList class is an extension of the Apple event typeAEList. It defines a number of properties for lists, but its structure is identical to typeList. AppleScript handles these properties on lists it manages.

Superclass	typeAEList
Default Descriptor Type	typeAEList

Properties

pLength

Description:	The number of items in the list
Object Class ID:	typeWildCard
Inherited?	No
Modifiable or Non-modifiable?	Non-modifiable

pRest

Description:	The rest of a list–everything but the first item
Object Class ID:	cList
Inherited?	No
Modifiable or Non-modifiable?	Non-Modifiable

pReverse

Description:	A new list with the items in reverse order
Object Class ID:	cList
Inherited?	No
Modifiable or Non-modifiable?	Non-modifiable

cRecord—A record of elements with other properties

The cRecord class is an extension of the Apple event typeAERecord. It's structure is identical to typeAERecord, but it defines a special field that contains other user-defined field names and their values.

Superclass typeAERecord

Default Descriptor Type typeAERecord

Properties (record fields)

keyASUserRecordFields

Description:	List of user-defined property names and their values
Object Class ID:	typeUserRecordFields
Inherited?	No
Modifiable or Non-modifiable?	Modifiable

Notes AppleScript doesn't allow the keyASUserRecordFields to be access directly, but instead unbundles the user-defined field names and their values into an actual record. For example, if in the AppleScript record {name:"fred", age:25} the name field is defined in the terminology resource (pName) but age is not, AppleScript will create the cRecord {'pnam':"fred", 'usrf':{"age", 25}}. Note that the user-defined record fields are a list of alternating names and their values. AppleScript also decodes this form of cRecord into the original format. If an application does not want to bother with user-defined record field names, an appropriate error can be returned.

cScript—Script Objects

Descriptors of type cScript contain script data in their data handle such as that written out by OSAStore. cScripts correspond to both script objects in the AppleScript language, and to contexts in the AppleScript OSA interface.

Superclass cObject

Default Descriptor Type cScript

Properties

pASParent

Description:	The parent script of this script
Object Class ID:	typeWildCard
Modifiable or Non-modifiable?	Modifiable

Notes AppleScript will send a cScript object for any event parameter if a script object instance is passed to a remote event. It does this by calling OSAStore on the script object and adding the resulting script data to the event. However, since script objects are also recipients of messages, special rules apply whenever a script object is used as the direct parameter of a message: First, the message is sent to the script object itself. If the script handles the message, then we're done. If the script object either continues or fails to handle the message and inherits from another object belonging to an application (e.g. an object specifier) then one of two cases apply: (1) The direct parameter of the event specifies that it requires data of type cScript. In this case the script object is saved with OSAStore and sent in the event. (2) The direct parameter of the event specifies that it requires any other type. In this case the first non-script object parent of the script object (e.g. the object specifier) is sent as the direct parameter of the event instead.

Descriptor types defined in the AppleScript suite

The descriptor types defined in the AppleScript suite are described in the following sections. Table X-4 lists these descriptor types.

n **Table X-4** Descriptor types defined in the AppleScript suite

Descriptor type	**Description**
typeUserRecordFields	List of user-defined record fields
typeAETE	Apple Event Terminology Extension
typeAEUT	Apple Event User Terminology

typeUserRecordFields—List of user-defined record fields

A typeUserRecordFields descriptor record specifies a list of user-defined record fields–their names and values. The list contains alternating name value pairs, where each name is typeIntlText and each value is typeWildCard.

Data Size Variable

Notes Notes

typeAETE—Apple Event Terminology Extension

A typeAETE descriptor record contains information about the Apple event objects and events supported by an application.

Description The dataHandle field of a typeAETE descriptor record contains the same data as the resource type 'aete'.

For complete information about resource type 'aete' , see Chapter 8, “Apple Event Terminology Resources,” in *Inside Macintosh, Interapplication Communication*.

typeAEUT—Apple Event User Terminology

A typeAEUT descriptor record contains information about the Apple event objects and events supported by the system and extensions.

Description The dataHandle field of a typeAEUT descriptor record contains the same data as the resource type 'aeut' (which is the same as the type 'aete').

For complete information about resource type 'aeut' , see Chapter 8, “Apple Event Terminology Resources,” in *Inside Macintosh, Interapplication Communication*.

Constants defined in the AppleScript suite

Table X-7 lists the constants defined in the AppleScript suite.

n **Table X-7** Constants defined in the AppleScript suite

Constant	Value
kASAppleScriptSuite	'ascr'
kASTypeNamesSuite	'tpnm'
typeAETE	'aete'
typeAEUT	'aeut'
kGetAETE	'gdte'
kGetAEUT	'gdut'
kASCommentEvent	'cmnt'
kASLaunchEvent	'noop'
kASSubroutineEvent	'psbr'
keyASSubroutineName	'snam'
keyASArg	'arg '
keyASUserRecordFields	'usrf'
typeUserRecordFields	'list'
kASEqual	kAEEquals
kASNotEqual	'≠ '
kASGreaterThan	kAEGreaterThan
kASGreaterThanOrEqual	kAEGreaterThanEquals
kASLessThan	kAELessThan
kASLessThanOrEqual	kAELessThanEquals
kASStartsWith	kAEBeginsWith
kASEndsWith	kAEEndsWith
kASContains	kAEContains
keyASPrepositionAt	'at '
keyASPrepositionIn	'in '
keyASPrepositionFrom	'from'
keyASPrepositionFor	'for '

```
keyASPrepositionTo      'to  '
keyASPrepositionThru    'thru'
keyASPrepositionThrough 'thgh'
keyASPrepositionBy      'by  '
keyASPrepositionOn      'on  '
keyASPrepositionInto    'into'
keyASPrepositionOnto    'onto'
keyASPrepositionBetween 'btwn'
keyASPrepositionAgainst 'agst'
keyASPrepositionOutOf   'outo'
keyASPrepositionInsteadOf           'isto'
keyASPrepositionAsideFrom           'asdf'
keyASPrepositionAround  'arnd'
keyASPrepositionBeside  'bsid'
keyASPrepositionBeneath 'bnth'
keyASPrepositionUnder   'undr'
keyASPrepositionOver    'over'
keyASPrepositionAbove   'abve'
keyASPrepositionBelow   'belw'
keyASPrepositionApartFrom           'aprt'
keyASPrepositionAbout   'abou'
keyASPrepositionSince   'snce'
keyASPrepositionUntil   'till'
cEventIdentifier        'evnt'
cScript                 'scpt'
cList                   'list'
cRecord                 'reco'
cSeconds                'scnd'
pLength                 'leng'
pReverse                'rvse'
pRest                   'rest'
pASParent               'pare'
pASPrintLength          'prln'
pASPrintDepth           'prdp'
pASPrintDepth           'prdp'
enumConsiderations      'cons'
kAECase                 'case'
```

kAEDiacritic	'diac'
kAEWhiteSpace	'whit'
kAEHyphens	'hyph'
kAEExpansion	'expa'
kAEPunctuation	'punc'
kAEZenkakuHankaku	'zkhk'
kAESmallKana	'skna'
kAEKataHiragana	'hika'
kASConsiderReplies	'rmte'
cZone	'zone'
cMachine	'mach'