NSScriptClassDescription Class Reference

Cocoa > Scripting & Automation

2007-07-17

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

NSScriptClassDescription Class Reference

Inherits from	NSClassDescription : NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Declared in	NSScriptClassDescription.h
Companion guides	Cocoa Scripting Guide Key-Value Coding Programming Guide
Related sample code	Quartz Composer WWDC 2005 TextEdit TextEditPlus

Overview

An instance of `NSScriptClassDescription` describes a script class that a Cocoa application supports.

A scriptable application provides scriptability information that describes the commands and objects scripters can use in scripts that target the application. That includes information about the classes those scriptable objects are created from.

An application's scriptability information is collected automatically by an instance of `NSScriptSuiteRegistry`. The registry object creates an `NSScriptClassDescription` for each class it finds and caches these objects in memory. Cocoa scripting uses registry information in handling scripting requests that target the application.

A class description instance stores the name, attributes, relationships, and supported commands for a class. For example, a scriptable `document` class for a drawing application might support attributes such as `file` and `file type`, relationships such as collections of `circles`, `rectangles`, and `lines`, and commands such as `align` and `rotate`.

As with many of the classes in Cocoa's built-in scripting support, your application may never need to directly work with instances of `NSScriptClassDescription`. However, one case where you might need access to a class description is if you override `objectSpecifier` in a scriptable class. For information on how to do this, see Object Specifiers in *Cocoa Scripting Guide*.

Another case where your application may need access to class description information is if you override `indicesOfObjectsByEvaluatingWithContainer:count:` in a specifier class.

Although you can subclass `NSScriptClassDescription`, it is unlikely that you would need to do so, or even to create instances of it.

Tasks

Initializing a Script Class Description

- `initWithSuiteName:className:dictionary:` (page 12)
  Initializes and returns a newly allocated instance of `NSScriptClassDescription`.

Getting a Script Class Description

+ `classDescriptionForClass:` (page 7)
  Returns the class description for the specified class or, if it is not scriptable, for the first superclass that is.
- `classDescriptionForKey:` (page 9)
  Returns the class description instance for the class type of the specified attribute or relationship.
- `superclassDescription` (page 15)
  Returns the class description instance for the superclass of the receiver's class.

Getting Basic Information About the Script Class

- `className` (page 9)
  Returns the name of the class the receiver describes, as provided at initialization time.
- `defaultSubcontainerAttributeKey` (page 10)
  Returns the value of the `DefaultSubcontainerAttribute` entry of the class dictionary from which the receiver was instantiated.
- `implementationClassName` (page 11)
  Returns the name of the Objective-C class instantiated to implement the scripting class.
- `isLocationRequiredToCreateForKey:` (page 12)
  Returns a Boolean value indicating whether an insertion location must be specified when creating a new object in the specified to-many relationship of the receiver.
- `suiteName` (page 14)
  Returns the name of the receiver's suite.

Getting and Comparing Apple Event Codes

- `appleEventCode` (page 8)
  Returns the Apple event code associated with the receiver's class.
- `appleEventCodeForKey:` (page 8)
  Returns the Apple event code for the specified attribute or relationship in the receiver.

- `matchesAppleEventCode:` (page 13)
  Returns a Boolean value indicating whether a primary or secondary Apple event code in the receiver matches the passed code.

Getting Attribute and Relationship Information

- `hasOrderedToManyRelationshipForKey:` (page 10)
  Returns a Boolean value indicating whether the described class has an ordered to-many relationship identified by the specified key.
- `hasPropertyForKey:` (page 10)
  Returns a Boolean value indicating whether the described class has a property identified by the specified key.
- `hasReadablePropertyForKey:` (page 11)
  Returns a Boolean value indicating whether the described class has a readable property identified by the specified key.
- `hasWritablePropertyForKey:` (page 11)
  Returns a Boolean value indicating whether the described class has a writable property identified by the specified key.
- `keyWithAppleEventCode:` (page 13)
  Given an Apple event code that identifies a property or element class, returns the key for the corresponding attribute, one-to-one relationship, or one-to-many relationship.
- `typeForKey:` (page 16)
  Returns the name of the declared type of the attribute or relationship identified by the passed key.
- `isReadOnlyKey:` (page 17) Deprecated in Mac OS X v10.5
  Returns a Boolean value indicating whether a specified property in the receiver is read-only. (Deprecated. Use `hasWritablePropertyForKey:` (page 11) instead.)

Getting Command Information

- `selectorForCommand:` (page 14)
  Returns the selector associated with the receiver for the specified command description.
- `supportsCommand:` (page 15)
  Returns a Boolean value indicating whether the receiver or any superclass supports the specified command.

Class Methods

classDescriptionForClass:

Returns the class description for the specified class or, if it is not scriptable, for the first superclass that is.

```
+ (NSScriptClassDescription *)classDescriptionForClass:(Class)aClass
```

Parameters

aClass

The class whose description is needed.

Return Value

The class description for the class specified by *aClass* or, if that class isn't scriptable, for the class description for the first superclass that is. Returns `nil` if it doesn't find a scriptable class.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSScriptClassDescription.h`

Instance Methods

appleEventCode

Returns the Apple event code associated with the receiver's class.

```
- (FourCharCode)appleEventCode
```

Return Value

The Apple event code associated with the receiver's class. This is the primary code used to identify the class in Apple events.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `appleEventCodeForKey:` (page 8)
- `matchesAppleEventCode:` (page 13)

Declared In

`NSScriptClassDescription.h`

appleEventCodeForKey:

Returns the Apple event code for the specified attribute or relationship in the receiver.

```
- (FourCharCode)appleEventCodeForKey:(NSString *)key
```

Parameters

key

The identifying key for an attribute or relationship of the receiver.

Return Value

The four-character Apple event code associated with the attribute or relationship identified by *key* in the receiver or, if none exists, in the class description for the receiver's superclass. Returns `0` if no such attribute or relationship is found.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `appleEventCode` (page 8)
- `matchesAppleEventCode:` (page 13)

Declared In
`NSScriptClassDescription.h`

classDescriptionForKey:

Returns the class description instance for the class type of the specified attribute or relationship.

```
- (NSScriptClassDescription *)classDescriptionForKey:(NSString *)key
```

Parameters

key
The identifying key for an attribute or relationship of the receiver.

Return Value
The instance of `NSScriptClassDescription` for the type of the attribute or relationship specified by *key*. Returns `nil` if no scriptable property corresponds to *key*.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `superclassDescription` (page 15)

Declared In
`NSScriptClassDescription.h`

className

Returns the name of the class the receiver describes, as provided at initialization time.

```
- (NSString *)className
```

Return Value
A class name. This may be either the human-readable name for the class—that is, the name that is used in a script—or the name of the Objective-C class that is instantiated to implement the class. To reliably obtain the implementation name, use `implementationClassName` (page 11).

Availability
Available in Mac OS X v10.0 and later.

See Also
- `suiteName` (page 14)

Declared In
`NSScriptClassDescription.h`

defaultSubcontainerAttributeKey

Returns the value of the `DefaultSubcontainerAttribute` entry of the class dictionary from which the receiver was instantiated.

```
- (NSString *)defaultSubcontainerAttributeKey
```

Return Value
The value of the default subcontainer attribute entry. Returns `nil` if the there was no such entry.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`NSScriptClassDescription.h`

hasOrderedToManyRelationshipForKey:

Returns a Boolean value indicating whether the described class has an ordered to-many relationship identified by the specified key.

```
- (BOOL)hasOrderedToManyRelationshipForKey:(NSString *)key
```

Parameters

key
The identifying key for a property of the receiver.

Return Value
`YES` if the described class has an ordered to-many relationship identified by the specified key; otherwise, `NO`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSScriptClassDescription.h`

hasPropertyForKey:

Returns a Boolean value indicating whether the described class has a property identified by the specified key.

```
- (BOOL)hasPropertyForKey:(NSString *)key
```

Parameters

key
The identifying key for a property of the receiver.

Return Value
`YES` if the described class has a property identified by the specified key; otherwise, `NO`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSScriptClassDescription.h`

hasReadablePropertyForKey:

Returns a Boolean value indicating whether the described class has a readable property identified by the specified key.

```
- (BOOL)hasReadablePropertyForKey:(NSString *)key
```

Parameters

key

The identifying key for a property of the receiver.

Return Value

`YES` if the described class has a readable property identified by the specified key; otherwise, `NO`.

Discussion

To determine if a property is read-only, invoke `hasWritablePropertyForKey:` (page 11)/

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSScriptClassDescription.h`

hasWritablePropertyForKey:

Returns a Boolean value indicating whether the described class has a writable property identified by the specified key.

```
- (BOOL)hasWritablePropertyForKey:(NSString *)key
```

Parameters

key

The identifying key for a property of the receiver.

Return Value

`YES` if the described class has a writable property identified by the specified key; otherwise, `NO`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSScriptClassDescription.h`

implementationClassName

Returns the name of the Objective-C class instantiated to implement the scripting class.

```
- (NSString *)implementationClassName
```

Return Value

An Objective-C class name.

Discussion
The name returned by the `className` (page 9) method for an instance of `NSScriptClassDescription` resulting from an sdef class declaration is the human-readable name for the class—that is, the name that is used in a script. To obtain the name of the Objective-C class instantiated to implement the class, use `implementationClassName`.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSScriptClassDescription.h`

initWithSuiteName:className:dictionary:

Initializes and returns a newly allocated instance of `NSScriptClassDescription`.

```
- (id)initWithSuiteName:(NSString *)suiteName className:(NSString *)className
    dictionary:(NSDictionary *)classDeclaration
```

Parameters

suiteName
The name of the suite (in the application's scriptability information) that the class belongs to. For example, "`AppName Suite`".

className
The name of the class that this instance describes.

classDeclaration
A class declaration dictionary of the sort that is valid in script suite property list files. This dictionary provides information about the class such as its attributes and relationships.

Return Value
The initialized instance. Returns `nil` if the event code value for the class description itself is missing or is not an `NSString`. Also returns `nil` if the superclass name or any of the subdictionaries of descriptions are not of the right type.

Discussion
This method registers `self` with the application's global instance of `NSScriptSuiteRegistry`.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`NSScriptClassDescription.h`

isLocationRequiredToCreateForKey:

Returns a Boolean value indicating whether an insertion location must be specified when creating a new object in the specified to-many relationship of the receiver.

```
- (BOOL)isLocationRequiredToCreateForKey:(NSString *)toManyRelationshipKey
```

Parameters

toManyRelationshipKey

The key for the to-many relationship that may require an insertion location.

Return Value

YES if an insertion location must be specified; otherwise, NO.

Discussion

A script command object that creates a new object in a to-many relationship needs to know whether an explicitly specified insertion location is required. It can get this information from an instance of NSScriptClassDescription. For example, NSMakeCommand uses this method to determine whether or not a specific make AppleScript command must have an at parameter.

Availability

Available in Mac OS X v10.2 and later.

Declared In

NSScriptClassDescription.h

keyWithAppleEventCode:

Given an Apple event code that identifies a property or element class, returns the key for the corresponding attribute, one-to-one relationship, or one-to-many relationship.

```
- (NSString *)keyWithAppleEventCode:(FourCharCode)appleEventCode
```

Parameters

appleEventCode

An Apple event code that identifies a property or element class.

Return Value

The key that corresponds to the property or element class identified by *appleEventCode* in the receiver or, if none exists, in a class description in the receiver's superclasses.

The four-character Apple event code associated with the attribute or relationship identified by *key* Returns 0 if no such attribute or relationship is found. Returns nil if it cannot find any such attribute or relationship.

Availability

Available in Mac OS X v10.0 and later.

See Also

- isReadOnlyKey: (page 17)
- typeForKey: (page 16)

Declared In

NSScriptClassDescription.h

matchesAppleEventCode:

Returns a Boolean value indicating whether a primary or secondary Apple event code in the receiver matches the passed code.

```
- (BOOL)matchesAppleEventCode:(FourCharCode)appleEventCode
```

Parameters

appleEventCode

An Apple event code to compare against the receiver's primary or secondary codes.

Return Value

YES if the receiver's primary four-character Apple event code or any of its secondary codes (its synonyms) matches *code*; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- appleEventCode (page 8)
- appleEventCodeForKey: (page 8)

Declared In

NSScriptClassDescription.h

selectorForCommand:

Returns the selector associated with the receiver for the specified command description.

```
- (SEL)selectorForCommand:(NSScriptCommandDescription *)commandDescription
```

Parameters

commandDescription

A description for a script command, such as duplicate, make, or move. Encapsulates the scriptability information for that command, such as its Objective-C selector, its argument names and types, and its return type (if any).

Return Value

The selector from the receiver for the command specified by *commandDescription*. Searches in the receiver first, then in any superclass. Returns NULL if no matching selector is found.

Availability

Available in Mac OS X v10.0 and later.

See Also

- supportsCommand: (page 15)

Declared In

NSScriptClassDescription.h

suiteName

Returns the name of the receiver's suite.

```
- (NSString *)suiteName
```

Return Value

The receiver's suite name. Within an application's scriptability information, named suites contain related sets of information.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `className` (page 9)

Declared In
`NSScriptClassDescription.h`

superclassDescription

Returns the class description instance for the superclass of the receiver's class.

```
- (NSScriptClassDescription *)superclassDescription
```

Return Value
A class description instance that describes the superclass of the receiver's class. Returns `nil` if the class has no superclass.

Discussion
The instance of `NSScriptClassDescription` that describes the superclass can be in the same suite as the receiver or in a different suite.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `classDescriptionForKey:` (page 9)

Declared In
`NSScriptClassDescription.h`

supportsCommand:

Returns a Boolean value indicating whether the receiver or any superclass supports the specified command.

```
- (BOOL)supportsCommand:(NSScriptCommandDescription *)commandDescription
```

Parameters
commandDescription
A description for a script command, such as `duplicate`, `make`, or `move`. Encapsulates the scriptability information for that command, such as its Objective-C selector, its argument names and types, and its return type (if any).

Return Value
`YES` if an the receiver or the instance of `NSScriptClassDescription` of any superclass of the receiver's class lists the command described by *commandDesc* among its supported commands; otherwise, `NO`.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `selectorForCommand:` (page 14)

Declared In
NSScriptClassDescription.h

typeForKey:

Returns the name of the declared type of the attribute or relationship identified by the passed key.

```
- (NSString *)typeForKey:(NSString *)key
```

Parameters

key

The identifying key for an attribute, one-to-one relationship, or one-to-many relationship of the receiver.

Return Value
The name of the declared type of the attribute or relationship identified by *key*; for example, "NSString". Searches in the receiver first, then in any superclass. Returns `nil` if no match is found.

Availability
Available in Mac OS X v10.0 and later.

See Also
- `isReadOnlyKey:` (page 17)
- `keyWithAppleEventCode:` (page 13)

Declared In
NSScriptClassDescription.h

Deprecated NSScriptClassDescription Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5

isReadOnlyKey:

Returns a Boolean value indicating whether a specified property in the receiver is read-only. (Deprecated in Mac OS X v10.5. Use `hasWritablePropertyForKey:` (page 11) instead.)

```
- (BOOL)isReadOnlyKey:(NSString *)key
```

Parameters

key

The identifying key for a property of the receiver.

Return Value

`YES` if the property specified by *key* exists in the receiver or in the `NSScriptClassDescription` for any superclass, and is read only; otherwise, `NO`.

Special Considerations

This method could return `NO` either because `key` is unrecognized or because writing to the property is not supported. Use `hasWritablePropertyForKey:` (page 11) instead.

Availability

Available in in Mac OS X v10.0.

Deprecated in Mac OS X v10.5.

See Also

- `keyWithAppleEventCode:` (page 13)
- `typeForKey:` (page 16)

Declared In

`NSScriptClassDescription.h`

Document Revision History

This table describes the changes to *NSScriptClassDescription Class Reference*.

Date	Notes
2007-07-17	Added new methods for Mac OS X version 10.5.
	The new methods are `implementationClassName` (page 11), `hasPropertyForKey:` (page 10), `hasOrderedToManyRelationshipForKey:` (page 10), `hasReadablePropertyForKey:` (page 11), and `hasWritablePropertyForKey:` (page 11).
	Noted that the method `isReadOnlyKey:` (page 17) is deprecated in Mac OS X v10.5 and that you should use `hasWritablePropertyForKey:` (page 11) instead.
	Clarified the description for the `className` (page 9) method.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index