NSScriptSuiteRegistry Class Reference

Developer

Contents

NSScriptSuiteRegistry Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in OS X v10.0 and later.
Companion guide	Cocoa Scripting Guide
Declared in	NSScriptSuiteRegistry.h

Overview

`NSScriptSuiteRegistry` functions as the top-level repository of scriptability information for an application at runtime.

Scriptability information specifies the terminology available for use in scripts that target an application. It also provides information, used by AppleScript and by Cocoa, about how support for that terminology is implemented in the application. This information includes descriptions of the scriptable object classes in an application and of the commands the application supports.

There are two standard formats for supplying scriptability information: the older script suite format, consisting of a script suite file and one or more script terminology files, and the newer scripting definition (or sdef) format, consisting of a single sdef file.

There is one instance of `NSScriptSuiteRegistry` per scriptable application. This registry object collects scriptability information when the application first needs to respond to an Apple event for which Cocoa hasn't installed a default event handler. It then creates one instance of `NSScriptClassDescription` for each object class and one instance of `NSScriptCommandDescription` for each command class, and installs a command handler for each command.

When a user executes an AppleScript script, Apple events are sent to the targeted application. Using the information stored in the registry object, Cocoa automatically converts incoming Apple events into script commands (based on `NSScriptCommand` or a subclass) that manipulate objects in the application.

The public methods of `NSScriptSuiteRegistry` are used primarily by Cocoa's built-in scripting support. You should not need to create a subclass of `NSScriptSuiteRegistry`.

For information on scriptability information formats, loading of scriptability information, and related topics, see "Scriptability Information" in Overview of Cocoa Support for Scriptable Applications in *Cocoa Scripting Guide*.

Tasks

Getting and Setting the Shared Instance

`+ setSharedScriptSuiteRegistry:` (page 6)
Sets the single, shared instance of `NSScriptSuiteRegistry` to `registry`.

`+ sharedScriptSuiteRegistry` (page 6)
Returns the single, shared instance of `NSScriptSuiteRegistry`, creating it first if it doesn't exist.

Getting Suite Information

`– suiteForAppleEventCode:` (page 12)
Returns the name of the suite definition associated with the given four-character Apple event code, `code`.

`– suiteNames` (page 12)
Returns the names of the suite definitions currently loaded by the application.

Getting and Registering Class Descriptions

`– classDescriptionsInSuite:` (page 8)
Returns the class descriptions contained in the suite identified by `suiteName`.

`– classDescriptionWithAppleEventCode:` (page 8)
Returns the class description associated with the given four-character Apple event code, `code`.

`– registerClassDescription:` (page 11)
Registers class description `classDescription` for use by Cocoa's built-in scripting support by storing it in a per-suite internal dictionary under the class name.

Getting and Registering Command Descriptions

– `commandDescriptionsInSuite:` (page 9)

Returns the command descriptions contained in the suite identified by `suiteName`.

– `commandDescriptionWithAppleEventClass:andAppleEventCode:` (page 9)

Returns the command description identified by a suite's four-character Apple event code of the class (`eventClass`) and the four-character Apple event code of the command (`commandCode`).

– `registerCommandDescription:` (page 12)

Registers command description `commandDesc` for use by Cocoa's built-in scripting support by storing it in a per-suite internal dictionary under the command name.

Getting Other Suite Information

– `aeteResource:` (page 7)

Returns an `NSData` object that contains data in `'aete'` resource format describing the scriptability information currently known to the application.

– `appleEventCodeForSuite:` (page 7)

Returns the Apple event code associated with the suite named `suiteName`, such as `'core'` for the Core suite.

– `bundleForSuite:` (page 8)

Returns the bundle containing the suite-definition property list (extension `.scriptSuite`) identified by `suiteName`.

Loading Suites

– `loadSuiteWithDictionary:fromBundle:` (page 10)

Loads the suite definition encapsulated in `dictionary`; previously, this suite definition was parsed from a `.scriptSuite` property list contained in a framework or in `bundle`.

– `loadSuitesFromBundle:` (page 10)

Loads the suite definitions in bundle `aBundle`, invoking `loadSuiteWithDictionary:fromBundle:` (page 10) for each suite found.

Class Methods

setSharedScriptSuiteRegistry:

Sets the single, shared instance of `NSScriptSuiteRegistry` *to* `registry`*.*

```
+ (void)setSharedScriptSuiteRegistry:(NSScriptSuiteRegistry *)registry
```

Availability
Available in OS X v10.0 and later.

Declared in
`NSScriptSuiteRegistry.h`

sharedScriptSuiteRegistry

Returns the single, shared instance of `NSScriptSuiteRegistry`*, creating it first if it doesn't exist.*

```
+ (NSScriptSuiteRegistry *)sharedScriptSuiteRegistry
```

Discussion
If it creates an instance, and if the application provides scriptability information in the script suite format, the method loads suite definitions in all frameworks and other bundles that the application currently imports or includes; if information is provided in the sdef format, the method loads information only from the specified sdef file. If in reading scriptability information an exception is `raised` because of parsing errors, it handles the exception by printing a line of information to the console.

Availability
Available in OS X v10.0 and later.

See Also
– `loadSuiteWithDictionary:fromBundle:` (page 10)

Declared in
`NSScriptSuiteRegistry.h`

Instance Methods

aeteResource:

Returns an `NSData` object that contains data in `'aete'` resource format describing the scriptability information currently known to the application.

```
- (NSData *)aeteResource:(NSString *)languageName
```

Discussion
This method is typically invoked to implement the `get aete` Apple event for an application that provides scriptability information in the script suite format. The `languageName` argument is the name of a language for which a localized resource directory (such as `English.lproj`) exists. This language indication specifies the set of `.scriptTerminology` files to be used to generate the data. `NSScriptSuiteRegistry` does not create an `'aete'` resource unless this method is called.

Availability
Available in OS X v10.0 and later.

See Also
– `appleEventCodeForSuite:` (page 7)

Declared in
`NSScriptSuiteRegistry.h`

appleEventCodeForSuite:

Returns the Apple event code associated with the suite named `suiteName`, such as `'core'` for the Core suite.

```
- (FourCharCode)appleEventCodeForSuite:(NSString *)suiteName
```

Availability
Available in OS X v10.0 and later.

See Also
– `suiteForAppleEventCode:` (page 12)

Declared in
`NSScriptSuiteRegistry.h`

bundleForSuite:

Returns the bundle containing the suite-definition property list (extension `.scriptSuite`) identified by `suiteName`.

```
- (NSBundle *)bundleForSuite:(NSString *)suiteName
```

Availability
Available in OS X v10.0 and later.

Declared in
`NSScriptSuiteRegistry.h`

classDescriptionsInSuite:

Returns the class descriptions contained in the suite identified by `suiteName`.

```
- (NSDictionary *)classDescriptionsInSuite:(NSString *)suiteName
```

Discussion
Each class description (instance of `NSScriptClassDescription`) in the returned dictionary is identified by class name.

Availability
Available in OS X v10.0 and later.

See Also
– `classDescriptionWithAppleEventCode:` (page 8)
– `registerClassDescription:` (page 11)

Declared in
`NSScriptSuiteRegistry.h`

classDescriptionWithAppleEventCode:

Returns the class description associated with the given four-character Apple event code, `code`.

```
- (NSScriptClassDescription *)classDescriptionWithAppleEventCode:(FourCharCode)code
```

Discussion
Overriding behavior is important here. Multiple classes can have the same code if the classes have an uninterrupted linear inheritance from one another. For example, if class B is a subclass of A and class C is a subclass of B, and all three classes have the same four-character Apple event code, then this method returns the class description for class C.

Availability
Available in OS X v10.0 and later.

See Also
– `classDescriptionsInSuite:` (page 8)
– `registerClassDescription:` (page 11)

Declared in
`NSScriptSuiteRegistry.h`

commandDescriptionsInSuite:

Returns the command descriptions contained in the suite identified by `suiteName`.

```
- (NSDictionary *)commandDescriptionsInSuite:(NSString *)suiteName
```

Discussion
Each command description (instance of `NSScriptCommandDescription`) in the returned dictionary is identified by command name.

Availability
Available in OS X v10.0 and later.

See Also
– `commandDescriptionWithAppleEventClass:andAppleEventCode:` (page 9)
– `registerCommandDescription:` (page 12)

Declared in
`NSScriptSuiteRegistry.h`

commandDescriptionWithAppleEventClass:andAppleEventCode:

Returns the command description identified by a suite's four-character Apple event code of the class (`eventClass`) and the four-character Apple event code of the command (`commandCode`).

```
- (NSScriptCommandDescription
*)commandDescriptionWithAppleEventClass:(FourCharCode)eventClass
andAppleEventCode:(FourCharCode)commandCode
```

Availability
Available in OS X v10.0 and later.

See Also
– `commandDescriptionsInSuite:` (page 9)
– `registerCommandDescription:` (page 12)

Declared in
`NSScriptSuiteRegistry.h`

loadSuitesFromBundle:

Loads the suite definitions in bundle `aBundle`, invoking `loadSuiteWithDictionary:fromBundle:` (page 10) for each suite found.

```
- (void)loadSuitesFromBundle:(NSBundle *)aBundle
```

Discussion
If errors occur while method is parsing a suite-definition file, the method logs error messages to the console.

Availability
Available in OS X v10.0 and later.

Declared in
`NSScriptSuiteRegistry.h`

loadSuiteWithDictionary:fromBundle:

Loads the suite definition encapsulated in `dictionary`; previously, this suite definition was parsed from a `.scriptSuite` property list contained in a framework or in `bundle`.

```
- (void)loadSuiteWithDictionary:(NSDictionary *)dictionary fromBundle:(NSBundle
*)bundle
```

Discussion

The method extracts information from the dictionary and caches it in various internal collection objects. If keys are missing or values are of the wrong type, it logs messages to the console. It also registers class descriptions and command descriptions. In registering a class description, it invokes the `NSClassDescription` class method `registerClassDescription:forClass:`. In registering a command description, it arranges for the Apple event translator to handle incoming Apple events that represent the defined commands.

This method is invoked when the shared instance is initialized and when bundles are loaded at runtime. Prior to invoking it, `NSScriptSuiteRegistry` creates the dictionary argument from the `.scriptSuite` property list. If you invoke this method in your code, you should try to do it before the application receives its first Apple event.

Availability

Available in OS X v10.0 and later.

See Also

- `loadSuitesFromBundle:` (page 10)
- `registerClassDescription:` (page 11)
- `registerCommandDescription:` (page 12)
+ `sharedScriptSuiteRegistry` (page 6)

Declared in

`NSScriptSuiteRegistry.h`

registerClassDescription:

Registers class description `classDescription` for use by Cocoa's built-in scripting support by storing it in a per-suite internal dictionary under the class name.

```
- (void)registerClassDescription:(NSScriptClassDescription *)classDescription
```

Availability

Available in OS X v10.0 and later.

See Also

- `loadSuiteWithDictionary:fromBundle:` (page 10)
- `registerCommandDescription:` (page 12)

Declared in

`NSScriptSuiteRegistry.h`

registerCommandDescription:

Registers command description `commandDesc` for use by Cocoa's built-in scripting support by storing it in a per-suite internal dictionary under the command name.

```
- (void)registerCommandDescription:(NSScriptCommandDescription *)commandDesc
```

Discussion
Also registers with the single, shared instance of `NSAppleEventManager` to handle incoming Apple events that should be handled by the command.

Availability
Available in OS X v10.0 and later.

See Also
- `loadSuiteWithDictionary:fromBundle:` (page 10)
- `registerClassDescription:` (page 11)

Declared in
`NSScriptSuiteRegistry.h`

suiteForAppleEventCode:

Returns the name of the suite definition associated with the given four-character Apple event code, `code`.

```
- (NSString *)suiteForAppleEventCode:(FourCharCode)code
```

Availability
Available in OS X v10.0 and later.

See Also
- `suiteNames` (page 12)

Declared in
`NSScriptSuiteRegistry.h`

suiteNames

Returns the names of the suite definitions currently loaded by the application.

```
- (NSArray *)suiteNames
```

Availability

Available in OS X v10.0 and later.

See Also

– suiteForAppleEventCode: (page 12)

Declared in

NSScriptSuiteRegistry.h

Document Revision History

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

Date	Notes
2007-04-10	Modified the class overview.
2006-05-23	First publication of this content as a separate document.

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