SBObject Class Reference

Cocoa > Interapplication Communication

2007-05-29

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

Simultaneously published in the United States and Canada.

Contents

SBObject Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/ScriptingBridge.framework
Availability	Available in Mac OS X v10.5
Declared in	SBObject.h

Overview

The `SBObject` class declares methods that can be invoked on any object in a scriptable application. It defines methods for getting elements and properties of an object, as well as setting a given object to a new value.

Each `SBObject` is built around an object specifier, which tells Scripting Bridge how to locate the object. Therefore, you can think of an `SBObject` as a reference to an object in an target application rather than an object itself. To bypass this reference-based approach and force evaluation, use the `get` (page 7) method.

Typically, rather than create `SBObject` instances explictly, you receive `SBObject` objects by calling methods of an `SBApplication` subclass. For example, if you wanted to get an `SBObject` representing the current iTunes track, you would use code like this (where `iTunesTrack` is a subclass of `SBObject`):

```
iTunesApplication *iTunes = [SBApplication
applicationWithBundleIdentifier:@"com.apple.iTunes"];
iTunesTrack *track = [iTunes currentTrack];
```

You can discover the names of dynamically generated classes such as `iTunesApplication` and `iTunesTrack` by examining the header file created by the `sdp` tool. Alternatively, you give these variables the dynamic Objective-C type `id`.

Tasks

Initializing a Scripting Bridge Object

`- init` (page 7)
Initializes and returns an instance of an `SBObject` subclass.

`- initWithData:` (page 8)
Returns an instance of an `SBObject` subclass initialized with the given data.

- `initWithProperties:` (page 9)
  Returns an instance of an `SBObject` subclass initialized with the specified properties.
- `initWithElementCode:properties:data:` (page 8)
  Returns an instance of an `SBObject` subclass initialized with the specified properties and data and added to the designated element array.

Getting Referenced Data

- `get` (page 7)
  Forces evaluation of the receiver, causing the real object to be returned immediately.

Sending Apple Events

- `sendEvent:id:parameters:` (page 10)
  Sends an Apple event with the given event class, event ID, and format to the target application.
- `setTo:` (page 11)
  Sets the receiver to a specified value.

Getting Properties and Elements

- `propertyWithClass:code:` (page 9)
  Returns an object of the designated scripting class representing the specified property of the receiver
- `propertyWithCode:` (page 10)
  Returns an object representing the specified property of the receiver.
- `elementArrayWithCode:` (page 6)
  Returns an array containing every child of the receiver with the given class-type code.

Instance Methods

elementArrayWithCode:

Returns an array containing every child of the receiver with the given class-type code.

```
- (SBElementArray *)elementArrayWithCode:(DescType)code
```

Parameters

code
  A four-character code that identifies a scripting class.

Return Value
An `SBElementArray` object containing every child of the receiver whose class matches *code*.

Discussion
`SBObject` subclasses use this method to implement application-specific property accessor methods. You should not need to call this method directly.

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

See Also
- `propertyWithCode:` (page 10)

Declared In
`SBObject.h`

get

Forces evaluation of the receiver, causing the real object to be returned immediately.

```
- (id)get
```

Return Value
The object referenced by the receiver.

Discussion
This method forces the current object reference (the receiver) to be evaluated, resulting in the return of the referenced object. By default, Scripting Bridge deals with references to objects until you actually request some concrete data from them or until you call the `get` method.

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

Declared In
`SBObject.h`

init

Initializes and returns an instance of an `SBObject` subclass.

```
- (id)init
```

Return Value
An `SBObject` object or `nil` if the object could not be initialized.

Discussion
Scripting Bridge does not actually create an object in the target application until you add the object returned from this method to an element array (`SBElementArray`).

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

See Also
- `initWithProperties:` (page 9)
- `initWithData:` (page 8)
- `initWithElementCode:properties:data:` (page 8)

Declared In
`SBObject.h`

initWithData:

Returns an instance of an `SBObject` subclass initialized with the given data.

```
- (id)initWithData:(id)data
```

Parameters

data

An object containing data for the new `SBObject` object. The data varies according to the type of scripting object to be created.

Return Value
An `SBObject` object or `nil` if the object could not be initialized.

Discussion
Scripting Bridge does not actually create an object in the target application until you add the object returned from this method to an element array (`SBElementArray`).

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

See Also
- `init` (page 7)
- `initWithProperties:` (page 9)
- `initWithElementCode:properties:data:` (page 8)

Declared In
`SBObject.h`

initWithElementCode:properties:data:

Returns an instance of an `SBObject` subclass initialized with the specified properties and data and added to the designated element array.

```
- (id)initWithElementCode:(DescType)code properties:(NSDictionary *)properties
    data:(id)data
```

Parameters

code

A four-character code used to identify an element in the target application's scripting interface. See *Apple Event Manager Reference* for details.

properties

A dictionary with keys specifying the names of properties (that is, attributes or to-one relationships) and the values for those properties. Pass `nil` if you are initializing the object by *data* only.

data

An object containing data for the new `SBObject` object. The data varies according to the type of scripting object to be created. Pass `nil` if you initializing the object by *properties* only.

Return Value
An `SBObject` object or `nil` if the object could not be initialized.

Discussion
Unlike the other initializers of this class, this method not only initializes the `SBObject` object but adds it to a specified element array. This method is the designated initializer.

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

See Also
- `init` (page 7)
- `initWithData:` (page 8)
- `initWithProperties:` (page 9)

Declared In
`SBObject.h`

initWithProperties:

Returns an instance of an `SBObject` subclass initialized with the specified properties.

```
- (id)initWithProperties:(NSDictionary *)properties
```

Parameters

properties

A dictionary with keys specifying the names of properties (that is, attributes or to-one relationships) and the values for those properties.

Return Value
An `SBObject` object or `nil` if the object could not be initialized.

Discussion
Scripting Bridge does not actually create an object in the target application until you add the object returned from this method to an element array (`SBElementArray`).

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

See Also
- `init` (page 7)
- `initWithData:` (page 8)
- `initWithElementCode:properties:data:` (page 8)

Declared In
`SBObject.h`

propertyWithClass:code:

Returns an object of the designated scripting class representing the specified property of the receiver

```
- (SBObject *)propertyWithClass:(Class)class code:(AEKeyword)code
```

Parameters

class

The `SBObject` subclass with which to instantiate the object.

code

A four-character code that uniquely identifies a property of the receiver.

Return Value
An instance of the designated *class* that represents the receiver's property identified by *code*.

Discussion
`SBObject` subclasses use this method to implement application-specific property accessor methods. You should not need to call this method directly.

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

See Also
- `propertyWithCode:` (page 10)

Declared In
`SBObject.h`

propertyWithCode:

Returns an object representing the specified property of the receiver.

```
- (SBObject *)propertyWithCode:(AEKeyword)code
```

Parameters

code

A four-character code that uniquely identifies a property of the receiver.

Return Value
An object representing the receiver's property as identified by *code*.

Discussion
`SBObject` subclasses use this method to implement application-specific property accessor methods. You should not need to call this method directly.

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

See Also
- `propertyWithClass:code:` (page 9)
- `elementArrayWithCode:` (page 6)

Declared In
`SBObject.h`

sendEvent:id:parameters:

Sends an Apple event with the given event class, event ID, and format to the target application.

```
- (id)sendEvent:(AEEventClass)eventClass id:(AEEventID)eventID
    parameters:(DescType)firstParamCode,...
```

Parameters

eventClass

The event class of the Apple event to be sent.

eventID

The event ID of the Apple event to be sent.

firstParamCode,...

A list of four-character parameter codes (`DescType`) and object values (`id`) terminated by a zero.

Return Value
The target application's Apple event sent in reply; it is converted to a Cocoa object of an appropriate type.

Discussion
Scripting Bridge uses this method to communicate with target applications. If the target application responds to this method by sending an Apple event representing an error, the receiver calls its delegate's `eventDidFail:withError:` method. If no delegate has been assigned, the receiver raises an exception.

You should rarely have to call this method directly.

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

See Also
`- setTo:` (page 11)

Declared In
`SBObject.h`

setTo:

Sets the receiver to a specified value.

```
- (void)setTo:(id)value
```

Parameters

value

The data the receiver should be set to. It can be an `NSString`, `NSNumber`, `NSArray`, `SBObject`, or any other type of object supported by the Scripting Bridge framework.

Discussion
You should not call this method directly.

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

See Also
`- sendEvent:id:parameters:` (page 10)

Declared In
`SBObject.h`

Document Revision History

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

Date	Notes
2007-05-29	New document that describes the Scripting Bridge class that provides methods for communicating with objects in a scriptable application.

Document Revision History

Index