UINib Class Reference

Data Management: File Management

2010-05-26

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

Simultaneously published in the United States and Canada.

Contents

UINib Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 4.0 and later.
Declared in	UINib.h

Overview

Instances of the `UINib` class serve as object wrappers, or containers, for Interface Builder nib files. An `UINib` object caches the contents of a nib file in memory, ready for unarchiving and instantiation. When your application needs to instantiate the contents of the nib file it can do so without having to load the data from the nib file first, improving performance. The `UINib` object can automatically release this cached nib data to free up memory for your application under low-memory conditions, reloading that data the next time your application instantiates the nib. Your application should use `UINib` objects whenever it needs to repeatedly instantiate the same nib data. For example, if your table view uses a nib file to instantiate table view cells, caching the nib in a `UINib` object can provide a significant performance improvement.

When you create an `UINib` object using the contents of a nib file, the object loads the object graph in the referenced nib file, but it does not yet unarchive it. To unarchive all of the nib data and thus truly instantiate the nib your application calls the `instantiateWithOwner:options:` method on the `UINib` object. The steps that the `UINib` object follows to instantiate the nib's object graph are described in detail in *Resource Programming Guide*.

Tasks

Creating a Nib Object

`+ nibWithNibName:bundle:` (page 6)
Returns an `UINib` object initialized to the nib file in the specified bundle.

`+ nibWithData:bundle:` (page 6)
Creates an `UINib` object from nib data stored in memory.

Instantiating a Nib

`- instantiateWithOwner:options:` (page 7)
Unarchives and instantiates the in-memory contents of the receiver's nib file, creating a distinct object tree and set of top level objects.

Class Methods

nibWithData:bundle:

Creates an `UINib` object from nib data stored in memory.

```
+ (UINib *)nibWithData:(NSData *)data bundle:(NSBundle *)bundleOrNil
```

Parameters

data
A block of memory that contains nib data.

bundleOrNil
The bundle in which to search for resources referenced by the nib. If you specify `nil`, this method looks for the nib file in the main bundle.

Return Value
The initialized `NSNib` object or `nil` if there were errors during initialization.

Discussion
The `UINib` object looks for the nib file in the bundle's language-specific project directories first, followed by the `Resources` directory.

The preferred mechanism for instantiating `UINib` objects is with the `nibWithNibName:bundle:` class method. A `UINib` object instantiated using the `nibWithData:bundle:` class method cannot release the cached data under low memory conditions. Your application should be prepared to release the `UINib` object and the data under low memory conditions, recreating both the next time the application needs to instantiate the nib.

Availability
Available in iOS 4.0 and later.

Declared In
`UINib.h`

nibWithNibName:bundle:

Returns an `UINib` object initialized to the nib file in the specified bundle.

```
+ (UINib *)nibWithNibName:(NSString *)name bundle:(NSBundle *)bundleOrNil
```

Parameters

name
The name of the nib file, without any leading path information.

bundleOrNil

The bundle in which to search for the nib file. If you specify `nil`, this method looks for the nib file in the main bundle.

Return Value
The initialized `UINib` object or `nil` if there were errors during initialization or the nib file could not be located.

Discussion
The `UINib` object looks for the nib file in the bundle's language-specific project directories first, followed by the `Resources` directory.

Availability
Available in iOS 4.0 and later.

Declared In
`UINib.h`

Instance Methods

instantiateWithOwner:options:

Unarchives and instantiates the in-memory contents of the receiver's nib file, creating a distinct object tree and set of top level objects.

```
- (NSArray *)instantiateWithOwner:(id)ownerOrNil options:(NSDictionary *)optionsOrNil
```

Parameters

ownerOrNil

The object to use as the owner of the nib file. If the nib file has an owner, you must specify a valid object for this parameter.

optionsOrNil

A dictionary containing the options to use when opening the nib file. For a list of available keys for this dictionary, see *NSBundle UIKit Additions Reference*.

Return Value
An autoreleased `NSArray` object containing the top-level objects from the nib file.

Discussion
You can use this method to instantiate the objects in a nib and provide them to your code. This method unarchives each object, initializes it, sets its properties to their configured values, and reestablishes any connections to other objects. For detailed information about the nib-loading process, see *Resource Programming Guide*.

If the nib file contains any proxy objects beyond just the File's Owner proxy object, you can specify the runtime replacement objects for those proxies using the options dictionary. In that dictionary, add the `UINibExternalObjects` key and set its value to a dictionary containing the names of any proxy objects (the keys) and the real objects to use in their place. The proxy object's name is the string you assign to it in the Name field of the Interface Builder inspector window.

Availability
Available in iOS 4.0 and later.

Declared In
`UINib.h`

Document Revision History

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

Date	Notes
2010-05-26	New document that describes an object for unarchiving and instantiating a nib file.

Document Revision History