NSObject UIKit Additions Reference

2008-03-26

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

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

Apple, the Apple logo, Cocoa, Cocoa Touch, iPhone, Keychain, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSObject UIKit Additions Reference

Inherits from	none
Framework	/System/Library/Frameworks/UIKit.framework
Availability	Available in iOS 2.0 and later.
Declared in	UINibLoading.h

Overview

This category adds methods to the Foundation framework's `NSObject` class. The method in this category provides support for loading nib files into your application.

Tasks

Responding to Being Loaded from a Nib File

`– awakeFromNib` (page 5)
Prepares the receiver for service after it has been loaded from an Interface Builder archive, or nib file.

Instance Methods

awakeFromNib

Prepares the receiver for service after it has been loaded from an Interface Builder archive, or nib file.

```
- (void)awakeFromNib
```

Discussion
The nib-loading infrastructure sends an `awakeFromNib` message to each object recreated from a nib archive, but only after all the objects in the archive have been loaded and initialized. When an object receives an `awakeFromNib` message, it is guaranteed to have all its outlet and action connections already established.

You must call the `super` implementation of `awakeFromNib` to give parent classes the opportunity to perform any additional initialization they require. Although the default implementation of this method does nothing, many UIKit classes provide non-empty implementations. You may call the `super` implementation at any point during your own `awakeFromNib` method.

Note: During Interface Builder's test mode, this message is also sent to objects instantiated from loaded Interface Builder plug-ins. Because plug-ins link against the framework containing the object definition code, Interface Builder is able to call their `awakeFromNib` method when present. The same is not true for custom objects that you create for your Xcode projects. Interface Builder knows only about the defined outlets and actions of those objects; it does not have access to the actual code for them.

During the instantiation process, each object in the archive is unarchived and then initialized with the method befitting its type. Objects that conform to the `NSCoding` protocol (including all subclasses of `UIView` and `UIViewController`) are initialized using their `initWithCoder:` method. All objects that do not conform to the `NSCoding` protocol are initialized using their `init` method. After all objects have been instantiated and initialized, the nib-loading code reestablishes the outlet and action connections for all of those objects. It then calls the `awakeFromNib` method of the objects. For more detailed information about the steps followed during the nib-loading process, see Nib Files and Cocoa in *Resource Programming Guide*.

Important: Because the order in which objects are instantiated from an archive is not guaranteed, your initialization methods should not send messages to other objects in the hierarchy. Messages to other objects can be sent safely from within an `awakeFromNib` method.

Typically, you implement `awakeFromNib` for objects that require additional set up that cannot be done at design time. For example, you might use this method to customize the default configuration of any controls to match user preferences or the values in other controls. You might also use it to restore individual controls to some previous state of your application.

Availability

Available in iOS 2.0 and later.

See Also

`awakeAfterUsingCoder:` (NSObject class)

`initWithCoder:` (NSCoding protocol)

`initialize` (NSObject class)

Related Sample Code

SpeakHere

Declared In

`UINibLoading.h`

Document Revision History

This table describes the changes to *NSObject UIKit Additions Reference*.

Date	Notes
2008-03-26	First revision of this document.