NSNibAwaking Protocol Reference

Cocoa > Resource Management

2007-02-28

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

Simultaneously published in the United States and Canada.

Contents

NSNibAwaking Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/AppKit.framework
Companion guide	Resource Programming Guide
Declared in	NSNibLoading.h

Overview

This informal protocol consists of a single method, `awakeFromNib` (page 5). Classes can implement this method to initialize state information after objects have been loaded from an Interface Builder archive (nib file).

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
An `awakeFromNib` message is sent to each object loaded from the archive, but only if it can respond to the message, and 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 instance variables set.

Note: During Interface Builder's test mode, this method is also sent to objects instantiated from loaded palettes, which include executable code for the objects. It is not sent to objects created using the Classes display of the nib file window in Interface Builder.

During the instantiation process, each object in the archive is unarchived and then initialized with the method befitting its type. Cocoa views (and custom views that can be customized using an associated Interface Builder palette) are initialized using their `initWithCoder:` method. Custom views are initialized using their `initWithFrame:` method. Custom classes that have been instantiated in the nib are initialized using their `init` method.

Once all objects have been instantiated and initialized from the archive, the nib loading code attempts to reestablish the connections between each object's outlets and the corresponding target objects. If your custom objects have outlets, an `NSNib` object attempts to reestablish any connections you created in Interface Builder. It starts by trying to establish the connections using your object's own methods first. For each outlet that needs a connection, the `NSNib` object looks for a method of the form `set`*OutletName*`:` in your object. If that method exists, the `NSNib` object calls it, passing the target object as a parameter. If you did not define a setter method with that exact name, the NSNib object searches the object for an instance variable (of type `IBOutlet id`) with the corresponding outlet name and tries to set its value directly. If an instance variable with the correct name cannot be found, initialization of that connection does not occur. Finally, after all the objects are fully initialized, each receives an `awakeFromNib` message.

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 `awakeFromNib`—by which time it's assured that all the objects are unarchived and initialized (though not necessarily awakened, of course).

Typically, you implement `awakeFromNib` for the class you associate with the "File's Owner" of the nib file. You might also want to implement this method for any other classes you instantiate directly in your nib file. The job of these objects is to give you a hook for connecting the nib file objects to other objects in your application. Once that job is finished, you can either dispose of the objects or use them as a controller for the nib file objects.

An example of how you might use `awakeFromNib` is shown below. Suppose your nib file has two custom views that must be positioned relative to each other at runtime. Trying to position them at initialization time might fail because the other view might not yet be unarchived and initialized yet. However, you can position both of them in the nib file owner's `awakeFromNib` method. In the code below, `firstView` and `secondView` are outlets of the file's owner:

```
- (void)awakeFromNib {
    NSRect viewFrame;

    viewFrame = [firstView frame];
    viewFrame.origin.x += viewFrame.size.width;
    [secondView setFrame:viewFrame];
    return;
}
```

It is recommended that you maintain a one-to-one correspondence between your File's Owner objects and their associated nib files. Loading two nib files with the same File's Owner object causes that object's `awakeFromNib` method being called twice, which could cause some data structures to be reinitialized in undesired ways. It is also recommended that you avoid loading other nib files from your `awakeFromNib` method implementation.

You should call the `super` implementation of `awakeFromNib` only if you know for certain that your superclass provides an implementation. Because the Application Kit does not provide a default implementation of the `awakeFromNib` method, calling `super` results in an exception if the parent class does not implement it. Classes whose immediate parent class is `NSObject` or `NSView` do not need to call the `super` implementation. For any other classes, you can use the `instancesRespondToSelector:` class method of `NSObject` to determine if the parent class responds to `awakeFromNib` and call the method if it does.

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

See Also
`+ loadNibNamed:owner:` (NSBundle Additions)
`- awakeAfterUsingCoder` (NSObject class)
`+ instancesRespondToSelector:` (NSObject class)
`- initWithCoder:` (NSCoding protocol)
`+ initialize` (NSObject class)

Related Sample Code
Dicey
iSpend
MyPhoto
NumberInput_IMKit_Sample
ViewController

Declared In
`NSNibLoading.h`

Document Revision History

This table describes the changes to *NSNibAwaking Protocol Reference*.

Date	Notes
2007-02-28	Updated guidance on when to call super implementation of awakeFromNib. Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index

A