NSNibAwaking

(informal protocol)

Category Of: NSObject

Declared In: AppKit/NSNibLoading.h

Protocol Description

This informal protocol consists of a single method, **awakeFromNib**. Classes can implement this method to perform final initialization of state after objects have been loaded from an Interface Builder archive.

Instance Methods

awakeFromNib

– (void)**awakeFromNib**

Implemented to prepare the receiver for service after it has been loaded from an Interface Builder archive, or *nib file*. 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's guaranteed to have all its outlet instance variables set.

Note: This method is also sent during Interface Builder's test mode to objects instantiated from loaded palettes, which include executable code for the objects. It isn't sent to objects defined solely by using the Classes display of the nib file window in Interface Builder.

When an Interface Builder archive is loaded into an application, each custom object from the archive is first initialized with an **init** message, or **initWithFrame:** if the object is a kind of NSView. It's then more specifically initialized with the properties that it was configured with using Interface Builder. This part of the initialization process uses any **set***Variable***:** methods that are available (where *variable* is the name of an instance variable whose value was set in Interface Builder). Finally, after all the objects are fully initialized, each receives an **awakeFromNib** message.

The order in which objects are loaded from the archive is not guaranteed. Therefore, it's possible for a **set***Variable***:** message to be sent to an object before its companion objects have been unarchived. For this reason, **set***Variable***:** methods should not send messages to other objects in the archive. However, messages to other objects can safely be sent from within **awakeFromNib**—by which time it's assured that all the objects are unarchived and initialized (though not necessarily awakened, of course).

Typically, **awakeFromNib** is implemented for classes whose instances are used as the owners of a loaded nib file (shown as "File's Owner" in Interface Builder). Such a class has the express purpose of connecting the loaded objects with objects in the application, and can thereafter be disposed of, or remain in the capacity of a controller or coordinator for the loaded objects. For example, suppose that a nib file contains two custom views that must be positioned relative to each other at run time. Trying to position them when either one of the views is initialized (in **initWithCoder:** or a **set***Variable***:** method) might fail, since the other views might not be unarchived and initialized yet. However, it can be done in the nib file owner's **awakeFromNib** method (**firstView** and **secondView** are outlets of the file's owner):

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

    if ([[self superclass] instancesRespondToSelector:@selector(awakeFromNib)]) {
        [super awakeFromNib];
    }
    viewFrame = [firstView frame];
    viewFrame.origin.x += viewFrame.size.width;
    [secondView setFrame:viewFrame];
    return;
}
```

Note the testing of the superclass before invoking its implementation of **awakeFromNib**. The Application Kit declares a prototype for this method, but doesn't implement it. Because there's no default implementation of **awakeFromNib**, be sure to invoke it only when the object does in fact respond.

See also: + **loadNibNamed:owner:** (NSBundle Additions),
– **awakeAfterUsingCoder** (NSObject class of the Foundation Kit),
– **initWithCoder:** (NSCoding protocol of the Foundation Kit)
+ **initialize** (NSObject class of the Foundation Kit)