awakeFromNib

Implemented to prepare an object for service after it has been loaded from an Interface Builder archiveÐa so-called ªnib file.º 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.

When loadNibFile:owner: or a related method loads an Interface Builder archive into an application, each custom object from the archive is first initialized with an init message (initFrame: if the object is a kind of View). It's then more specifically initialized with the properties that it was programmed to have in Interface Builder. This part of the initialization process uses any setVariable: methods that are available (where variable is the name of an instance variable). Finally, after all the objects are fully initialized, they each receive an awakeFromNib message.

The order in which objects are loaded from the archive is not guaranteed. Therefore, it's possible for a setVariable: message to be sent to an object before its companion objects have been unarchived. For this reason, setVariable: methods should not send messages to other objects in the archive. However, messages to other objects can safely be sent from within awakeFromNibÐwhen it's assured that that all the objects are unarchived and fully initialized.

Typically, awakeFromNib is implemented for only one object in the archive, the controlling or ªownerº object for the other objects that are archived with it. For example, suppose that a nib file contained two 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 a setVariable: method) might fail, since the other View might not be unarchived and initialized yet. However, it can be done in an awakeFromNib method:

loadNibFile:owner:withNames:fromZone: (Application class)