destination

Implement to return the object that is the destination of the connection.

source

establishConnection

Implement to connect the source and destination objects. Interface Builder sends this message to each connector object after all objects have been unarchived from the nib file.

free

Implement to release the storage for the connector object.

nibInstantiate

Implement to verify the identities of the connector's source and destination objects.

Interface Builder sends a nibInstantiate message to a connector object to give it an opportunity to verify that its source and destination instance variables point to the intended objects. For example, consider the case in which a user puts a CustomView in a window and then reassigns the CustomView's class to MyView. The MyView class has a textfield outlet that the user connects to a neighboring TextField object. This action causes Interface Builder to create a connector object and set its destination to the TextField and its source to the CustomView. (The source can't be set to the MyView object since that class doesn't exist in InterfaceBuilderÐthat's why the CustomView was used in the first place.)

When the resulting nib file is loaded in the finished application, the connector object is unarchived and sent a nibInstantiate message. It's at this point that the connector must reset its source instance variable from the CustomView object to the MyView object.

The Application Kit, in a category of Object, provides a default implementation of this method. This implementation returns self. (Please note that the method isn't publically declared, a problem that will be remedied in a later release.) Consequently, all objects can respond to a nibInstantiate message. Your connector, therefore, should minimally implement this method to send nibInstantiate messages to its source and destination objects. For example, assuming the outlets are named theSource and theDestination, the implementation is:

renewObject:old to:new

Implement to update a connector by replacing its old source or destination object (old) with a new
used by Interface Builder, for example, when a user drags a Button object into a Matrix of ButtonC
the Button was connected, the connection information must be updated to reflect that fact that the
replaced by a ButtonCell. Interface Builder updates this information by sending the appropriate co
renewObject:to: message with the Button as old and the ButtonCell as new.

source

Implement to return the object that is the source of the connection.

destination

write:(NXTypedStream *)stream

Implement to archive the connector object to stream.

read