Q: How can I make connections in InterfaceBuilder between objects in different nib files, or in the same one loaded multiple times?

A: You can't. You need to make the association in your code. The object that is the nib file's owner should take care of communicating with any other objects (including the owners of other nib files). For example, the file's owner can have methods that return instance variables referring to any objects from the nib file that you want other objects to be able to talk to. (You must have created outlets in Interface Builder for these objects, in order for the objects to be assigned to instance variables.) The file's owner can also set the delegate of an object it owns to be some external object.

It's conceptually impossible to make the connections in Interface Builder, because there's no way to know which *instance* of an object from one nib file is connected to an instance of

some object from another nib file. Each nib file is loaded at a different time during the program execution, and any nib file can get loaded multiple times. Only your code "knows" when the various objects get createdÐnamely, when you call **loadNibSection:owner:**. Remember that nib files are just templates; the objects are unarchived when you load the nib file. When you make connections in InterfaceBuilder, you're really just making the template of these connections. Only when the nib file is loaded do the objects and their interconnections get instantiated, so to speak. If the other nib file hasn't been loaded yet, you can't connect to its objects, since they don't exist in your program yet. That's why connections are limited to nib files. It's also one reason every nib file must have an owner: so that objects from that nib file can communicate with other objects in your program.

The same logic applies to loading the same nib file multiple times. A typical use of loading the same nib file multiple times is in applications that have the notion of a document. The

document is described by a nib file that gets loaded each time you want a new document (usually when the user clicks on the "New" menu). There should generally be a new instance of the owner of the nib file for each time you want to load the same nib file. The only way to communicate between the objects in different documents is through their owners.

One example of multiply loaded nib files is /NextDeveloper/Examples/SoundAndMusic/SoundKit/SoundEditor. In this application, SoundController is the main object that loads in the application's main nib file. It also is responsible for creating SoundDocuments, each of which loads in its own nib data. In other words, a new instance of the SoundDocument class is the file's owner for each "instance" of SoundDocument.nibÐor more precisely, for the objects collectively unarchived each time

```
[NXApp loadNibSection:SoundDocument.nib owner:self];
```

is called in SoundDocument's **init** method. Although it wasn't necessary, there could have been methods in SoundDocument to return the objects "inside" each SoundDocument's nib file. For example, SoundDocument could have implemented a method called **-soundView** that returned the variable **mySoundView**, which is an InterfaceBuilder outlet. This way any other objects could have talked to the SoundView, by messaging SoundController to get the current document (with the **document** method), and then messaging the current document to get its SoundView (with the **soundView** method).

See ../Development_Environment/InterfaceBuilder_file_owner.rtf for more on the concept of a file's owner.

QA699

Valid for 1.0, 2.0, 3.0