Interface Builder Release Notes

This file contains release notes for the version of Interface Builder (release number 457) distributed with the Enterprise Objects Framework.   Release notes for earlier versions are at the end of this file.

New Features

This release of Interface Builder supports these new features:

New Palette Features

·   Interface Builder now supports the creation of dynamic palettes. Dynamic palettes are a convenient way to store frequently-used objects that have no code files

associated with them.

- Use the New command on the Palettes submenu in the Tools menu to create a dynamic palette.   Use the Save command on the same menu to save a dynamic palette.
- Copy any view object from a nib file to a dynamic palette by holding down the Alternate key and dragging it onto the palette.
- Remove objects from the dynamic palette by Alternate-dragging them off of the palette.
- Create an icon for the palette by dragging a TIFF file from the Workspace Manager's File Viewer onto the palette.

· There is a new palette for Enterprise Objects Framework, **/NextDeveloper/Palettes/EOPalette.palette**.

· Interface Builder no longer automatically adds the palettes from the **/NextDeveloper/Palettes** directory to the Palettes Preferences panel. You can add these palettes, or any other palettes, to the Preferences panel by dragging them from the Workspace Manager's File Viewer onto the panel. To remove a palette from the Preferences panel, hold down the Alternate key and drag the icon off of the panel.

· There is now a Palettes submenu in the Tools menu.   The Open... command on

this submenu loads palettes. The Palettes... command displays the palettes viewer if the viewer is currently closed.   The rest of the commands in this submenu are involved in creating dynamic palettes.

·   Palette dependency information is now stored in the nib file.   If you open a nib file that depends on a palette which isn't loaded into Interface Builder, Interface Builder will load the palette automatically.   This applies only to nib files created with this version of Interface Builder or nib files that you have previously modified with this version of Interface Builder. If you haven't modified the nib file with this version yet and the nib file depends on a palette that isn't loaded, Interface Builder displays a message telling you why it can't open the nib file.

Modified Nib File Window

·   The suitcases in the nib file window are replaced by a tab list. Click a tab or drag the mouse onto a tab to display the corresponding nib file view. The Instances view now has two modes: icon and outline. The icon mode is the same as the Instances view in previous versions. You now perform all steps involved in subclassing entirely within the Classes view of the nib file. The Class Inspector panel   no longer exists.

New Outline Mode for Nib File Window

- The Instances view of the nib file window now has an outline mode. Outline mode displays the hierarchy of objects in the interface.   You can cut, copy, paste, and connect objects in outline mode the same way you perform these tasks in icon mode.

- A filled button to the left of an object name represents an object that owns other objects. Click the button to display the objects that object owns. Click the button again to hide the objects. Hold down the Command key while clicking to open the entire hierarchy under the object.   Also, the right and left arrow keys open and close hierarchies of objects, respectively.

- Double-clicking an object in outline mode selects it in the user interface.

- There are two arrows, connection-out and connection-in, to the left of each object. Click the connection-out arrow (the left arrow) to see   connections from this object to other objects. Click the connection-in arrow (the right arrow) to see connections to this object from other objects. Each arrow can show both outlet and action connections. Click the arrow again to hide the connections.

- You can disconnect two objects by displaying the connection, holding down the Control key, and clicking the line between the objects. The cursor changes to a pair of scissors to notify you that you may cut the connection.

- To see exactly which object in the user interface is represented by a line in the outline mode display, hold down the Control and Shift keys and click the line in the outline mode display. A large arrow points to the object in the interface.   You also can select the object in the interface and press Command-e. The line representing the object is selected in the outline mode display.

- You can Alternate-drag objects from outline mode onto a dynamic palette. You can drag more than one object at a time as long as all of the selected objects are on the top level of the hierarchy except for the object directly under the cursor. The object directly under the cursor does not have to be in the top level. Copying more than one object at a time preserves any connections between two objects that are part of the selection. This function allows you to save the following items to a dynamic palette:

  - Any object and its delegate
  - A menu item and the panel it opens (for example, the Info... command and its panel)
  - Objects that are not explicitly part of the interface that are connected together

Modified Classes View for Nib File Window

- The Classes view in the nib file window now displays classes as a class

hierarchy. The buttons to the left of the class name work the same way as the buttons in the outline mode of the Instances view.

· There is no longer a Find field, but you can type the first few characters of a class name to select the class in the display.

· The right side of the display shows you how many outlets and actions each class has. The left icon is for outlets, and the right icon is for actions. You can click either of these icons to display the outlet and action names underneath the class.

· To subclass a class, select the class and choose Subclass from the Operations pull-down list. You now change the name of the new class and add outlets and actions to it within the Classes view instead of going to the Class Inspector panel.

- To change the name, select it, edit it, and press Return.
- To add outlets and actions, choose Edit Class from the Operations pull-down list, select either the word Outlets or the word Actions, and choose Add Outlet or Add Action from the Operations pull-down list. Then edit the outlet or action name and press return.
- As a shortcut, if you press Return while an outlet is selected, a new outlet is created. Similarly, if you press Return while an action is selected, a new action is created.

- If the Classes view of the nib file window is active, it displays the class of object currently selected in the user interface.

New Formatting Features

- You can drag sound and image files directly onto an object from the Workspace Manager's File Viewer. Interface Builder prompts you to add the file to either the project or the nib file if the sound or image is not already included in your project.

- The arrow keys now move objects. Pressing an arrow key moves an object one pixel if the grid is disabled. If the grid is enabled, the object moves to the next grid mark.

- There is a new menu item, Group In SplitView, under Group in the Format menu. This item groups the selected objects into a SplitView object.

- To copy an object from one window to another window in the same nib file without breaking the object's connections, hold down the Alternate key while dragging the object. You can also copy the object, minus its connections, to a window in a different nib file by holding down the Alternate key while dragging to the window. Alternate-dragging will automatically copy any images, sounds, and class definitions associated with the object to the nib file if the nib file does not already

have them.

· The default main menu and the Edit menu now match the menus in the *NEXTSTEP User Interface Guidelines*. You get these menus only if you use the New Application command in Interface Builder to create the nib file. If you use the default file created in Project Builder, you will get the old menus.

· In addition to \r (Carriage Return), you can use the following new escape sequences to specify key equivalents for buttons:

\b Backspace
\d Delete
\e Esc
\t Tab

Improved File Permissions Support

· When saving a modified nib file, the file permissions are preserved.

New Documentation

This version of Interface Builder comes with a new manual and new on-line help:

· Taking a task-based approach, the manual *Working With Interface Builder* describes how to use Interface Builder when you develop NEXTSTEP applications and explains some of the concepts underlying this application. It also steps you through some of the basic programming tasks in NEXTSTEP development, such as implementing a subclass. *Working With Interface Builder* is a beta version of a future manual that will cover all NEXTSTEP development tools.

 The book comes in both printed and on-line forms; the on-line version is in **/NextLibrary/Documentation/NextDev/DevTools/NewInterfaceBuilder** and can be accessed through Digital Librarian's Development Tools target.

· Interface Builder on-line help has been greatly expanded. You can access the new help by choosing Help from the Info menu. Help now contains a new Power Tips topic, which lists some shortcuts that Interface Builder gives you for performing certain tasks. In addition, this edition of help makes more extensive use of hyperlinks than previous versions.

NeXT's Developer Publications is very interested in what you think of the manual

and of on-line help. Please send your comments and suggestions to **techpubs_feedback@next.com**.

Problems Fixed in This Release

These problems have been fixed in this release of Interface Builder:

Reference: 22990, 27815

Problem: Interface Builder doesn't allow you to remove palettes from the Palettes Preferences panel.

Description: In the Palettes Preferences panel, select a custom palette, hold down the Alternate key, and drag the palette icon off the panel.

Reference: 25912

Problem: Accidentally typing Command-r (or keyboard equivalents other than Command-c, Command-x, and Command-v) while the Save Panel is open causes Interface Builder to hang.

Description: Open the Save Panel by choosing Save As from the Document menu.

Type Command-r to enter Test Interface mode.   Return from Test Interface and Interface Builder hangs.

Reference: 28301

Problem: Inspector panel doesn't show attributes for images or sounds.

Description: Choose the Images or Sounds view in the nib file window.   Select an image or sound, and display the Inspector panel.   The Inspector panel say ªempty selection.º

Reference: 28553

Problem: Interface Builder crashes after permitting you to open non-nib files if you type the file name in the Open Panel rather than selecting the name.

Description: In the Open Panel, type the name of any file.   Interface Builder allows you to open this file, resulting in a more or less blank nib file.   If you try to close this file, Interface Builder crashes.

Reference: 29601

Problem: Interface Builder thinks a nib file has changes when you select a form.

Description: Add a form object to your nib file's main window object.   Save the nib file, bring up the Inspector Panel, and select the form object.   The nib file window indicates that changes have been made.

Reference: 30509

Problem: Can't implement help in subprojects.

Description: Interface Builder doesn't go up the directory hierarchy to find the project's Help directory, so it's impossible to make connections from the project Help files to a subproject's nib file.

Reference: 30892

Problem: Interface Builder doesn't allow you to override its own keyboard equivalents.

Description: If you define a keyboard equivalent that conflicts with one of Interface Builder's, Interface Builder overrides your equivalent.

Reference:	31717
Problem:	Interface Builder crashes if you try to parse multiple header files at the same time.
Description:	In the Classes display of the File window, choose Parse from the Operations pop-up list. If you choose more than one header file in the Open panel that appears and click OK, Interface Builder crashes.

Reference:	32990
Problem:	Library sounds aren't accepted by the Sounds suitcase.
Description:	Few of the sounds found in the well-known locations (**~/Library/Sounds**, **/LocalLibrary/Sounds**, **/NextLibrary/Sounds**) show up in Interface Builder's Sounds suitcase. If you drag a sound from one of these locations into Interface Builder's File window, it won't be accepted. Instead, Interface Builder displays a panel saying that the sound already exists.

Reference:	33245
Problem:	If you delete all the menu items from an application's main menu, you can no longer add new items.

Description: After deleting the last menu item from an application's main menu, you can't add any new ones.   Then, if you close the main menu window, double-clicking its icon in the File window fails to bring it back.

Reference: 35938

Problem: Interface Builder uncompresses compressed TIFF files.

Description: If you have compressed image files stored in your nib file, Interface Builder uncompresses them when it saves the nib file.

Reference: 37681

Problem: Text in Matrix can't be set selectable using ªmatch prototype.º

Description: Set the prototype for a Matrix of TextFieldCells in the Inspector panel to ªnot editable, selectable.º   Click ªMatch prototypeº in the Inspector panel. The cells are set to ªnot editable, not selectable.º

Reference: 39629

Problem: Remove from StringTables doesn't mark nib file as edited.

Description: If you remove an entry from a StringTable in a nib file with no other changes, the nib file still reflects that no changes have been made.

Reference: 39664

Problem: NaN in Inspector panel.

Description: Create a matrix of sliders, edit the prototype, and set the current value to something.   Match the prototype, and select one of the sliders in the matrix.   The minimum value is now NaN.

Reference: 44839

Problem: Interface Builder sorts attributes differently than EOModeler.

Description: If you have an existing entity defined in the EOModeller and you add a new attribute, after updating the display, the new attribute appears in alphabetical order in EOModeller.   However, the new attribute appears at the *bottom* of the list in the EOController inspector in Interface builder.

Fix: Interface Builder and EOModeler now display attributes sorted alphabetically.

Known Problems

These problems exist in this release of Interface Builder:

· Interface Builder defaults are not written by the standard mechanism. If you have any problems with the defaults (for example, which palettes are loaded at startup) and want to get rid of them, just edit or remove the file **~/.NeXT/defaults.nibd.**

· Interface Builder does not allow you to change the size of the docView when editing a ScrollView.

· The context-sensitive help links don't work for anything in the palettes window, and they don't work in the ScrollView Attributes display.

· Model files copied from a mail message are incorrectly referenced.   In Interface Builder, if you drag a model file in from a mail message, it turns into an entity controller.   Don't make a copy, and save the nib file.   Then drag the model file from the mail message into **~/Library/Models**.   Go back into the nib file and delete the combined controller/data source object.   Then drag the model from **~/Library/Models** into Interface Builder.   The resulting entity controller still

references the model file in your mailbox.

49811

Problem: Interface Builder sometimes doesn't notice changes made in EOModeler.

Description: The following procedure illustrates the problem:

1. Create a project.
2. Open the project's nib file and change file's owner.
3. Create a model and save it in **~/Library/Models**.
4. Drag an entity into the nib file.
5. Add the model to the project.
6. Double-click the controller in the nib file to open the model.
7. Add a relationship to the model, then save it.
8. Go back to the nib file; the relationship doesn't show up, and Interface Builder hasn't marked the document as having changed.

Workaround: Quit and restart Interface Builder, then re-open the offending nib file.

Reference: 50215

Problem: Interface Builder crashes when you reference a model with unknown

adaptor.

Description: In Interface Builder, if you attempt to work with a model that references an adaptor that isn't installed, Interface Builder will crash.

Workaround: Rebuild your model so that it uses an existing adaptor. This solution is particularly important if you are changing from the Oracle to the Oracle7 adaptor, since it will cause your application to run faster.

Because it takes some effort to rebuild a model with a new adaptor, there are two ways that you can temporarily work around the problem in the Oracle to Oracle7 case. Note, however, that neither of these solutions results in improved performance, so rebuilding your model is the preferred solution. These two alternate solutions are:

· Using Edit, change the adaptor name in the model file from "Oracle" to "Oracle7"

· Create a symbolic link so that the Oracle adaptor appears to exist. The following commands illustrate how to do this:

```
cd /NextLibrary/Adaptors
ln -s Oracle7.dbadaptor Oracle.dbadaptor
cd Oracle7.dbadaptor
ln -s Oracle7 Oracle
```

Notes Specific to Release 3.2

There are no changes to note for Release 3.2.

Notes Specific to Release 3.1

Bugs Fixed in Release 3.1

These bugs have been fixed in Release 3.1:

Reference
28510

Problem
Interface Builder can lose target/action connection information for a custom object if the object's class is reassigned.

Description
If you reassign the class of a custom object to a subclass of its former class, connection information can be lost. For example, if you

reassign the class of a custom View from GraphView to ColorGraphView (a subclass of GraphView), a Button whose target had previously been the GraphView object becomes disconnected.

Reference: 29773

Problem: Invoking **registerDocumentController:** during Interface Builder's launch fails.

Description: If a custom palette is loaded as part of Interface Builder's launch and some object within the palette attempts to register itself as a document controller, the registration fails.   This is because the List object that would store the document controller isn't created until after the Palettes window appears.

Reference: 30400

Problem: Resizing Views is slow.

Description: Resizing Views within a window of an application under construction is very slow.   The larger the View, the slower the resizing.

Notes Specific to Release 3.0

These notes were included with the Release 3.0 version of Interface Builder. Sections that are no longer relevant have been marked with an italicized comment.

Interface Builder has been fully rewritten for release 3.0.   Here are the highlights:

· Project management has been removed from Interface Builder.   It has been moved to ProjectBuilder, which is installed in **/NextDeveloper/Apps**.   Project Builder creates projects for you, creating default project files (a makefile, a main file, and so on), and lets you access other development tools such as Edit and Interface Builder.

Given this new arrangement, it's best to start a new application by creating a project in Project Builder, which provides you with a default nib file for the type of project you are creating.   You can then open this nib file and make the changes appropriate for your application.

· The nib file format has changed.   Nib files are no longer files but file packages. **Important:** Nib files created with the new Interface Builder can no longer be

included in Mach-0 segments of an executable file.

· The format for loadable palettes has changed.   The old format is no longer
  supported.

· Most inspectors don't have OK or Revert buttons.

· The File window has changed. It now has a shelf with suitcases corresponding the
  object, image, sound, and class resources in your application.

· Interface Builder now supports the NEXTSTEP help system.   Once a Help
  directory has been added to your project using Project Builder, Interface Builder's
  Help Builder panel will let you view help files and connect user-interface objects to
  specific pieces of that text.

· Inspector categories have been changed.   They now consist of Attributes,
  Connections, Size, and Help.

The process of making custom palettes has changed for the 3.0 release.   In addition,
you can now make custom palettes for non-View objectsÐWindows, Panels,

MenuCellsÐas well as for objects that are instantiated by being dragged into the File window.   The tutorial in **/NextLibrary/Documentation/NextDev/DevTools/18_CustomPalette** has been updated to describe the new process for creating palettes of View objects.   For more information on creating other types of palettes, see the outlines below and refer to **/NextLibrary/Documentation/NextDev/GeneralRef/08_InterfaceBuilder**.

How to Make a Palette of Views

· Open ProjectBuilder
· Choose the Project/New... command
· in the SavePanel set the name of your palette project to 'MyPalette' for instance.
· in the SavePanel use the popup to set the ProjectType to 'Palette'.
· hit OK
· a new palette project has been created with one nib file (MyPalette.nib) and one class file (MyPalette.[hm])
· add your extra files in the project
· open MyPalette.nib and add the views you want in the palette
· open the ascii file palette.table:

```
Class = MyPalette;                    /* (a subclass of IBPalette) */
NibFile = MyPalette;                  /* (a nib file name) */
/* Icon =;                           (a tiff/eps file name) */
```

```
/* ExportClasses = ();                        (a list of class names) */
/* ExportIcons = ();                          (a list of icon names) */
/* ExportSound = ();                          (a list of sound names) */
```

· to get a custom icon for your palette, add a tiff file MyPaletteIcon.tiff in your project
  images (it should be smaller than 48x48 and have appropriate alpha in order to fit in a
  48x48 bordered button). Then change the 3rd line of palette.table to:

```
...
Icon = MyPaletteIcon;
...
```

· if you want to have some custom classes (resp. images, resp. sounds) that you added
  to your palette nib file,   to be accessible to the palette users, edit the appropriate lines
  in palette.table:

```
...
ExportClasses = (MyFirstClass, MySecondClass);
...
```

· build

How to Make a Palette of Objects/Menus/Windows
  · Proceed as above
  · The second step is now to associate a view in the palette view to some
    object/menucell/window. If you want to do it for a window for instance
  · Create another window in MyPalette.nib
  · Create 2 new instance variables for the class MyPalette 'windowView' and
    'windowData' for instance

· Connect windowView to the View in the palette that will represent the window (that the user will drag and drop
· Connect windowData to the new window you created
· Implement the following method for MyPalette
   - finishInstantiate {
[self associateObject:windowData type:IBWindowPboardType with:windowView];
return self;
   }
· For objects (that the user will deposit in the file window) use the IBObjectPboardType constant, for MenuCells without submenu use IBMenuCellPboardType, for MenuCells with submenu use IBMenuPboardType

How to Make an Inspector
· implement your own IBInspector subclass:

```
#import <apps/InterfaceBuilder.h>
@interface MyInspector:IBInspector <IBInspectors> {
....
}
@end

@implementation MyInspector
- (BOOL)wantsButtons { return YES/NO; }
- revert:sender {
    ....
    return [super revert:sender];
```

```
        }
        - ok:sender {

            ....
            return [super ok:sender];
        }
        @end
```

· define getInspectorClassName for your own class:

```
        - (const char *)getInspectorClassName { return "MyInspector"; }
```


Known Problems

· InterfaceBuilder defaults are not written by the standard mechanism. If you get any problems with the defaults (for example, which palettes are loaded at startup) and want to get rid of them just remove the file **~/.Next/defaults.nibd**

· Copy/paste is not working for the Class Browser.

· Copy/paste is not working for the Class inspector.

· IB is unpredictable when going into Test mode if there is a target/action connection to a custom subclass of   Control.

· Cannot connect (or select) a custom view that is grouped within a scrollview.   You'll need to use Ungroup to make it accessible.

· Unable to create matrix with custom subclass of control.