

5

Using Dynamic Palettes

Creating and saving dynamic palettes

Storing view objects on dynamic palettes

Arranging objects on dynamic palettes

Storing top-level objects on dynamic palettes

Putting connected view and top-level objects on a dynamic palette

Managing palettes

Who hath not seen thee oft amid thy store?

Sometimes whoever seeks abroad may find
Thee sitting careless on a granary floor,
Thy hair soft-lifted by the winnowing wind...

John Keats, from *To Autumn*

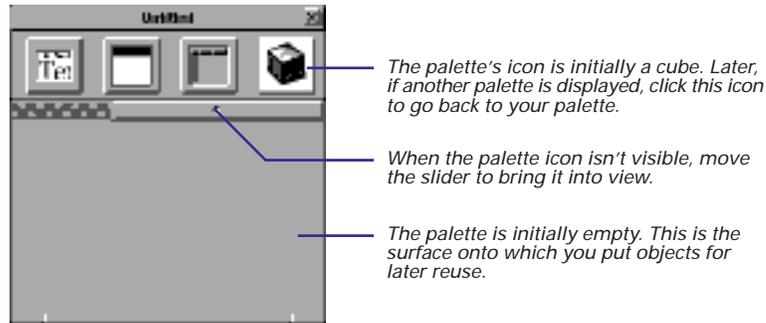
The superfluous is very necessary.

Voltaire

Creating and saving dynamic palettes

- ▶ To create a palette, choose **Tools ▶ Palettes ▶ New**.
- ▶ To save a palette, choose **Tools ▶ Palettes ▶ Save**.

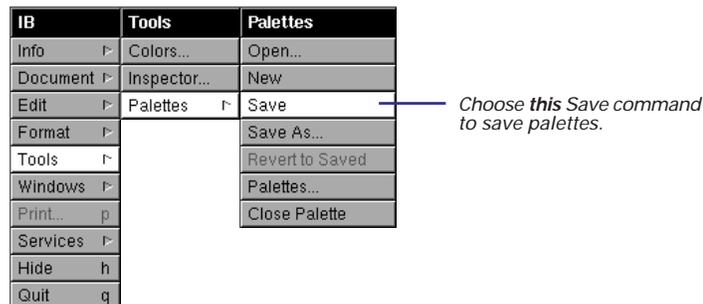
When you create a dynamic palette, an empty palette appears in the Palette window.



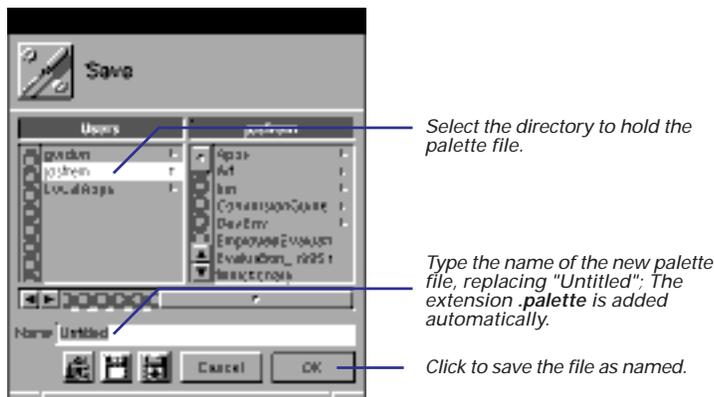
You can customize the icon for your dynamic palette. The task “Managing palettes” in this chapter tells you how to do this and also describes how to unload palettes in Interface Builder.

As with the standard Application Kit palettes, you can choose an existing dynamic palette by clicking its icon in the Palette window (when created, dynamic palettes have the generic cube icon). To use an object on a dynamic palette, follow the same procedure as for objects on the standard palettes: Drag the object from the palette and drop it onto an appropriate surface.

You must save your dynamic palettes. If you do not save a palette after you create it, you lose it when you quit Interface Builder. (Interface Builder prompts you if you try to quit without saving a palette.) Choose the Save command to save dynamic palettes, *but the Save command from the Palettes menu, not the Document menu*.



Interface Builder brings up the Save panel, allowing you to designate a name for the palette.



Tools for Interface Crafters: Static and Dynamic Palettes

A palette is a special display that holds one or more reusable objects. You can drag these objects from the palette to your application's interface. There are two types of palettes: static and dynamic. To the user, they seem identical, but the differences are many.

Static palettes are built as a project and have code defining their objects; dynamic palettes include no special code—they're unique configurations of (mostly) standard OpenStep objects. Consequently, static palettes must be compiled, but you can create dynamic palettes on the fly, without writing and compiling code. Objects on static palettes can have inspectors and editors, which dynamic-palette objects cannot have.

Creating static palettes (and their inspectors and editors) is a more complex process than creating dynamic palettes, but the resulting product has more value added to it. For example, if you want to store a button that has the title OK, you use a dynamic palette because the change involves only the Interface Builder Inspector panel. However, if you want to store a custom subclass of `NSButton`, you use a static palette. A static palette can store both the button and your custom code.

Dynamic palettes are a great convenience. You can save collections of your objects, with or without their interconnections,

to a dynamic palette at any time. You can save dynamic palettes and store them in the file system, just as you do with the traditional compiled palette. You can remove the palette from the Palette window and, when you need it again, just load it back into Interface Builder.

The possible practical uses of dynamic palettes are numerous. You can use them to:

- Store collections of often-used view objects configured with specific sizes and other attributes.
- Hold windows and panels that are replicated in your projects (such as Info panels).
- Store versions of interfaces.
- Keep interconnected objects as a template that you can later use as-is or modify for particular circumstances. For instance, you could store a group of text fields and their delegate, or a set of controls and their connections to a controller object.

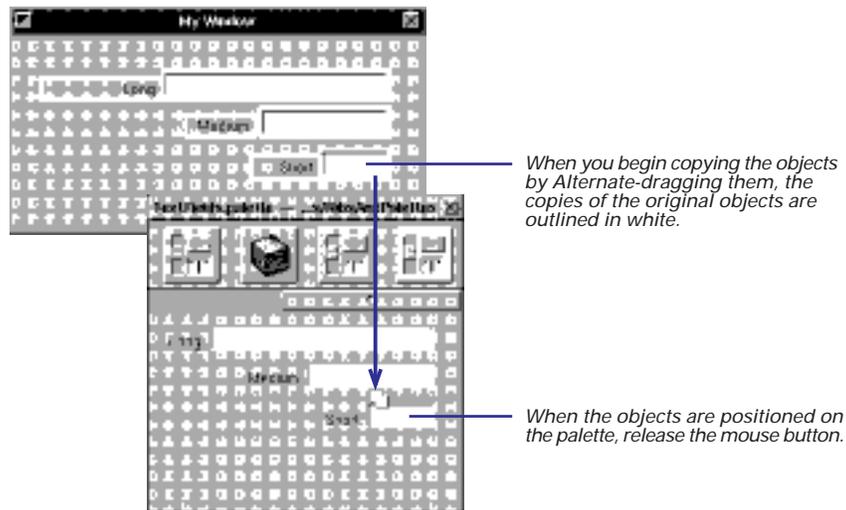
You can also use dynamic palettes for prototyping and group work. For instance, you could design an interface or a part of an interface, store the objects on a dynamic palette, and then mail the palette file to all interested parties.

Storing view objects on dynamic palettes

- 1 **Configure one or more view objects.**
- 2 **Select the objects.**
- 3 **Alternate-drag them to the palette.**

You can save any single view object or group of view objects to a dynamic palette so that you can use them again. The objects stored in this manner preserve the size and other attributes they have when you store them.

First, size each object to be stored and, through the Inspector panel, set its important attributes. If you are storing a group of objects, such as a set of controls within a box, make sure to position all objects in proper relation to each other.



When you begin copying the objects by Alternate-dragging them, the copies of the original objects are outlined in white.

When the objects are positioned on the palette, release the mouse button.

If there are several objects you want to store on a palette, you can drag each object onto the palette individually, or you can make a multiple selection in the interface. (Draw a selection rectangle around the objects, or Shift-click the objects in succession.) If you store a group of objects that are connected—say the three fields above were grouped in a box and connected through the **nextKeyView** outlet—their connections are copied also.

Tip: You can also copy view objects to a dynamic palette from the outline mode of the Instances display.

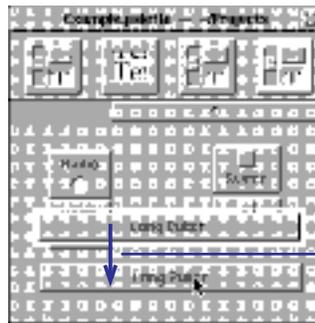
Once an object is on a palette, you can move it around the palette or remove it from the palette. See the next task, “Arranging objects on dynamic palettes” for details.

You can also store view objects that are connected to top-level objects. See “Storing top-level objects on dynamic palettes” for more information.

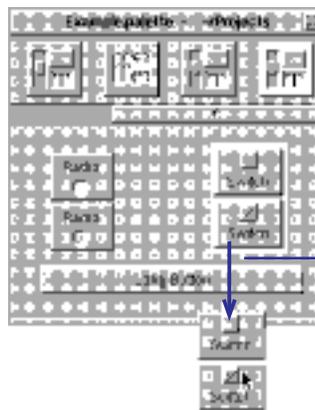
Arranging objects on dynamic palettes

- ▶ To position a palette object, Alternate-drag it within the palette.
- ▶ To remove a palette object, Alternate-drag it off the palette.

You can do two things to objects once they're on a dynamic palette: move them around the palette and delete them from the palette.



To reposition an object, Alternate-drag it to its new location on the palette.



To remove an object, Alternate-drag it until it is off the palette, then release the mouse button.

If you store an object on a dynamic palette and later discover that its size, connections, or attributes must be changed, you must:

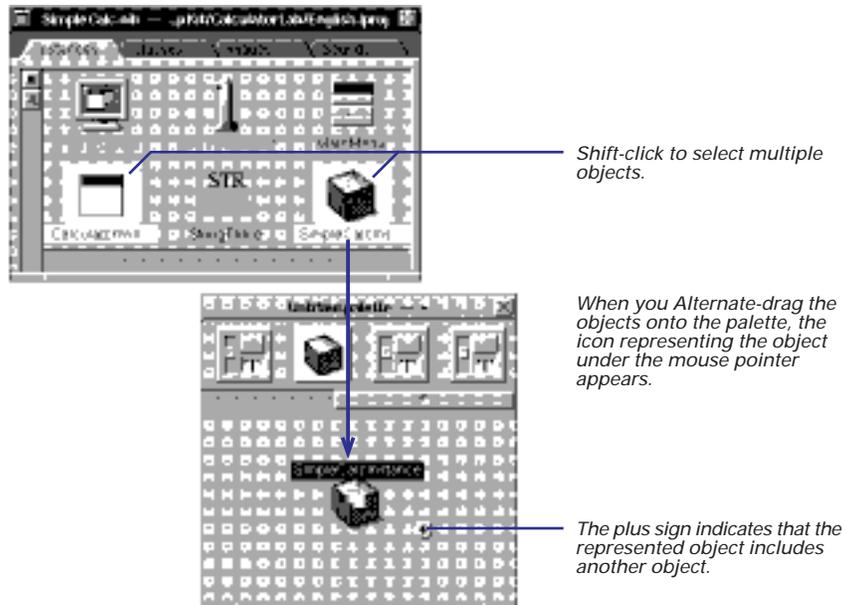
- Remove it from the palette.
- Resize it, reconnect it, or reset its attributes.
- Store it on the palette again (by Alternate-dragging it).

Storing top-level objects on dynamic palettes

- 1 In the Instances display, select one or more windows, panels, or custom objects.
- 2 Alternate-drag them and drop them on a dynamic palette.

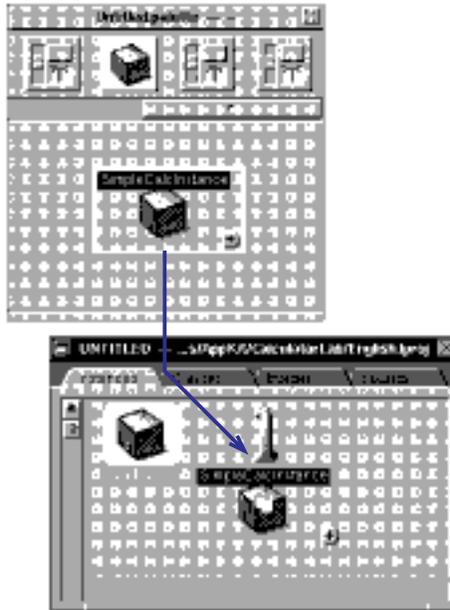
You can put custom objects, windows, and panels on dynamic palettes and reuse them again and again. You can store these top-level objects individually or as connected sets of objects. When you select a controller object and a window and store them together, the connections between them are also stored on the dynamic palette. In addition, all connections between a window or panel and its views are preserved as well as the connections among the views themselves.

To store a single top-level object on a dynamic palette, Alternate-drag it from the Instances view of the nib file window and drop it onto the palette. To store multiple, connected objects, make sure they're selected as a group first.



Tip: You can perform this same task whether in the outline or the icon mode of the Instances display.

When you drag the object or objects from the dynamic palette to add them to another nib file (or to duplicate them in the same nib file), make sure that the “surface” on which you drop the object (as represented by the icon) is compatible.



To reuse the connected objects, drag the object and drop it over a suitable surface.

You'll know which surface is compatible by the icon representing the object. If it's a cube (custom object), you must drop it over the nib file window. If it's a window or panel, you can drop it anywhere on the screen, including over the nib file window.

Putting connected view and top-level objects on a dynamic palette

- 1 In outline mode, select a connected top-level object and a view object.
- 2 Alternate-drag the objects to the palette.

There might be situations when you don't want to store an entire window with the custom object that manages that window's views. You just want to store the custom object and some of the window's views, or you want to store the window and only some of its views. You can do this from Instances view outline mode.

For example, here's how three slider objects hooked up to a controller object (ImageController) look in outline mode when the connections are displayed.



Select the top-level object and view object by Shift-clicking them. You can select only one view object and one top-level object.



Select a top-level object and a view object connected to it. (In this case, the view is a box containing several subviews.)

When you Alternate-drag the selected objects onto the dynamic palette, a representation of the object under the mouse pointer appears.

The plus-sign icon indicates that this palette object contains multiple objects, including all their connections with each other.

Managing palettes

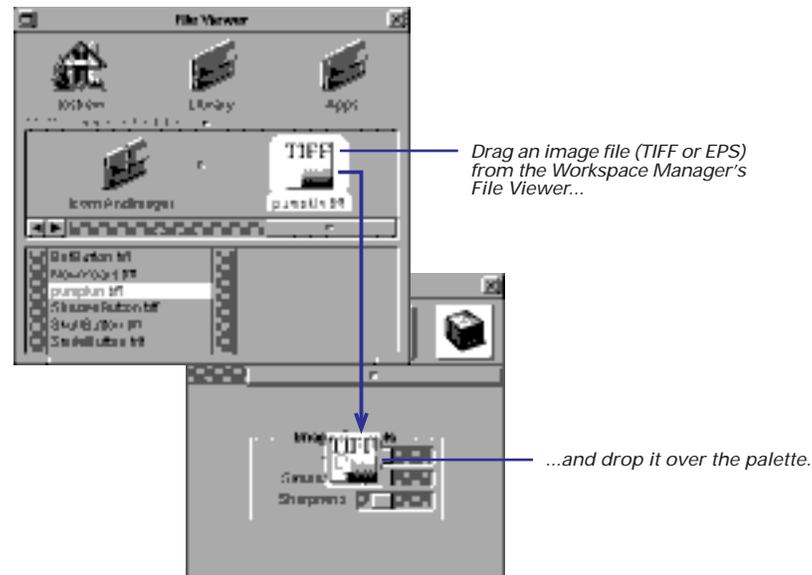
- ▶ **Customize the palette icon:**
Drag and drop an image file over the palette.
- ▶ **Install and uninstall palettes:**
Double-click a palette icon in the Palettes display of the Preferences panel.
- ▶ **Load a palette:**
Choose Tools ▶ Palettes ▶ Open and select a palette file.
- ▶ **Unload a palette:**
Alternate-drag a palette icon off the Palettes display of the Preferences panel.

Interface Builder gives you facilities for managing static and dynamic palettes. These management functions include:

- Customizing the palette icon (dynamic palettes only)
- Installing and uninstalling palettes on the Palette window
- Loading and unloading palettes

Customizing the Palette Icon

For dynamic palettes, Interface Builder gives you the option of specifying the icon that identifies the palette in the Palette window.

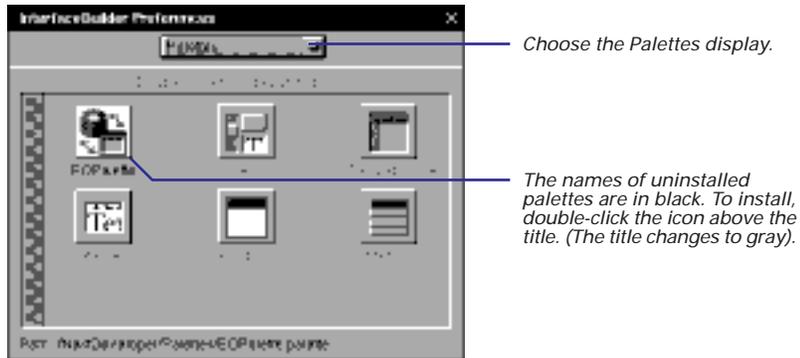


Tip: Since Interface Builder scales the image to fit into the icon rectangle, create or choose a TIFF or EPS image that is 35 X 35 pixels.

You can also customize the icons of static palettes, but you must do this programmatically, specifying the image file in a `palette.table` file.

Installing and Uninstalling Palettes

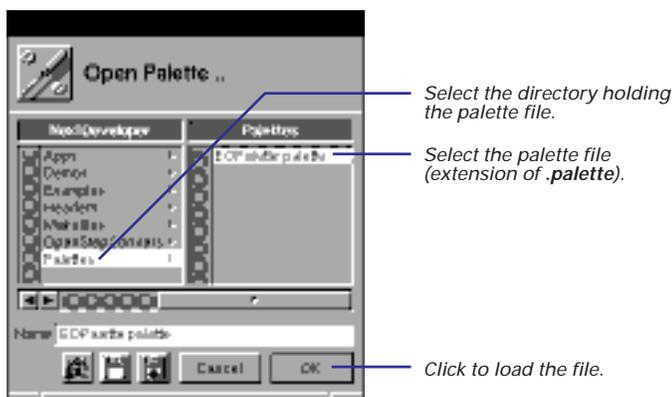
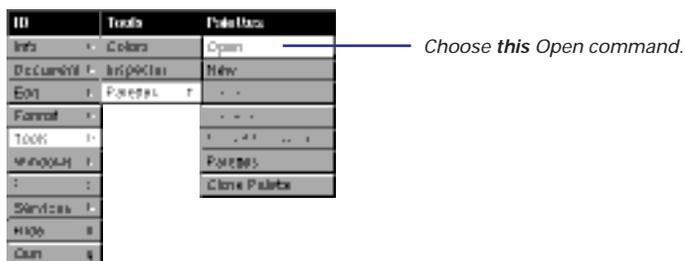
Palettes can be installed or uninstalled; installed palettes appear in the Palette window. Interface Builder stores as a user preference the names of the palettes you have previously installed and whether you currently want them installed. You can find out which palettes are uninstalled in the Palettes display of the Preferences panel. To see this display, choose Preferences from the Info menu.



To uninstall a palette, double-click its icon in the Preferences panel or use the Palette menu's Close Palette command.

Loading and Unloading Palettes

If the palette doesn't appear in the Preferences panel, open it using the Open command of the Palettes menu (not the Open command of the Document menu).



You can also open palettes by dragging them from the Workspace Manager's File Viewer onto the Palettes Preferences panel.

You unload palettes—thereby removing from your user preferences—by Alternate-dragging them off the Palettes display of the Preferences panel.

