2 Composing the Interface

The last thing one knows in constructing a work is what to put first.
Blaise Pascal, *Pensées*

Measurement began our might.
W. B. Yeats, “Under Ben Bulben”

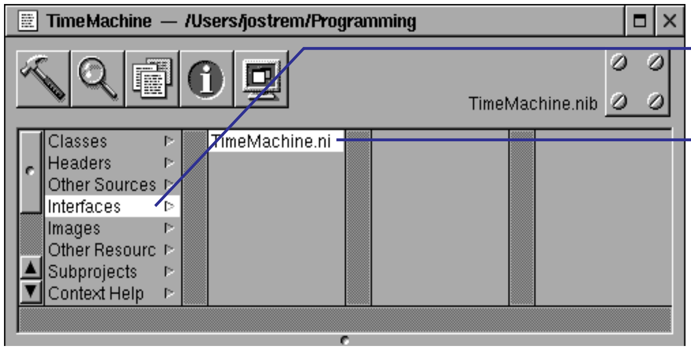
Opening a nib file

- **Double-click a nib file in Project Builder.**

  Or
- **Double-click a nib file in the Workspace's File Viewer.**

  Or
- **In Interface Builder, choose Document ▶ Open and select a nib file in the Open panel.**

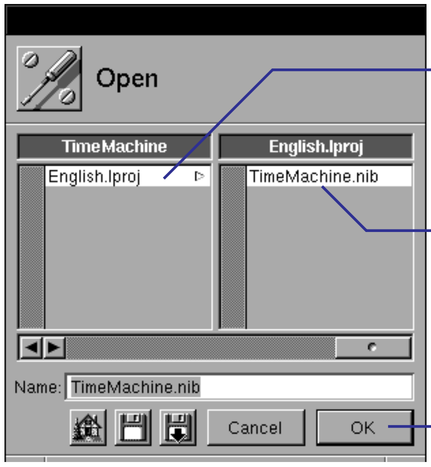
You'll usually open a nib file from Project Builder, since that is the central tool for application development. When you create an application project, Project Builder creates a nib file that has the same name as the project and, like all nib files, ends with the extension **.nib**. Opening a nib file switches control to the Interface Builder application, which you use to create the interface.



Nib files are under Interfaces.

Double-click the nib file or its icon.

You can also open nib files using the standard methods of opening files, such as using the Open panel.



*Select the appropriate **.lproj** directory.*

***English.lproj** contains nib files and other resources for applications that are localized for readers of English.*

Select a nib file.

Click here to open the nib file.

Nib files (files that have a **.nib** extension) are file packages that archive the class definitions, objects, and the connections between objects when you create an interface in Interface Builder. See "What's in a Nib File" in this chapter for some conceptual background.

When Interface Builder Starts Up

When you open a nib file, Interface Builder displays several windows and panels on your screen.

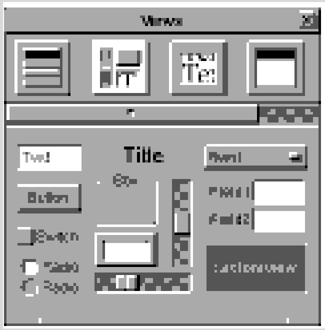
The palette window Holds all currently loaded palettes of objects. You select a palette by clicking its icon. Then you drag objects from the palette to the appropriate surface.

The interface window This window or panel displays the actual interface that you're working on. If this is the first time you've opened a main nib file in Project Builder, an empty window is displayed.

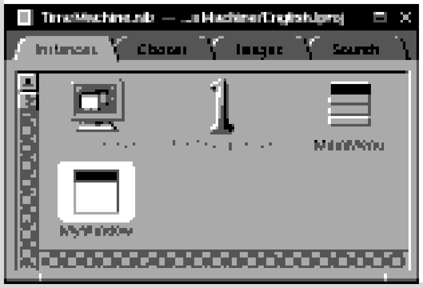
The nib file window This window contains multiple views that display the contents of the nib file. These views show archived objects, the connections among objects, the current class hierarchy (including any custom classes that you may have created), and the images and sounds stored in the nib file. Click a tab to switch the view.

The Inspector panel This multiform panel displays the attributes, connections, and size of a selected object. It also presents the object's resizing characteristics, its associated help, and which class it inherits from.

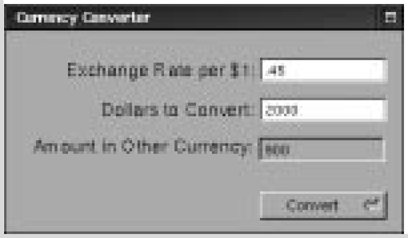
You can control whether the palette window and the Inspector panel appear when Interface Builder starts up by checking the appropriate boxes in the Preferences panel.



Palette window



Nib file window

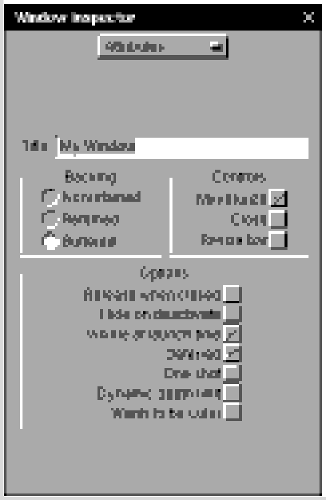


Exchange Rate per $1: .45

Dollars to Convert: 2000

Amount in Other Currency: 900

Interface window



Title: My Window

Backing: Nonretained, Retained, [✓ Buffered]

Controls: Miniaturize ✓, Close, Resize bar

Options: Release when closed, Hide on deactivate, Visible at launch time ✓, Deferred ✓, One shot, Dynamic depth limit, Wants to be color

Inspector panel

Creating a nib file

- ► **Choose one of the commands in the New Modules submenu of the Document menu.**

Or

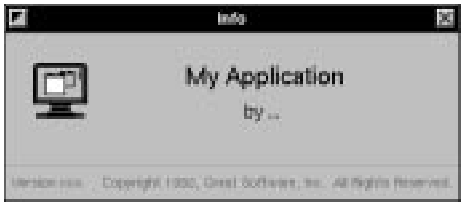
- ► **Choose New Empty from the New Modules submenu, then drag a window or panel from the Windows palette.**

Or

- ► **Choose New Application from the Document menu.**

Sometimes you need to create nib files directly in Interface Builder, typically when you want to add additional windows and panels to your application.

Most commands of the New Module submenu create nib files that contain a special kind of ready-made panel; your application can later load these nib files when it needs them. For example, if you choose New Info Panel, you'll get the following template panel:



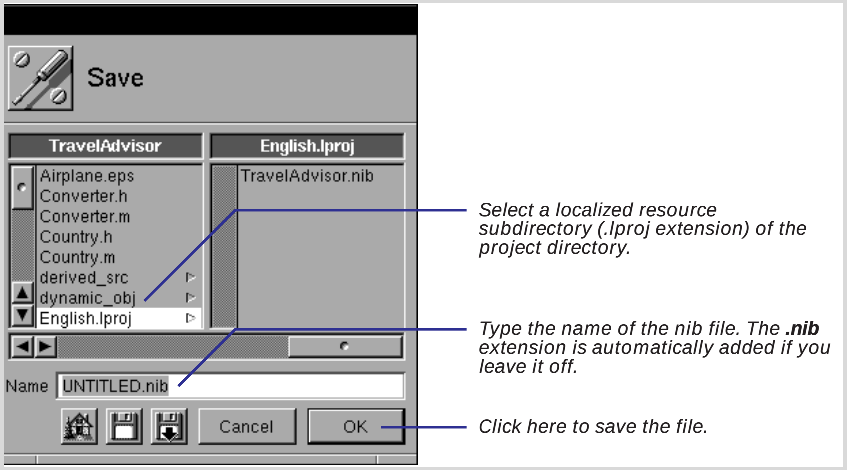
You can have auxiliary nib files, such as an Info panel, that you load into your program only when you need to. The programming technique of loading nib files on demand (lazy instantiation) is described in Chapter 11, "Dynamic Loading."

One reason to use New Application is to create a version of your interface for Microsoft Windows. For more information see Chapter 18.

The New Empty command just creates an empty nib file; you must create the windows and panels for it by dragging these objects from the Windows palette. The Document ► New Application command can create your application's main nib file (a nib file with the owner of NSApplication) if that hasn't already been done for you in Project Builder.

Saving the Nib File

An UNTITLED nib file window appears for each newly created nib file. After you make changes to an interface, remember to save the nib file. Choose Save from the Document menu and specify a path and file name in the Save panel. Interface Builder may ask if you want to insert the file into your project; confirm by clicking Yes.


Save
TravelAdvisor
English.lproj
Airplane.eps
Converter.h
Converter.m
Country.h
Country.m
derived_src
dynamic_obj
English.lproj
TravelAdvisor.nib
Name UNTITLED.nib
Cancel
OK
Select a localized resource subdirectory (.lproj extension) of the project directory.
Type the name of the nib file. The **.nib** extension is automatically added if you leave it off.
Click here to save the file.

What's in a Nib File

Every application has at least one nib file (actually a file package). The main nib file contains the main menu and often a window and other objects. An application can have other nib files as well. Each nib file contains:

Archived Objects Encoded information on OPENSTEP objects, including their size, location, and position in the object hierarchy (for view objects, determined by superview/subview relationship). At the top of the hierarchy of archived objects is the File's Owner object, a proxy object that points to the actual object that owns the nib file. (For a description of File's Owner, see the concept summary on File's Owner, First Responder, and Font Manager in Chapter 4.)

Sounds and Images Any sound or image files (TIFF or EPS) that you drag and drop over the nib file window.

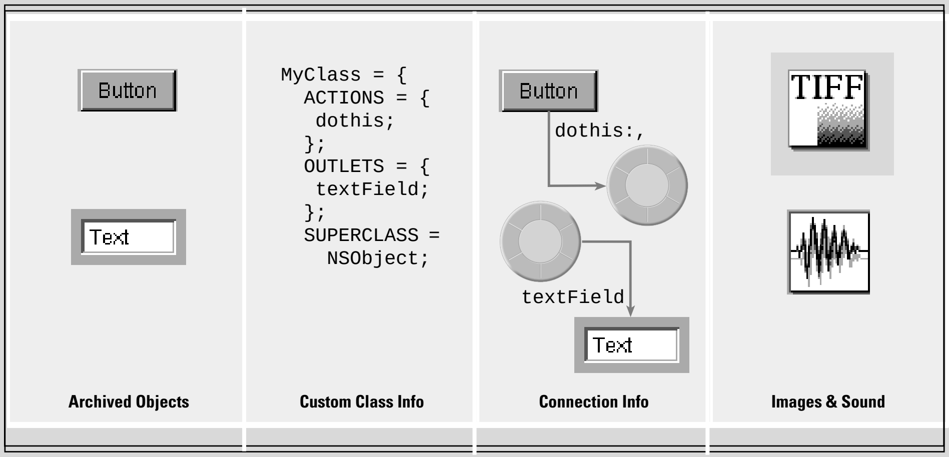
Class References Interface Builder can store the details of OPENSTEP objects and objects that you palettize (static palettes), but it does not know how to archive instances of your custom classes since it doesn't have access to the code. For these classes, Interface Builder stores a proxy object to which it attaches class information.

Connection Information Information about how objects within the object hierarchy are interconnected. Connector objects special to Interface Builder store this information. When you save the document, connector objects are archived in the nib file along with the objects they interconnect.

When You Load a Nib File

In your code, you can load a nib file by sending the NSBundle class the message **loadNibNamed:owner:** or **loadNibFile:externalNameTable:withZone:**. When you do this, the following things happen:

- The run-time system unarchives the objects from the object hierarchy, allocating memory for each object and sending it an **initWithCoder:** message.
- It unarchives each proxy object and queries it to determine the identity of the class that the proxy represents. Then it creates an instance of this custom class (**alloc** and **init**) and frees the proxy.
- The system unarchives the connector objects and allows them to establish connections, including connections to File's Owner.
- As the final step, the run-time system sends **awakeFromNib** to all objects that were derived from information in the nib file, signalling that the loading process is complete.



```
MyClass = {
  ACTIONS = {
   dothis;
  };
  OUTLETS = {
   textField;
  };
  SUPERCLASS =
    NSObject;
```

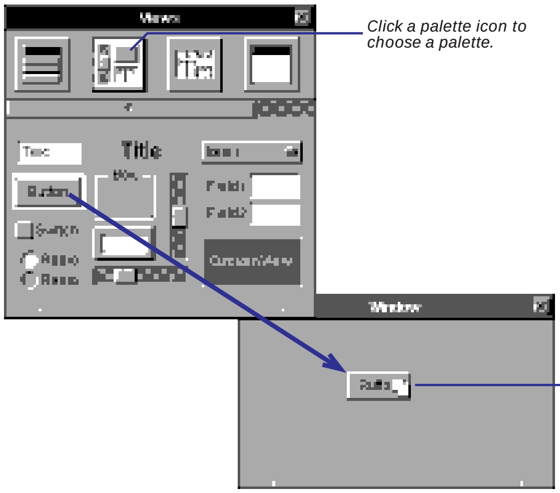
Using palettes

1 **Choose the palette you want.**

2 **Drag an object from the palette to the appropriate "surface."**

3 **Release the mouse button.**

The palette window of Interface Builder displays all currently loaded palettes. Each palette is represented in the window by an icon.


Views
Title
Button
Window
Click a palette icon to choose a palette.
Drag an object from the palette to the appropriate part of the interface. Release the mouse button when the object is in position.

"Where Palette Objects Go" in this chapter illustrates the proper "surfaces" for interface objects.

See the *Enterprise Objects Framework Developer's Guide* for more information on Enterprise Objects Framework palettes.

See "Managing palettes" in Chapter 5 for instructions on loading and installing palettes.

The palettes for the Application Kit are loaded by default. These palettes provide windows, panels, browsers, scroll views, buttons, text fields, and a number of other interface objects. You can also load palettes for other frameworks, such as Enterprise Objects , and you can load your own custom palettes.

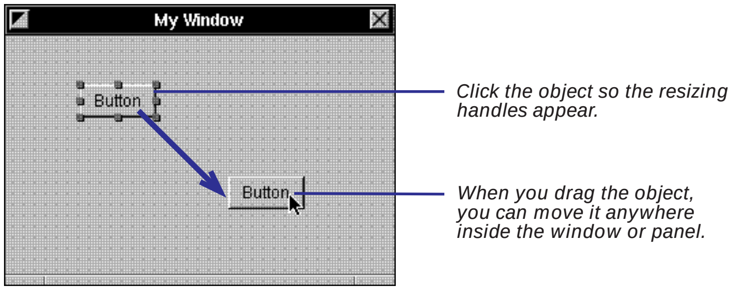
Note: Where you "drop" a window or panel is important, because it sets its initial position on the screen—the location where the window appears when the application starts up or when this particular nib file is loaded.

Placing interface objects

1 **Select the object you want to move.**

2 **Drag the object to the new location in the window or panel.**

To move an object around the "surface" of a window or panel, select the object and drag it with the mouse. The currently selected object has resize handles—small, gray rectangles— around its perimeter.


My Window
Button
Button
Click the object so the resizing handles appear.
When you drag the object, you can move it anywhere inside the window or panel.

You can adjust the size and location of objects precisely by specifying their origins, width, and height in the Size display of the object's Inspector.

See "Positioning and sizing precisely" in this chapter for details.

When you move an object, make sure that the mouse pointer is inside the object and not on a resize handle.

For greater precision, select an object and press the arrow keys; this moves the object an incremental distance in the required direction. If the alignment grid is off, this distance is one pixel; if it is on, the distance is eqaul to the size of the grid.

Selecting Multiple Objects

You can select multiple objects and then move, copy, or do other things with them as a group. There are two ways to select more than one object:

- Hold down the Shift key while you click objects in succession.
- Click in an empty area, then draw a "rubberbanding" rectangle around all objects you want selected.

After making the selection, press (**don't** momentarily click) the mouse pointer on one of the objects and drag the group to the new location. (Or do another suitable operation, such as copy and paste.)

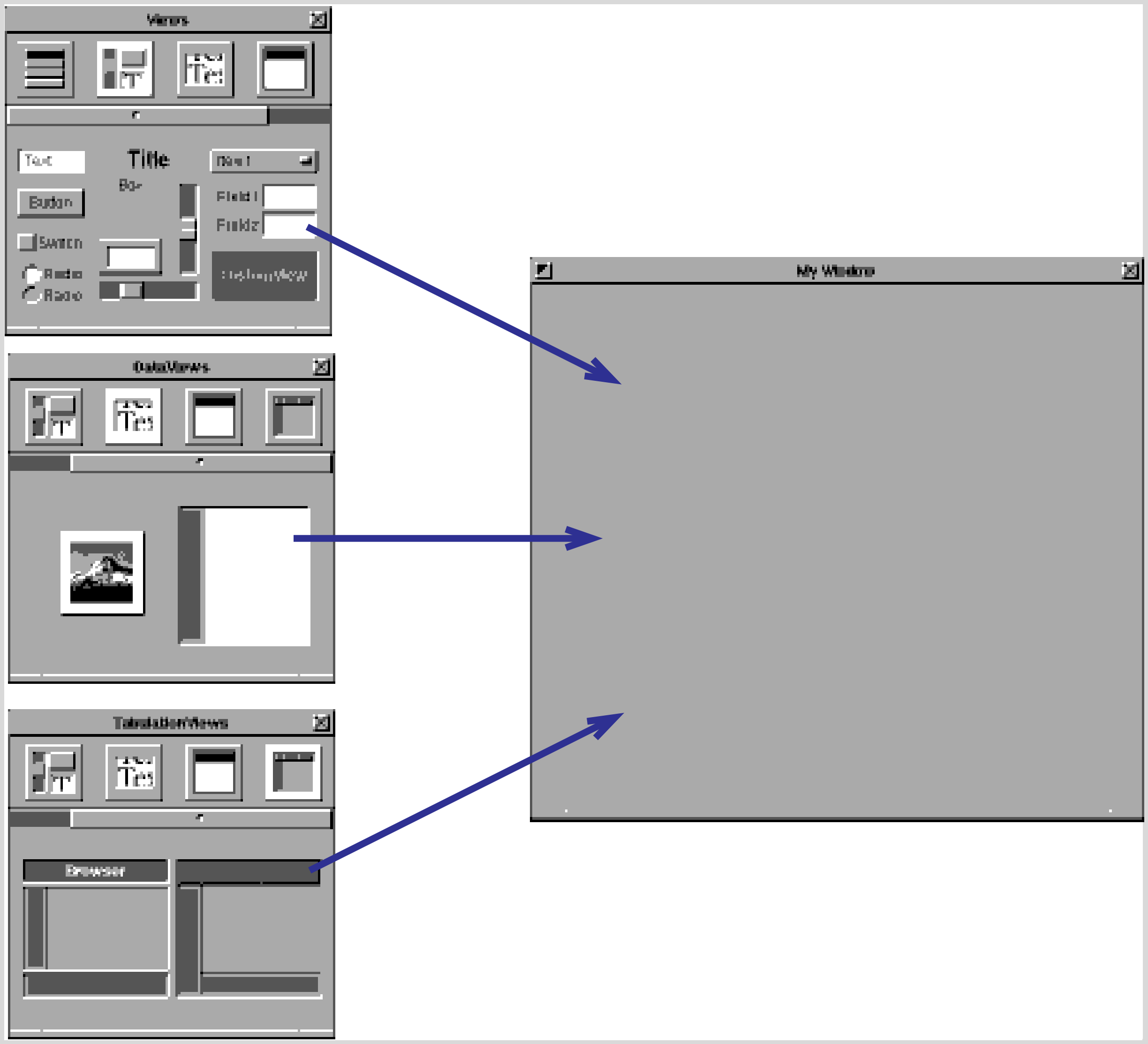
To deselect an object in a grouped selection, hold down the Shift key and click that object.

You cannot do sizing operations on multiple selected objects.

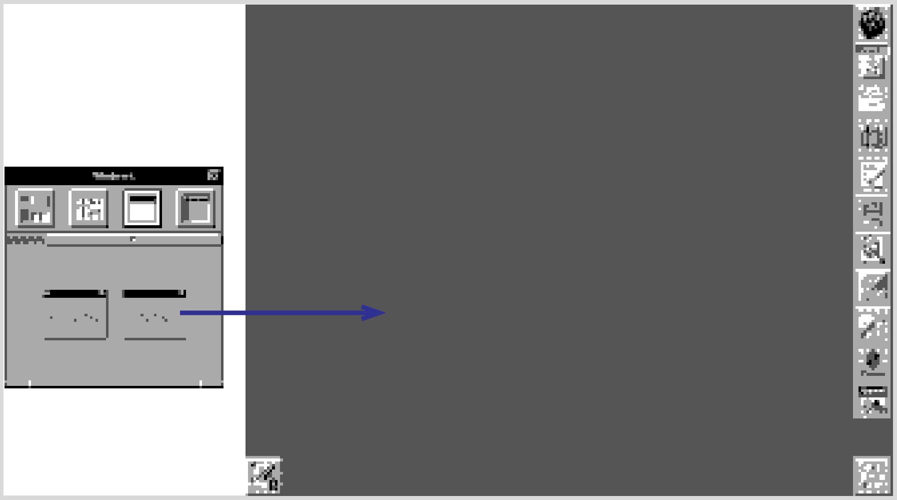
To select all objects in a window or panel, first select the window or panel, and then choose Select All from the Edit menu. You can also use this command to select all items in the Instances or Classes display of the nib file window. The key equivalent for Select All is Command-a.

Where Palette Objects Go

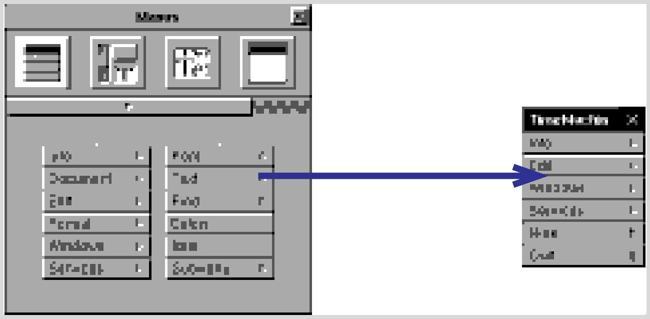
You put items from the Views, TabulationViews, and DataViews palettes anywhere within the bounds of a window or panel. These items include buttons, labels, fields, boxes, text fields, scroll views, browsers, and custom views.



You can put windows and panels anywhere in the work space. Nothing contains them except the screen.

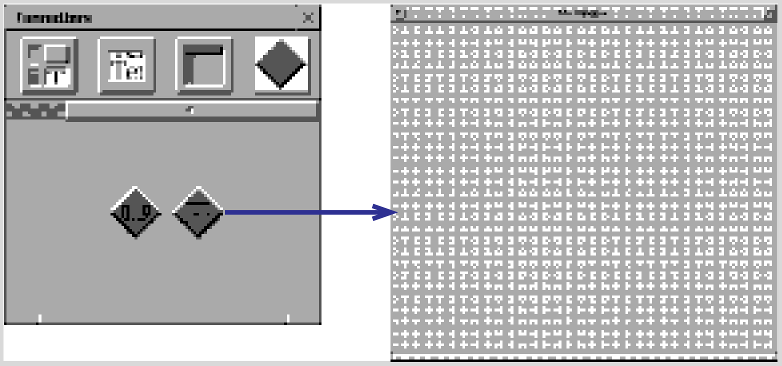


You drag a menu item from the Menus palette and drop it in the application's menu. When you release the mouse button, Interface Builder inserts the item between the two menu commands underneath it.

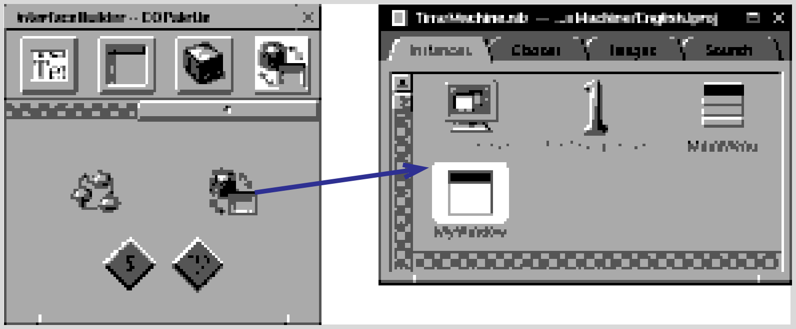


Where Palette Objects Go (continued)

You drag a formatter from the Formatters palette and drop it on a text field or a cell in a tableview. That formatter will control the formatting for all of the cells in that column.



Some palettes, like the one for the Enterprise Objects Framework, carry objects that do not appear on an interface. These usually are controller objects that perform management or computational functions. Drop these objects anywhere on the Instances display of the nib file window.



Initializing text

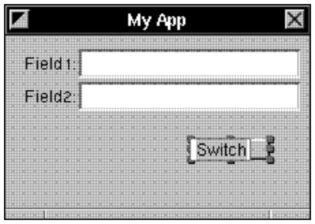
1 **Select the object.**

2 **Double-click the text inside the interface object.**

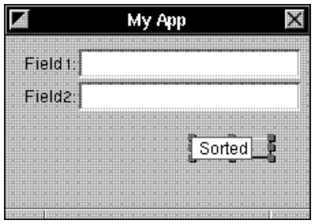
3 **Edit the text.**

4 **Deselect the text by clicking outside of the object.**

Many of the palette objects include text as a component. Buttons of all sorts usually have titles, boxes usually name the elements they group, and so on. Interface Builder initially sets the text in most of these objects to the name of the object itself (such as “Button” or “Text”). After you drag the palette object onto a window or panel, you will probably want to delete these text strings or rename them to something meaningful. This text is what is initially displayed when your application loads the nib file; the text can change later if one of the objects in your application requests it.



Double-click the line to select it.

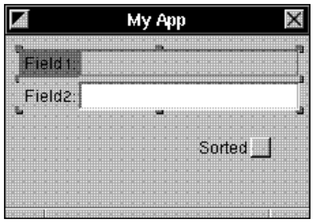


Type the new text. When finished, click outside the object to set the text.

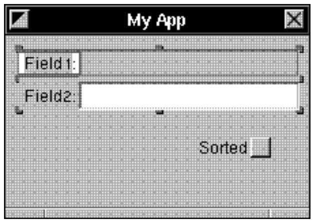
When text is selected, you can move the cursor among the characters by pressing the left and right arrow keys. You can delete characters by pressing the Delete key.

Matrices—compound objects, such as radio buttons and form fields—need a slightly different procedure for selecting text for initialization: You must double-click the embedded text item twice, the first time to select the embedded object, and the second time to select the text inside the object.

“Creating matrices of objects” in this chapter describes how to create these compound objects. Also see “Compound Objects” in Chapter 3 for a conceptual summary of matrices and other compound objects.



In matrices, double-click the text to select the embedded object.



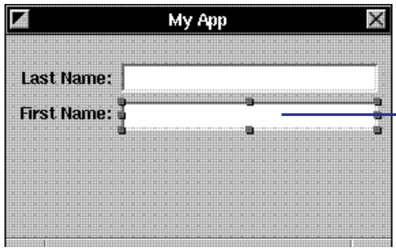
Double-click again to select the text.

Removing objects

1 **Select one or more objects.**

2 **Choose Cut from the Edit menu.**

To delete objects from an interface, select the objects and choose the Cut command. You can also use the Delete command, but the differences are significant. Cut saves the selected objects to the pasteboard, so you can retrieve the objects with the Paste command (Command-v). The Delete command permanently deletes the selected items.



Select one or more objects.

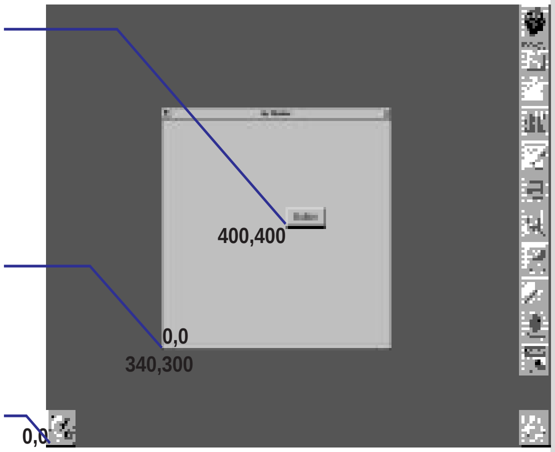
Choose Cut from the Edit menu or press Command-x.

The Coordinate System in Interface Builder

The Size display of an object's Inspector panel shows that object's precise location and dimensions. The **x** and **y** fields hold the origin point (horizontal and vertical) for the object within the drawing context of its enclosing window, panel, or superview. The **w** and **h** fields hold the width and height. All values are in pixels.

Within a window or panel, the lower left corner is origin 0,0. This is the point of reference for objects within that window or panel. Therefore, when you move or size objects downward or to the left, the values in the Size display are decreased.

The point of reference for a window or panel (or origin 0,0) is the lower-left corner of the screen. This means that the same relationship applies: if you decrease its **x** value in the Size display, it moves to the left; if you decrease its **y** value, it moves toward the bottom of the screen.



The origins and dimensions of view objects are based on the origin point of the window or panel that contains them. (The origin does not include the resize bar.)

In this example the window's placement is 340,300 relative to the screen origin. This same point on the screen is the origin for views in the window.

The origins and dimensions of windows and panels are based on the screen's origin point.

Duplicating objects

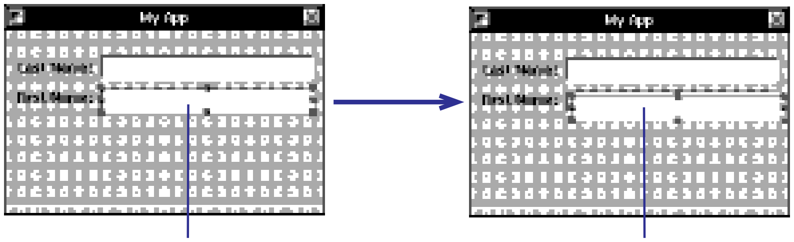
1 **Select an object.**

2 **Choose Copy from the Edit menu.**

3 **Choose Paste from the Edit menu.**

4 **Position the new object.**

You can duplicate an object just as you would with geometric shapes in a drawing application. The copied object has the dimensions and most other attributes of the original object.



Select an object.

Choose Copy, then Paste from the Edit menu.

The copy of the object is slightly offset over the original object.

In addition to the objects that appear on the interface, you can copy your custom non-UI objects—represented as cubes in the icon mode of the nib file window Instances display—as well as your windows and panels. Just click to select them and then copy and paste.

Tip: Instead of choosing Copy and Paste from the Edit menu, you can press Command-c (Copy) and Command-v (Paste).

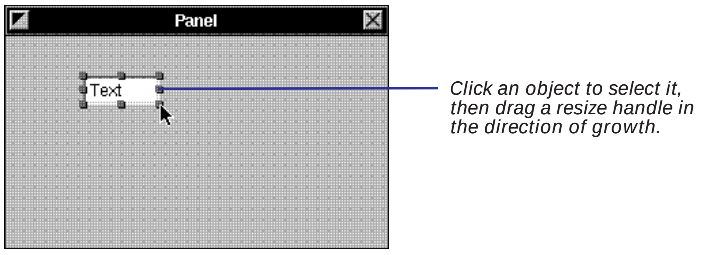
You can also duplicate groups of selected objects by copying them and then pasting them. See “Selecting Multiple Objects” in this chapter for details on making multiple selections of objects.

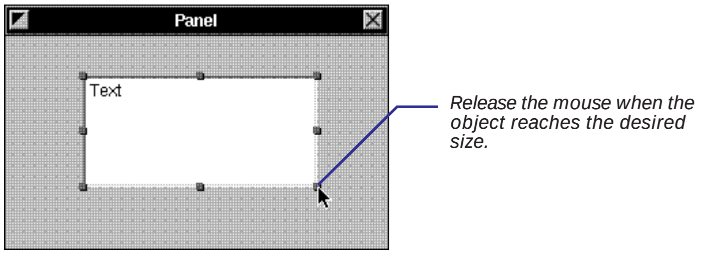
Sizing interface objects

1 **Select an object.**

2 **Drag a resize handle in the desired direction.**

Interface objects in Interface Builder resize to any practical dimension. You can, for instance, increase the size of a button so it fills a window. Most interface objects, however, do not resize below a certain minimum size of usefulness.


Panel
Text
Click an object to select it, then drag a resize handle in the direction of growth.


Panel
Text
Release the mouse when the object reaches the desired size.

To affect just one dimension of the object, drag a top, bottom or side handle. To adjust both dimensions simultaneously, drag one of the corner handles. To size both dimensions proportionally, hold down the Shift key while you drag a corner resize handle.

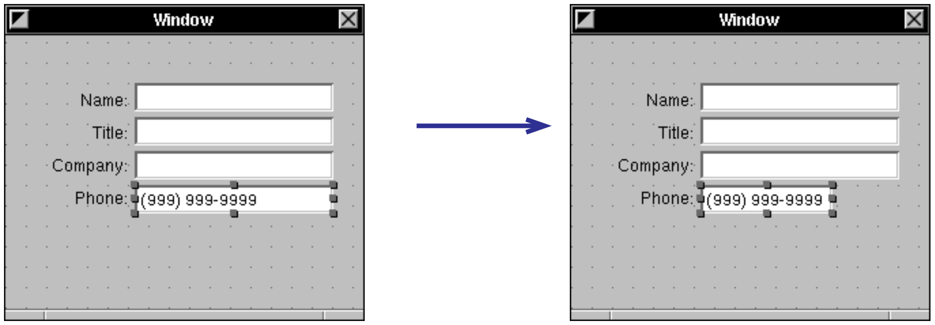
You can adjust the size and location of objects precisely by specifying their origins, width, and height in the Size display of the object's Inspector panel. See "Positioning and sizing precisely" for details.

Shrinking objects to their minimum size

1 **Select one or more objects.**

2 **Choose Format ▶ Size ▶ Size to Fit.**

To conserve screen real estate, or to enhance the appearance of your interface, you might want to have view objects just large enough for any text they contain. You can do this with the Size to Fit command.



Select an object that you want to shrink. *Choose Format* P *Size* P *Size to Fit* *The copy of the object is slightly offset over the original object.*

The Size to Fit command has no affect on matrices, custom views, and some other objects. If you delete the text from a button, text field, or other object that holds text, and then apply the Size to Fit command to it, that object shrinks to its minimum (and probably unusable) size.

Making interface objects the same size

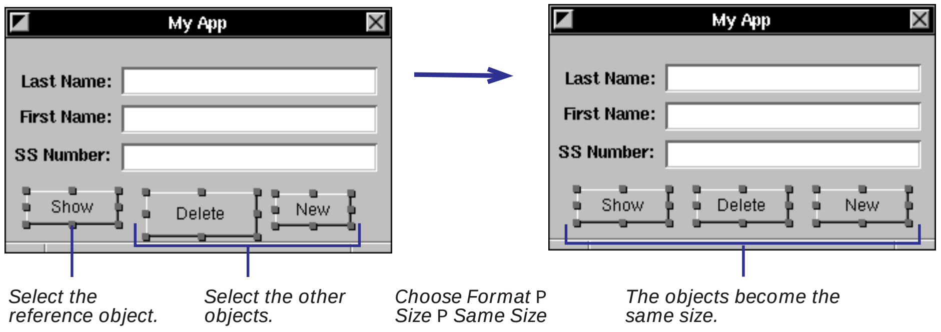
1 **Select the reference object.**

2 **Add to the same selection the objects that you want resized.**

3 **Choose Format ▶ Size ▶ Same Size.**

To lend a look of consistency to your interface, you often want to make similar objects have the same size. Buttons across the bottom of an attention panel, for instance, should be the same exact size. Interface Builder gives you an easy way to do this, allowing you resize selected objects to a *reference* object. You designate the reference object differently, depending on your method of selection:

- If you press the Shift key while clicking objects in succession, the first object clicked is the reference object.
- If you draw a selection rectangle around a group of objects, selecting the objects simultaneously, the topmost object in the selection (often the most recently added object) is the reference object.


My App
Last Name:
First Name:
SS Number:
Show
Delete
New
My App
Last Name:
First Name:
SS Number:
Show
Delete
New
Select the reference object.
Select the other objects.
Choose Format P Size P Same Size
The objects become the same size.

Tip: In most situations, you should select multiple objects by Shift-clicking them because this method gives you more control (you know which object will be the reference object).

Positioning and sizing precisely

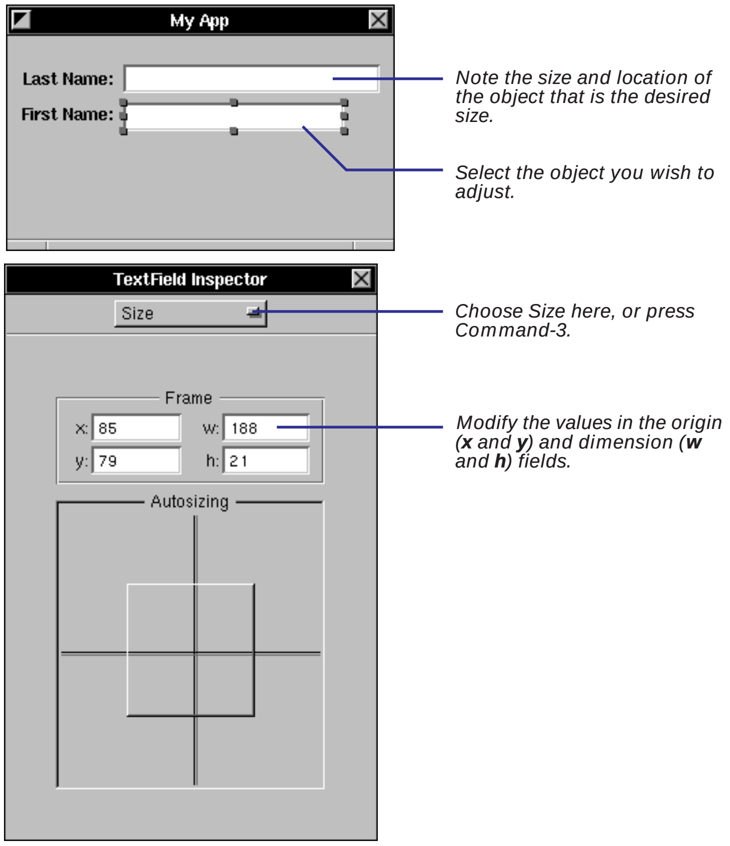
1 **Select an object.**

2 **Choose Tools ► Inspector to bring up the Inspector panel.**

3 **Choose Size from the Inspector pop-up list.**

4 **Modify the object's origin point or its dimensions.**

You can move and resize objects in your interface with numerical exactness using the Inspectors for those objects. You'll occasionally find need for such exactness, such as when you want to size an image view to the same dimensions as the image that it will display. More frequently, you'll use this method to align objects or make sure they're the same size.


My App
Last Name:
First Name:
Note the size and location of the object that is the desired size.
Select the object you wish to adjust.
TextField Inspector
Size
Choose Size here, or press Command-3.
Frame
Modify the values in the origin (**x** and **y**) and dimension (**w** and **h**) fields.
Autosizing

x: 85 w: 188
y: 79 h: 21

See "Automatically resizing objects" in Chapter 3 for information on the Autosizing area.

When you press Return in an origin or dimension field, the object moves to the new position or expands or contracts to the new size.

Tip: You can also move selected objects incrementally—and precisely—by pressing the arrow key that points in the required direction. Each incremental "nudge" moves the object the distance of the grid or, if the grid is turned off, one pixel.

Sizing windows and panels

▶ **Drag the resize bar in the direction you want the window to grow.**

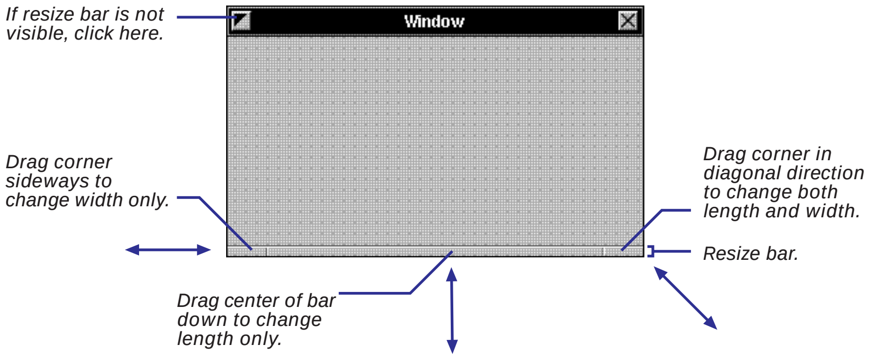
Or

1 **Select the window by clicking its titlebar.**

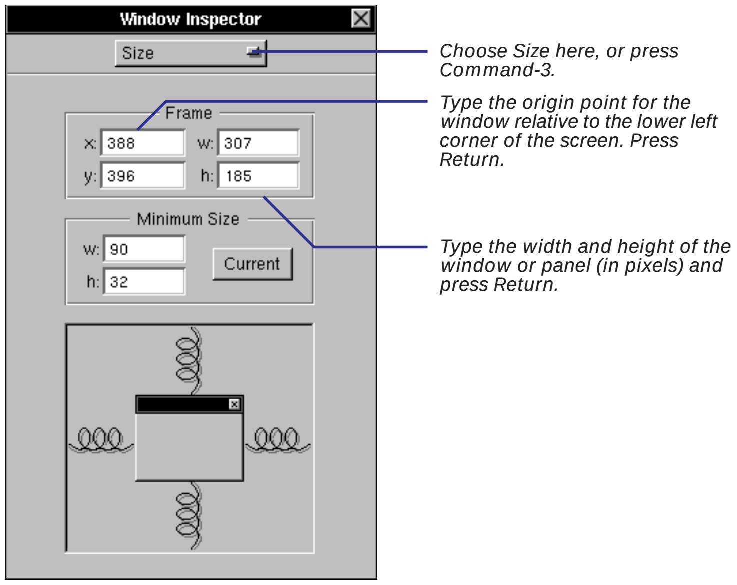
2 **Choose Tools ▶ Inspector to bring up the inspector window.**

3 **Enter the dimensions in the Size display of the Inspector panel.**

After you drag a window or panel from the Windows palette and drop it on the screen, you'll probably want to resize the object to a suitable dimension. To resize a window, drag the resize bar in the required direction.


If resize bar is not visible, click here.
Window
Drag corner sideways to change width only.
Drag corner in diagonal direction to change both length and width.
Resize bar.
Drag center of bar down to change length only.

To resize windows or panels with greater precision, enter the exact dimensions in the Size display of the Inspector panel.


Window Inspector
Size
Frame
x: 388
w: 307
y: 396
h: 185
Minimum Size
w: 90
h: 32
Current
Choose Size here, or press Command-3.
Type the origin point for the window relative to the lower left corner of the screen. Press Return.
Type the width and height of the window or panel (in pixels) and press Return.

You can also use the Inspector panel to size interface objects with numerical exactness. See "Positioning and sizing precisely" for more information.

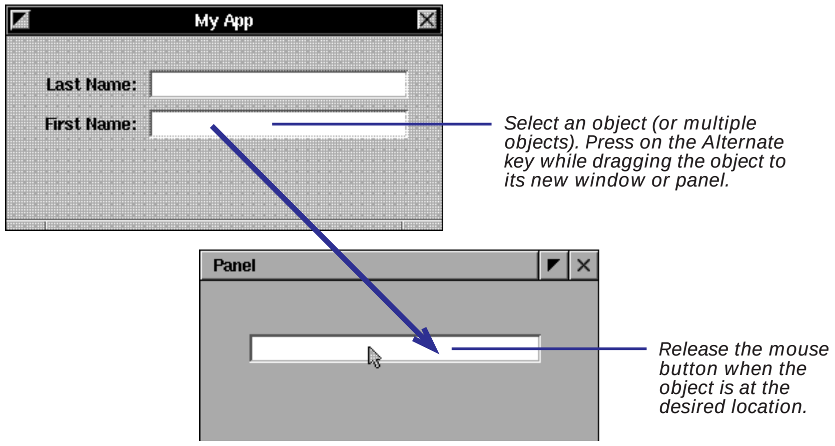
Also see "The Coordinate System in Interface Builder" for some conceptual background.

Moving objects to other windows

1 **Select one or more objects.**

2 **Alternate-drag the objects to the other window or panel.**

To move objects from one window or panel to another within the same nib file, hold down the Alternate key and drag them between windows. This action moves the objects if the windows are in the same nib file and copies them if they windows are in the different nib files.


My App
Last Name:
First Name:
Panel
Select an object (or multiple objects). Press on the Alternate key while dragging the object to its new window or panel.
Release the mouse button when the object is at the desired location.

If the windows are in the same nib file and you want to copy rather than move the selected objects, you have two alternatives:

- Copy the objects using the Copy command; click the other window or panel to activate it, and use the Paste command to copy the objects from the pasteboard.
- Copy and paste the objects in one window, then Alternate-drag the duplicated objects to their new window or panel.

Copying Objects to Other Interfaces

To copy objects to different nib files, simply select a group of objects in one nib file and Alternate-drag those objects to the appropriate "surface" of the other nib file.

You can copy entire windows or panels as well as custom, non-UI objects between interfaces.

The surface you drop objects onto must be compatible:

- Non-UI objects must be dropped over the nib file window.
- View objects are dropped over a window or panel or over the nib file window.
- Windows and panels must be dropped over the nib file window.

The basic technique of Alternate-drag also copies the connections among selected objects. See "Copying interconnected objects" in Chapter 4 for details.

Arranging objects

1 **Choose Format ▶ Align ▶ Alignment to bring up the Alignment panel.**

2 **Set the characteristics of the grid in the Alignment panel.**

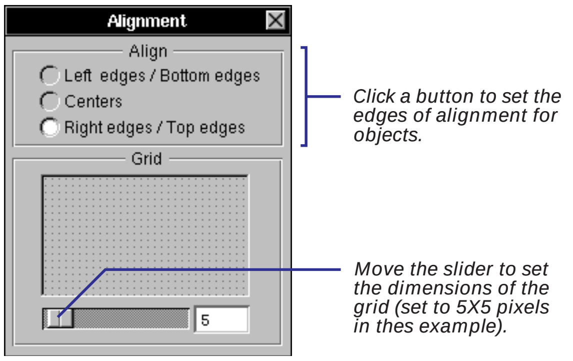
3 **Choose Format ▶ Align ▶ Turn Grid On to turn on the grid.**

4 **Align objects with the grid.**

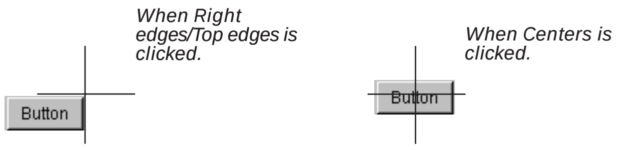
When you compose your interface, you usually want to arrange the objects in that interface in some appealingly regular way. You want buttons, for instance, to be aligned on the same invisible horizontal or vertical line. Or you want the distance between text fields in a form application to be exactly the same. Interface Builder gives you a set of tools for arranging objects.

Every window or panel has a grid associated with it. You may turn this grid off and on. When it is on and you move an object, an edge of that object "snaps," like a nail to a magnet, to the adjacent intersecting lines of the grid.

You set the dimensions of this grid and the edges of alignment in Interface Builder's Alignment panel.

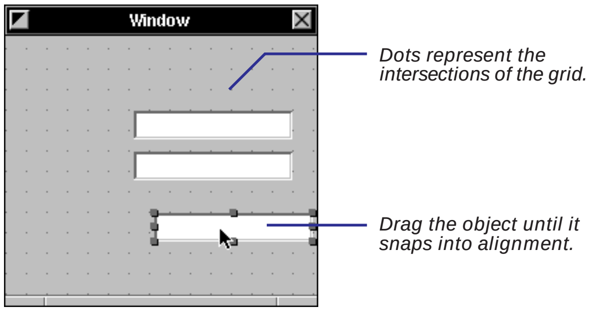

Alignment
Align
Left edges / Bottom edges
Centers
Right edges / Top edges
Grid
5
Click a button to set the edges of alignment for objects.
Move the slider to set the dimensions of the grid (set to 5X5 pixels in thes example).

The buttons in the Align section of the Alignment panel determine what point or edge of interface objects snaps to the grid.


When Right edges/Top edges is clicked.
Button
When Centers is clicked.
Button

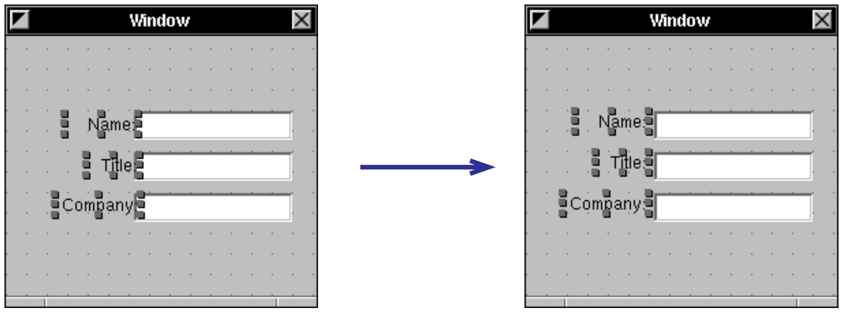
Once you have your grid set up, make sure the grid is turned on: Choose Format ▶ Align ▶ Turn Grid On. If you also want the grid visible, choose Show Grid from the same menu.

Now align the objects, either individually or as a group, using the grid.


Window
Dots represent the intersections of the grid.
Drag the object until it snaps into alignment.

There are other ways to align objects that don't require using the mouse. With the grid turned off, you can drag view objects from a palette and visually align them as precisely as possible. Then set the grid spacing, turn the grid on, and choose the Align to Grid command.

Once the grid is set and on, align the objects, either individually or as a group.



With grid off, place objects for alignment and then select them.

Choose Format P *Align* P*Align to Grid*

Objects become aligned, in this case moving on the grid to their left.

With the Align to Grid command, the direction of alignment is toward the origin point of the window or panel (in other words, toward the lower-left corner). You should be aware of this when placing objects for later alignment.

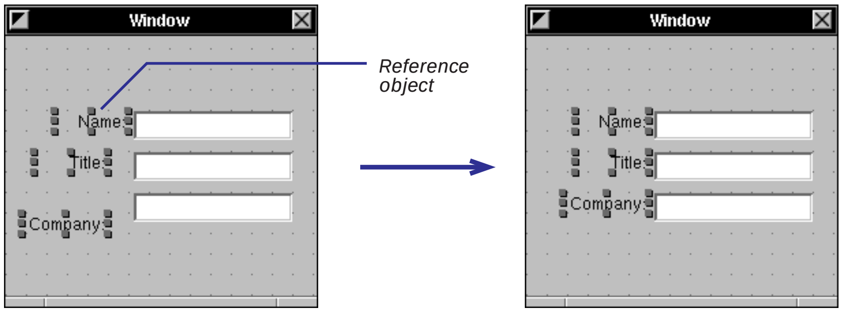
Tip: You can align selected objects to a grid, singly or as a group, by pressing the arrow keys in the direction of alignment. When the grid is turned on, the unit of increment changes from one pixel to whatever the grid spacing is.

Making Columns and Rows of Objects

It is more efficient to align groups of objects than to align single objects successively. With the Make Column and Make Row commands, Interface Builder aligns groups of selected objects to a *reference* object. You designate the reference object by the way you select multiple objects.:

- If you press the Shift key while clicking objects in succession, the first object clicked is the reference object.
- If you draw a selection rectangle around a group of objects, and so select objects simultaneously, the topmost object in the selection (often the most recently added object) is the reference object.

For most purposes, Shift-clicking objects is the preferred method because it permits more control.



Click the reference object first, then Shift-click the remaining objects to select them.

Choose Format P Align PMake Column

The objects become vertically aligned to the reference object.

NeXT's Basic UI Design Philosophy

Composing a user interface involves much more than techniques for placing, sizing, and arranging objects on a window. When you put your application's UI together in Interface Builder, keep in mind the following principles that NeXT has developed, through trial and test, to guide interface designers.

Make It Consistent

When all applications share the same basic interface, each application benefits. Consistency makes each application easier to learn, and so increases the likelihood of acceptance and use. Just as with so many natural "interfaces" in life, conventions count for a great deal. Although different applications are designed to accomplish different tasks, they all share, to some degree, a set of common operations such as selecting, editing, scrolling, and setting options. Reliable conventions are possible only when these operations are carried out the same way for all applications.

Make it Feel Natural

Try to make the screen a visual metaphor for the real world, so that the objects in it reflect the way the represented things actually behave. That's what an "intuitive" interface is – it behaves as we expect based on our experience with objects in the real world.

Modeled objects don't have to mimic every detail of their real counterparts, but they should behave in similar ways. For example, objects in the real world stay where we put them; they don't disappear and reappear again, unless someone causes them to do so. Users should immediately recognize the objects in your interface and should use them for the sorts of operations that people typically use their real counterparts for.

Put the User in Charge

Users should have the widest freedom of action. If an action makes sense, your application should allow it. In particular, avoid setting up arbitrary modes, periods during which only certain actions are permitted. On occasion, however, modes are a reasonable way of solving a problem, particularly in these forms:

- attention panels
- modal tools
- "spring-loaded" mode (while mouse or key down)

But these modes should be freely chosen, provide an easy way out, be visually apparent, and keep the user in control.

At the same time, you should try to anticipate what users will do and ease their way, reducing the actions they must perform. Give them freedom, but still act on their behalf without waiting for their instructions. These helping actions should be simple and convenient, like, in the Open panel, preselecting a directory that is probably in the path of the final selection.

Focus on the Mouse

The mouse is the most appropriate instrument for a graphical interface. The keyboard is principally used for entering text, but the mouse is the instrument by which users manipulate the objects of your interface. Your user interface should support three paradigms of mouse action:

- Direct manipulation
- Targeted action
- Modal tool

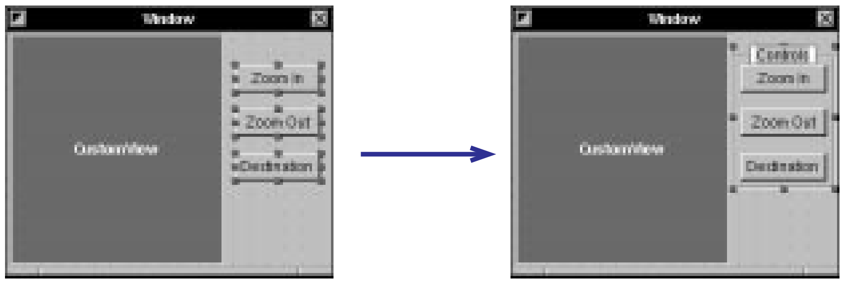
Grouping objects

1 **Select the objects you want grouped.**

2 **Choose Format ▶ Group ▶ Group in Box.**

When you group objects, Interface Builder draws a box around them. You select, move, copy, cut, and paste the objects within the box as a group. Interface Builder gives you two ways to group objects.

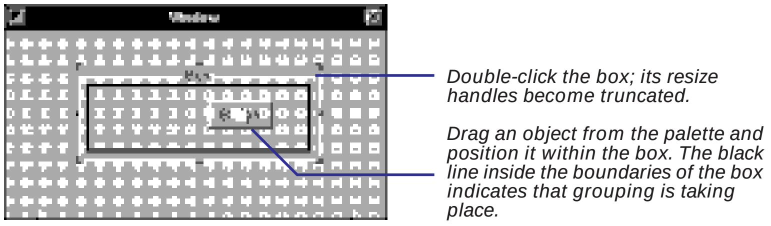
In the first method, you select the objects of the group and choose a menu command. The box has a title, initially "Title." To change this, double-click the title to select it (as in the example above, on the right). Then type the new name for the grouped objects.



The objects of your group should be adjacent. *Choose Format* P*Group* P*Group in Box* *A box encloses the objects.*

To ungroup the objects, making each object individually selectable again, select the group and choose Format ▶ Group ▶ Ungroup.

In the second method, you use the box object in the Views palette. First, drag a box onto a window or panel. Then add its contents:



Double-click the box; its resize handles become truncated.

Drag an object from the palette and position it within the box. The black line inside the boundaries of the box indicates that grouping is taking place.

See Chapter 3 "Setting an Object's Attributes" for other things you can do with box objects.

See the specifications of the NSScrollView and NSSplitView classes in the *Application Kit Reference*.

The Group submenu has two other interesting commands. With Group in Scroll View, you can bind an NSText object (or your own custom view object) to horizontal and vertical scrollbars. With Group in Split View, you can group two related views (often a custom view and a browser object) in a split view, which has a sizing bar between the views.

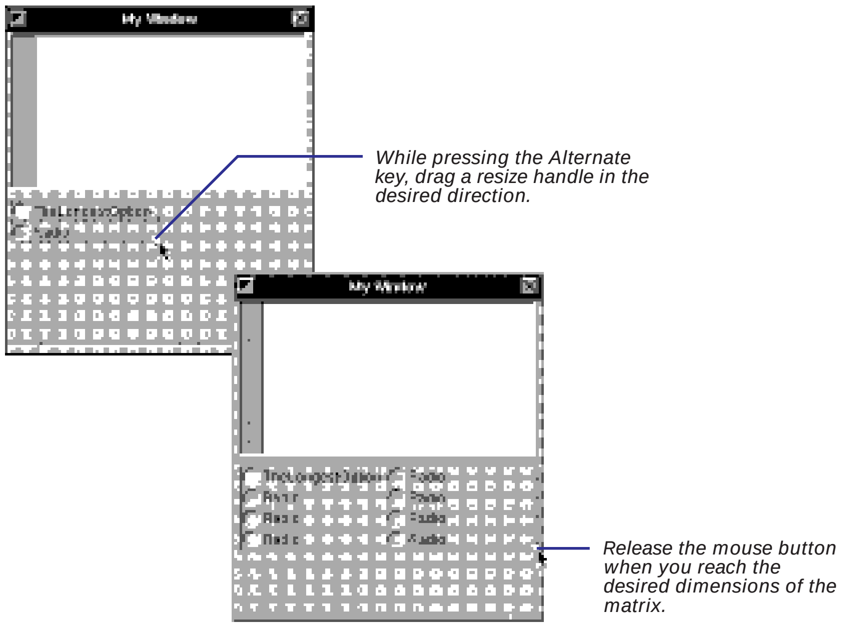
Creating matrices of objects

1 **Drag a suitable object from the Views or TabulationViews palette.**

2 **Alternate-drag a resize handle of the object.**

You can easily transform certain objects in the standard Interface Builder palettes into matrices of those objects. A matrix (defined by class NSMatrix) imposes a regular size and intervening distance on a set of identical objects. Matrices afford an easy way to compose forms, arrays of buttons and sliders, and multiple-column browsers. To create a matrix, drag one of these objects to a window or panel and size it to the maximum dimension you anticipate for a cell in the matrix:

- text field
- button
- switch button
- radio button
- form field
- slider (vertical or horizontal)


My Window
While pressing the Alternate key, drag a resize handle in the desired direction.
My Window
Release the mouse button when you reach the desired dimensions of the matrix.

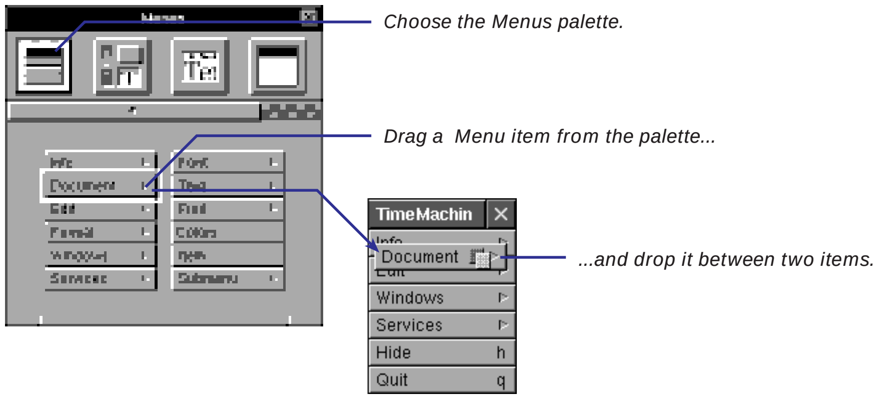
Tip: To make a browser with more than one column, drag a browser object from the TabulationViews palette onto your interface; then Alternate-drag the right resize handle until the desired number of columns appear.

Creating menus

1 **Drag a menu item from the Menus palette.**

2 **Drop it between two menu items in your application's menu.**

Menus are just as important as windows and panels for an interface. Menu commands initiate most of the standard functions of an application, such as printing, opening files, or cutting and pasting text. That's why Interface Builder's Menus palette holds a number of ready-made submenus and menu items.


Choose the Menus palette.
Drag a Menu item from the palette...
...and drop it between two items.
TimeMachin
Info
Document
Edit
Windows
Services
Hide h
Quit q

You can make menu items active or inactive by default. Select the item and set the Disabled button in the Attributes display of the cell's Inspector. See Chapter 3, "Setting an Object's Attributes" for more information on using the Inspector panel.

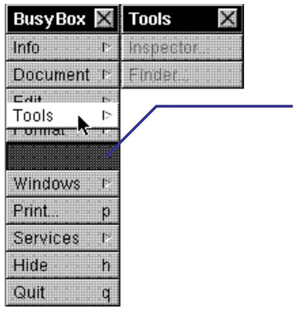
See Chapter 4, "Making and Managing Connections" to learn what you must do to connect menu items with the objects that are to handle menu commands.

Click several menu items in your application's main menu and note how some cells in the submenus are dimmed. Dimmed cells indicate that, as the default, the command is inactive until some condition occurs in your code that causes your application to activate the command.

You delete a menu item just as you do with any other object in Interface Builder: select it, then choose the Cut command from the Edit menu (Command-x) or press the Delete key. Also, as with other Interface Builder objects that display text, you can easily change the titles of menu items:

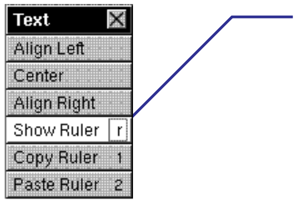
- Double-click the text to select it
- Type the new title or edit the old one
- Click outside the cell to set the new title or press Enter

You can also do two special tasks with menu items: re-sequencing and assigning Command keys. By re-sequencing, you change the order in which items are listed in a menu. By assigning a Command key to an item, you give the user of your application a command key equivalent—a shortcut way to invoke the command (as Command-x is a shortcut for invoking the Edit menu's Cut command).



Resequencing

Drag the menu item from its old location (shown by a black rectangle) and drop it between two menu items (its new location).



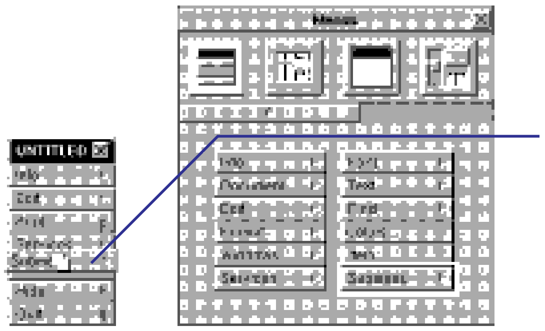
Command-key Assignment

Double-click the menu item near its right edge. In the square, type the letter for the Command key.

Make sure that the Command-key letter is not assigned to some other menu command in your application.

Custom menus

In addition to the standard menu commands and submenus, Interface Builder makes it easy for you to compose your own custom menus. Use the Submenu cell in the Menus palette to create custom submenus and use the Item cell for custom menu commands. The Print command is frequently added as a custom cell.



Drag the Submenu item from the Menus palette and drop it between two menu items in the application's main menu.

Change the title of Submenu and click the cell to expand it. Then add Items from the Menus palette to the new submenu and change their titles.

Layering objects

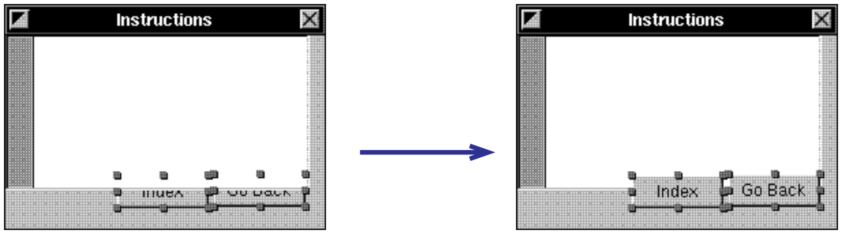
1 **Select an object.**

2 **Choose Format ► Bring to Front or Format ► Send to Back.**

Every object on a window or panel in Interface Builder is on its own layer. That's why when you move one object over another object, the first object moves in front of the second or moves behind it. The most recently added object is generally on the topmost layer.

Occasionally, you need to alter the layering order to make an object visible or to have it appear behind other objects. To do this, apply the Bring to Front command or the Send to Back command (on the Format menu) to selected objects.

These commands are only useful while you are editing the interface; the final version of your interface must not have overlapping views.



The buttons are behind the ScrollView. To fix this, select the buttons.

Choose Format PBring to Front

The buttons appear in front of the ScrollView.