

Reduce, Reuse, and Recycle: Working with Symbols

Imagine if you created a Flash movie in which all the elements were variations on the same theme. Until now, you would have to spend a great deal of time painstakingly creating each element individually. With the help of *symbols*, however, you will never have to toil in vain, oppressed under the weight of backbreaking digital labor. You will be able to create one master image and then base all subsequent images upon it, making specific changes to each as you go. Symbols help streamline the creative process and maximize your artistic juices. The purpose of this chapter, therefore, is to explore the glorious world of symbols.

The chapter covers the following topics:

- **Understanding symbols**
- **Creating native Flash symbols**
- **Using the Library to work with symbols**
- **Working with symbol instances**

Understanding Symbols

Essentially, symbols are reusable elements that reside in the Library. You can use them repeatedly in either the same Flash creation or another one entirely. Whenever you create a symbol, it is automatically placed in the Library, where it is stored for future use. When you drag a symbol out of the Library (discussed in the section “Adding a Symbol to the Stage” later in this chapter) onto the Stage, the symbol itself is not placed on the Stage. Instead, Flash creates a copy, called an *instance*, and places it on the Stage. You can change instances as many times or in as many different ways as you want without altering the original symbol. There will always be a copy of the original symbol residing in the Library.

Animation of any significant complexity can be accomplished only with symbols. As a result, it's important that most, if not all, of the graphical elements you create in Flash MX 2004 and Flash MX Professional 2004 are either converted to symbols or created as symbols right off the bat. Don't worry about how symbols are animated with the Timeline; that will be covered in Part III. For now, the only important things to understand are the differences between the various types of symbols.


Symbols fall into one of two categories: native Flash symbols and imported symbols. Native Flash symbols are created directly within Flash, while imported symbols are created in another program and imported into Flash.


In Flash MX 2004 and Flash MX Pro 2004, native symbols are also referred to as *symbol behaviors*. This is due to the fact that, as you'll soon discover, there are three types of native Flash symbols, all of which behave and have different purposes within your movie.

Symbols are beneficial for a couple of reasons. The first, which has already been discussed briefly, is that they allow you to streamline the creative process. When it comes to native Flash symbols, all you need to do is create one object and then manipulate its instance on the Stage to create multiple variations without having to re-create the object from scratch every time. The second reason that symbols are so powerful is that they help you reduce the overall file size of your Flash creation. Each time a symbol is used, Flash simply refers to the profile of the original in the Library. If you're using multiple instances based on a single symbol, Flash needs to save only the information about their differences. If you were to use a different symbol for each of the various objects, each would be included in the Flash file, increasing the overall file size.


Introducing Native Flash Symbols

Let's explore the three different types of native Flash symbols:

Graphic symbols Graphic symbols  are static graphic objects created with the various Flash drawing and painting tools. They work in conjunction with the main movie and are most commonly used as elements in Timeline animations.

Button symbols While the Graphic symbols are static elements, Button symbols  are dynamic, altering their appearance when clicked. Buttons are one of the most popular interactive elements you can create in Flash. Although you'll explore Button symbols (and how you make them) in Chapter 15, it's important to know they are made up of four different static images (referred to as *states*). Each state is visible based on how the user interacts with the button: when the button is up (the *up* state), when the user's mouse is over the button (the *over* state), and when the user clicks the button (the *down* state). The fourth button state, the *hit* state, is not a visible element in the button. Instead, it acts more like a hotspot that determines the active area of the button.

One of the most important things about Button symbols, at least in terms of creating interactivity, is that they can be tightly integrated with ActionScript.

Movie Clip symbols It's no exaggeration to say that Movie Clips  are probably one of the most important aspects of Flash itself. Movie Clips are smaller, self-contained movies that you can place within another movie. They are infinitely nestable so you could have a Movie Clip within a Movie Clip within a Movie Clip, and so on.

Movie Clips, which are in no way limited in their composition, run *independently* of the Timeline. They can also be placed within other symbols, so you could insert a Movie Clip into one of the previously mentioned Button symbol states, creating an animated button.

If you're really interested in finding out how you can create and manipulate Movie Clips, see Chapter 11.

Movie Clips become even *more* powerful when you realize that they can be controlled using ActionScript. As a result, you could have any number of Movie Clips in your movie and send various commands to them (based on either user interaction or some purely non-user-based occurrence in your movie) using ActionScript.

To learn more about ActionScripting for movies with multiple Movie Clips (referred to as *multiple timelines*), check out Chapter 20.

Introducing Imported Flash Symbols

Now that you've looked at the three native Flash symbols, you can move on to look at the various types of imported symbols:

Bitmaps You've already looked at how you import a bitmap into Flash. What you haven't learned is that when bitmaps are imported into Flash, they're automatically inserted in the Library and converted into symbols for your use.

Audio Unlike graphical elements, audio files can't be seen. As a result, when they are imported, they're placed only in the Library (and not also on the Stage like imported bitmaps). Because audio is such a large topic, at this point, there is little more you need to know other than audio files are one of the many types of nonnative Flash symbols.

For more about audio and Flash, see Part VI.

Digital Video One of the great things about Flash is that it lets you import and use a series of different digital video formats. If you have QuickTime 4 (or above) installed on your computer, you can import QuickTime files (MOV), Audio Video Interlace (AVI) files, Digital Video (DV) files, and Motion Picture Experts Group (MPEG) files.

You can also import sound-only QuickTime movie files and place them in the Library as symbols.

If you're working in Windows and you have DirectX 7 (or above) installed, you'll be able to import AVI, MPEG, and Window Media Files (WMV/ASF).

When imported, all digital video files are automatically placed in the Library to be used as symbols.

In addition, both Flash MX 2004 and Flash MX Pro 2004 users can import and use the Macromedia Flash Video (FLV) format.

For more information on importing and working with digital video (especially the new FLV features in Flash MX 2004 and MX Pro 2004), check out Chapter 31.

Creating Native Flash Symbols

There is little doubt that symbols are one of the most important elements of Flash. Mastering their creation and manipulation will open up a world of incredible creative

possibilities for you. In this section, you'll look at the two primary ways to create Graphic symbols: creating from scratch and converting from existing graphics.

Button and Movie Clip symbols are discussed in future chapters, so this section deals exclusively with creating Graphic symbols. For more information on creating Movie Clips, refer to Chapter 11. To learn more about creating Button Symbols, see Chapter 15.

Creating Symbols from Scratch

When it comes to Graphic symbols, it's more than likely that you'll create most within Flash with the painting and drawing tools:

1. Make sure you don't already have an object selected on the Stage.
2. Choose Insert → New Symbol or use the shortcut Cmd/Ctrl+F8. The Create New Symbol dialog box opens (see Figure 7.1).



Figure 7.1
The Create New
Symbol dialog box

You can switch between basic and advanced options in the Create New Symbol dialog box by clicking the Basic/Advanced button. The options available in the Advanced section relate to creating a Shared Library, which is discussed in the section “Working with Shared Symbol Libraries” later in this chapter.

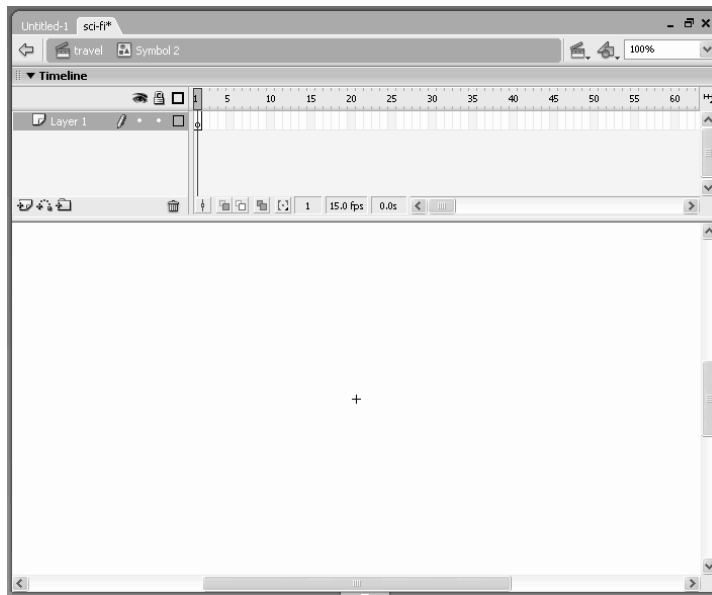
3. In the Create New Symbol dialog box, make sure the Graphic radio button is selected, enter a name for your symbol into the Name field, and click OK.

If your Flash creation will use a great deal of graphics, it's a good strategy to give each symbol a distinct name so you won't get confused. By default, if you don't name the symbol, Flash MX 2004 and Flash MX Pro 2004 will use the name *Symbol X*, where *X* is a sequential number.

4. From here, Flash adds the symbol to the Library and switches you to Symbol Editing mode (see Figure 7.2). The name of the symbol appears in the Edit Toolbar bar, just to the right of the name of the scene you're currently working in. For reference, the symbol's registration point is represented by the crosshairs in the middle of the Symbol Editor.

The Symbol Editor is a little tricky in that it looks almost exactly like the regular Flash environment. Don't get confused, though; you can tell the difference because the symbol's name is displayed in the Edit Toolbar just to the right of the actual scene's name, and a small cross, representing the symbol's eventual registration point, appears on the Stage.

Figure 7.2
Symbol Editing mode enables you to create your symbol with the same painting and drawing tools you use in the Stage.



5. Use the painting and drawing skills you mastered in Chapter 4 to create the desired symbol.
6. When you're finished, you need to exit Symbol Editing mode. To do this, either choose Edit → Edit Document or click the scene name in the Edit Toolbar.

Scenes are a way to break your Flash creation into manageable chunks. To find out more, see Chapter 12.

Converting Existing Objects to Symbols

Although many of your Graphic symbols will be made from scratch in the Symbol Editor, you might come upon a situation where you need to convert an existing graphic (something you didn't create in the Symbol Editor) into a symbol:

1. On the Stage, select the object you want to turn into a Graphic symbol.
2. Choose Modify → Convert to Symbol.
3. When the Convert to Symbol dialog box opens, make sure you select the radio button appropriate for the type of symbol you want to create.
4. Enter a name for your symbol in the Name field.

5. From here, you can set the location of your symbol's registration point. To do this, click one of the nine small boxes in the registration point diagram. Each box represents the location of the symbol's registration point within its square bounding box (see Figure 7.3).
6. When you're finished, click OK and the symbol is automatically added to the Library. The original object that you converted on the Stage is switched into an instance with a parent symbol residing in the Library.

Using the Library to Manage Symbol Assets

The Library, accessible by choosing Window → Library (also by using the shortcut F11 or Cmd/Ctrl+L), is the repository for all symbols in a given Flash movie. Whether you're using Graphic symbols, Button symbols, Movie Clips, or any of the various imported symbols in your Flash movie, the Library is where they'll be. The Library also enables you to preview animations and sound files quickly and easily.

The Library has a second cousin called the Common Library, which you access by selecting Window → Other Panels → Common Libraries. There you'll find four groups of premade symbols—Buttons, Classes, Learning Interactions, and Sounds—that you can use in your own movie. With this basic introduction to the Library, you can begin to take advantage of its power and versatility.

For more information on using the Button Common Library, refer to Chapter 15.

Exploring the Library

The Library is packed with information and tools designed to make managing and manipulating symbols considerably easier (see Figure 7.4). The only way to get a handle on them is to explore each individually.

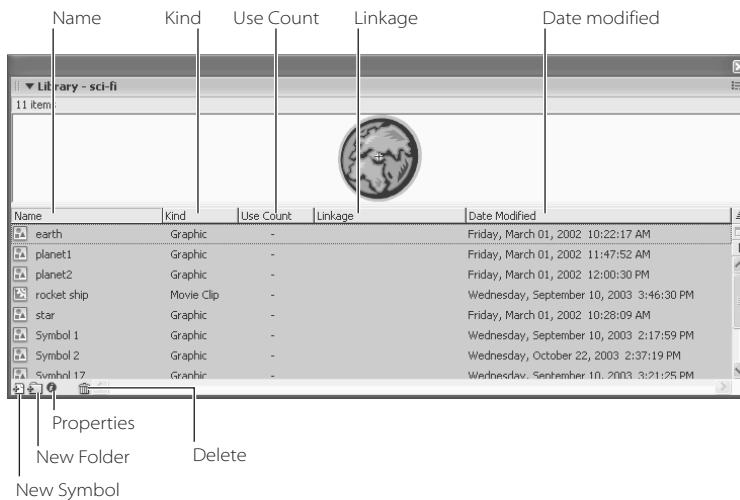
The Sort Window Columns

The Sort window displays information about each symbol in the Library in columns. Under normal circumstances, you can make all the columns visible only if you resize the Library horizontally (by clicking its left edge and dragging). Alternatively, you can also click the Wide Library View button in the right portion of the Library (just above the vertical scroll bar).



Figure 7.3
The Convert to Symbol dialog box; note the Registration Point diagram.

Figure 7.4
The Library



Clicking any aspect of a symbol (name, icon, linkage, date modified, and so on) will display a preview of it.

The Sort window contains the following columns:

Name Shows the name you assigned each symbol. The Name column also displays the filename for imported files such as audio files and bitmaps. The Name column is sorted alphabetically. To reverse the sort order, click the Sort Order button to the right of the Library window.

Kind Indicates whether the symbol is a Button, Bitmap, Graphic, Movie Clip, or Sound. To group items in the library by the type, click the Kind column header.

The type of symbol is also indicated by the small icon to the left of the name.

Use Count Tracks exactly how many times each symbol has been used. This is especially helpful if you're working on a particularly large project and you want to determine which symbols have actually been used in the final movie.

Linkage Indicates whether a symbol is being shared with another movie or imported from another movie.

Date Modified Indicates the last time a symbol or imported file was updated.

Library Panel Buttons

In addition to the Sort window columns, the bottom portion of the Library features several important buttons:

New Symbol Opens the Create New Symbol dialog box, allowing you to create new symbols directly from within the Library.

New Folder Allows you to set up folders in the Library so you can better organize your content. This feature is amazingly useful when working with large, complex movies that need to be well organized. To add an item to a folder, click the item and drag it to the folder. To expand a folder, double-click it.

Properties Opens the Symbol Properties dialog box. From there, you can change any of the selected symbol's properties.

Delete Lets you delete symbols from the Library. Just click the symbol you want to delete and click the Delete button to remove it from the Library.

The Library Options Menu

The Library Options menu, accessible by clicking the menu icon in the upper-right corner of the Library, allows you to manage all aspects of the Library. A subset of these options is also available by Ctrl+clicking (Mac) or right-clicking (Win) a Library item.

The New Symbol and New Folder options were discussed in the previous section.

The following settings are available in the Options menu:

New Symbol Like the New Symbol button at the bottom of the Library, the New Symbol option in the Library's Options menu opens the Create New Symbol dialog box, allowing you to create new symbols directly from within the Library.

New Folder The New Folder option, like the New Folder button at the bottom of the Library, allows you to create a folder that can be used to organize your movie's symbol assets.

New Font Lets you create font symbols that are stored in a shared Library. This is beneficial because you no longer have to embed a font directly in your Flash movie.

To learn more about Font symbols, see Chapter 5.

New Video Inserts an empty Video symbol into the Library. To fill the symbol, double-click the Video symbol's icon. When the Embedded Video Properties dialog box (see

Figure 7.5) opens, click the Import button. From there, navigate to where the desired video file is located, select it, and click OK.

Rename Allows you to rename a symbol directly within your Library. Select the symbol that has the name you want to change, choose the Rename option from the Options menu, and type a new name in the Sort window.

You can also double-click the symbol's name in the Sort window to change it.

Move to a New Folder Automatically creates a new folder on the Sort window into which it will move the currently selected symbol.

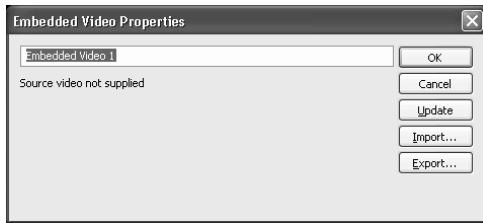


Figure 7.5
The Embedded
Video Properties
dialog box

Duplicate Makes an exact duplicate of the currently selected symbol. Select a symbol in the Sort window, choose Duplicate from the Options menu, and make any adjustments when the Duplicate Symbol dialog box opens.

Delete Works exactly the same as the Delete button at the bottom of the Library. Just select the unwanted Library item and select Delete from the Options menu.

Edit Opens the currently selected Graphic, Movie Clip, or Button in Symbol Editing mode. From there, you can make any changes you want, and the symbol in the Library will change accordingly.

Edit With The Edit With option opens the currently selected bitmap or sound with an external application. If you have Fireworks installed on your machine, Flash launches Fireworks for bitmap editing. Changes made using Fireworks do not affect the original bitmap file you imported; they appear only within Flash. When it comes to audio files in the Library, a dialog box appears, prompting you to choose a sound editor.

Properties Opens the Symbol Properties dialog box for the selected symbol. From there, you can change the symbol's type, edit its name, and manipulate its Linkage properties.

Linkage Lets you set several options needed to create a Shared Library. For more information on creating Shared Libraries, see the section “Working with Shared Symbol Libraries” later in this chapter.

Component Definition Lets you control the parameters of a Flash MX 2004 and Flash MX Pro 2004 UI Component through the Component Definition dialog box.

For more information about Components, see Chapter 16.

Select Unused Items Automatically selects any symbols not currently used in the Flash project.

Update Automatically updates a bitmap or audio file that you've altered in an external program rather than having to re-import it.

If you choose the Edit With option in conjunction with Macromedia Fireworks to make a change to artwork, you do not have to use the Update option; the process is automatic.

Play Plays any selected Movie Clip, Sound, or Button symbol.

You can also play a Movie Clip, Sound, or Button symbol by clicking the Play button in the upper-right corner of the preview window.

Expand Folder Automatically expands any selected folder so that all of its contents are visible.

Collapse Folder Collapses the currently selected folder so that none of its contents are visible—the opposite effect as the Expand Folder.

Expand All Folders Expands all the folders in the current Library so all of their contents are visible.

Collapse All Folders Collapses all the open folders in the current Library so that their contents are not visible.

Shared Library Properties Opens the Shared Library Properties dialog box, allowing you to assign a URL for a given shared Library.

Keep Use Counts Updated Tells Flash to continuously update the use counts.

Evoking the Keep Use Counts Updated option eats up a lot of your computer's processing power, often resulting in a drastic slowdown.

Update Use Counts Now Lets you manually update the use counts.

Using the Library to Work with Symbols

Now you'll take your knowledge about symbols and put the Library to work. In this section, you'll start by looking at how you add symbols to the Stage. You'll also explore how to use various Library options to manage and manipulate your symbols.

Adding a Symbol to the Stage

The Library is a two-way street: You can place symbols on the Stage just as easily as you can add them to the Library.

Remember that when you place a symbol onto the Stage, you're creating an *instance*. An instance is a copy of the parent symbol that you can change and manipulate without altering the original symbol.

To add a symbol to the Stage, follow these steps:

1. Open the Library by choosing Window → Library or use the shortcut F11 (or Cmd/Ctrl+L).
2. Click and drag the desired symbol onto the Stage.

Changing a Symbol's Name

When you initially create a symbol, you give it a name. There are actually a number of different quick and easy ways to change that name:

1. Open the Library (Window → Library, F11, or Cmd/Ctrl+L).
2. Select the symbol that has the name you want to change.
3. Do one of the following:
 - Cmd+click (Mac) or right-click (Win) and choose Rename. When the current name is highlighted in the Sort window, type a new name.
 - Choose Rename from the Options menu. When the current name is highlighted, just type a new one.
 - Choose Properties from the Options menu (or click the Properties button in the bottom of the Library) to open the Symbol Properties dialog box. From here, type in a new name into the Name field.
 - Double-click the symbol name (not the symbol icon) and type a new name.

Duplicating Symbols

Flash enables you to duplicate symbols in the Library with a minimum of trouble. You might find this handy if you've created a complex symbol that you wanted to copy, alter, and then add to the Library. Instead of jumping through a series of hoops, you can duplicate the symbol and then edit the copy in whatever way you want. Also, duplicating a symbol allows you to create a new symbol (to which you can make any changes) without

actually changing any of that symbol's instances on the Stage. Further, it's a great process if you want to make drastic experimental changes to a symbol without risking the original.

To duplicate a symbol, follow these steps:

1. Open the Library (Window → Library, F11, or Cmd/Ctrl+L).
2. Select the symbol you want to duplicate.
3. Do one of the following:
 - Cmd+click (Mac) or right-click (Win) and choose Duplicate.
 - Choose Duplicate from the Options menu.
4. When the Duplicate Symbol dialog box opens, enter a new name. If you want to retain its original format, make sure to select the appropriate radio button. If you want to turn the duplicate symbol into an alternative format, click the desired radio button.
5. Click OK. The duplicated symbol is automatically placed in the Library.
6. From here, you can edit the duplicate symbol and make any changes you want.

Editing a Symbol

You might find yourself in a situation where you want to make more significant changes to a symbol that go beyond changing its name. All you need to do is follow these steps:

1. Open the Library (Window → Library, F11, or Cmd/Ctrl+L).
2. Select the symbol that you want to edit.
3. Do one of the following:
 - Double-click the symbol's icon (to the left of the symbol name in the Sort window).
 - Cmd+click (Mac) or right-click (Win) and choose Edit.
 - Choose Edit from the Library's Options.
 - If the symbol you want to edit is a bitmap or audio file, select Edit With to open the appropriate external editor.
4. When Flash switches to the Symbol Editor, make any changes you want.
5. When you're finished, you need to exit Symbol Editing mode. To do this, either choose Edit → Edit Document or click the Scene button in the Edit toolbar.

When you change a symbol, all its associated instances automatically change as well.

Organizing Your Library

As your experience with Flash develops, you'll find that many of your projects use a large number of symbols. This can certainly lead to some workflow and organizational problems. To avoid this problem, you can do several things.

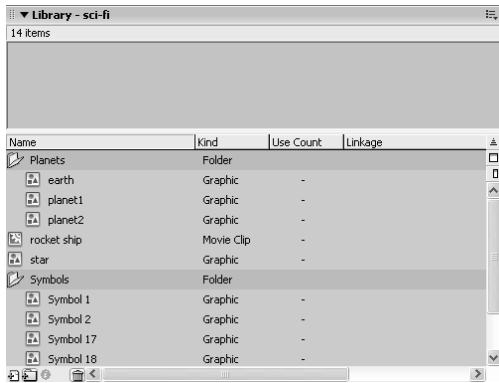


Figure 7.6
You can organize your Library by creating various folders to keep common or similar symbols in.

The first solution involves creating various folders to keep similar symbols in (see Figure 7.6). Just as you can create folders in a website to better organize your files, you can make folders in a Flash Library to better organize your symbols, bitmaps, and sounds.

To create a folder, click the New Folder button in the bottom-left corner of the Library window. You can also choose New Folder from the Library Options menu. From here, simply click and drag the various symbols into their intended folders. Alternatively, you can also select a symbol and choose Move to New Folder from the Library's Options menu.

The second way to organize your Library, which has decidedly less of an impact on the Library's overall organization (but is nonetheless useful), involves using the Sort window columns. By clicking the name of either the Date Modified or Name column, the contents of the Library are reorganized in descending order.

Opening External Flash Libraries

One of the great things about Flash is that you're not limited to the Library being used in your current Flash project; you can also open other Flash movie Libraries by using the Open External Library option.

External Libraries aren't a different kind of file. Essentially, they're just regular FLA files, the Library contents of which you've taken and put in your Library. Because of this, there's no special procedure for creating an external Library; you just need to create a Flash movie like any other.

Follow these steps:

1. Open the Library (Window → Library, F11, or Cmd/Ctrl+L).
2. Choose File → Import → Open External Library.
3. When the Open as Library dialog box opens, navigate to the FLA file that has the Library you want to access, select it, and click Open. The external Library appears as another Library panel in your movie (see Figure 7.7).

If you have not moved your original Library window, the new Library window might open directly on top of the old one. Drag the new Library window until you can see both windows.

4. Click and drag any symbol from the newly opened Library to your current movie's Library. This automatically copies the desired symbol from the imported Library into your current movie's Library, to be used as any other symbol would.

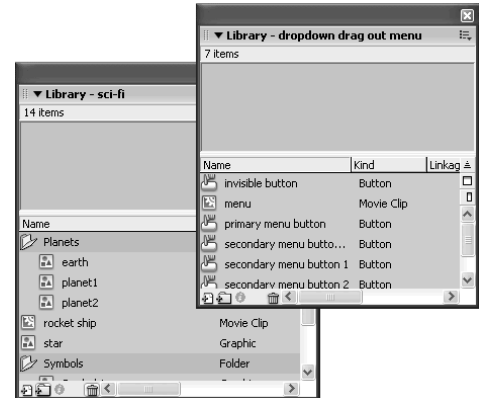
Figure 7.7

The original Library is on the left, and the Library on the right has been opened from an external Flash file.

Working with Shared Symbol Libraries

As Flash development teams become larger and projects become more complex, sharing assets becomes important. With Shared Symbol Libraries, also referred to as *Shared Library Assets* or *Shared Libraries*, you can use graphics, buttons, movies, audio files, and other assets in your movie by linking to the Library of a centrally located SWF file (on a web server, for example). The result is that a Flash movie no longer has to have its own individual Library.

The beauty of Shared Symbol Libraries is that teams can share a standard set of symbols across multiple movies. You can make any final modification to a symbol in the Shared Library. After the Shared Library has been published, any other movies that use the Library are updated automatically. In addition, when you draw your symbols from a Shared Symbol Library, your final movie doesn't need any embedded symbols and therefore is smaller.



Although Shared Symbol Libraries are beneficial to large projects, they have, unfortunately, proven to be somewhat unreliable and unpredictable. We strongly suggest that if you find the need to use Shared Libraries, you limit their use to storage of relatively small (in terms of file size) elements.

Before you learn how to create and work with Shared Symbol Libraries, you need to learn about the two models for Shared Libraries: *runtime* and *author-time*.

- In a runtime Shared Symbol Library, assets are loaded into the destination movie from a source movie when the movie actually plays. Although the source movie (the movie you're drawing the symbols from) doesn't have to be accessible when you initially author the movie, it must be posted to a web server and be accessible for the destination movie to draw the symbols at runtime. Using a runtime Shared Symbol Library is really appropriate only when you're distributing your Flash movie on the Web; this way, it will have access to the actual Shared Symbol Library.

- In an author-time Shared Symbol Library, you can replace any symbol in the movie you're currently working on with a symbol from another movie. Although the symbol in the destination movie retains its original name and properties, its contents are replaced. For example, you could use an author-time Shared Symbol Library when you want to integrate symbols into your movie that were created by someone else. The primary difference between an author-time Shared Symbol Library and a runtime Shared

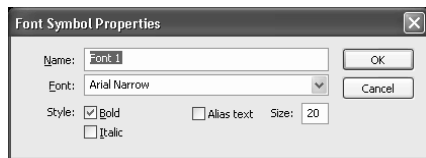


Figure 7.8

The Font Symbol Properties dialog box

Symbol Library is that the source movie must be accessible on your local network (or on your own hard drive). Because you integrate symbols from a Shared Symbol Library into your own movie when it's being created, author-time Shared Symbol Libraries can be used whether you're distributing your movie over the Web or by some other means.

Creating a Runtime Shared Symbol Library

Creating and using a runtime Shared Symbol Library involves two discrete steps. The first involves creating the source movie (in which the Shared Symbol Library resides) and identifying symbols (through the use of unique names) within the source movie. The assignment of unique identifiers for symbols in the Shared Symbol Library is vital so that the destination movie can successfully locate and acquire the symbols it needs to run properly.

If you plan to use runtime Shared Symbol Libraries, get into the habit of using the same name for the unique identifier as you did for the symbol's actual name; this might avoid some confusion in the long run.

The second part, which you do when you create the destination movie, involves telling Flash that the symbols being used will be drawn from a source movie at runtime.

Let's start by looking at how you create a runtime Shared Symbol Library:

1. Create a new Flash file by choosing File → New (or by using the shortcut Cmd/Ctrl+N) and selecting the Flash Document option from the New Document dialog box.
2. Create or import all the elements that you want to include in the final runtime Shared Symbol Library and add them to the currently active movie's Library. For how to create either native Flash symbols or imported Flash symbols, see the section "Understanding Symbols" earlier in this chapter.
3. If you want to include an entire font in your Shared Symbol Library, choose New Font from the Library Options menu. When the Font Symbol Properties dialog box (see Figure 7.8) opens, type a name for the font you want to include, locate the font in the Font menu, and select the style you want to be included.

4. Delete all objects from your Stage that will be included in the final Shared Symbol Library.

Now that you've included all the symbols that will ultimately reside in the final runtime Shared Symbol Library, you need to assign a unique identifier to each so that they can be successfully located and used by the destination movie:

1. Select one of the symbols in the Library that you're working on.
2. From here, you have a couple of different options:
 - Choose Linkage from the Library Options menu or from the Ctrl+click (Mac) or right-click (Win) context menu.
 - Select Properties from the Library Options menu or from the Ctrl+click (Mac) or right-click (Win) context menu. In the Symbol Properties dialog box that opens, click the Advanced button to access the Shared Symbol Library options (see Figure 7.9).

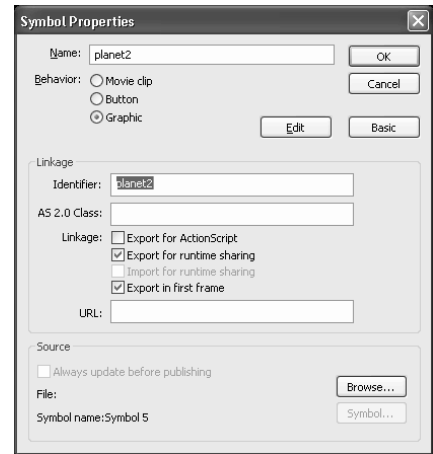


Figure 7.9
The Symbol Properties dialog box with Shared Symbol Library options displayed

If the item that you've selected is not a native Flash symbol (Button, Movie Clip, or Graphic), you won't have access to the properties.

3. Select the Export for Runtime Sharing option.
4. Type a unique name (without any spaces) into the Identifier field.
5. Enter the URL for the place where the source SWF file will be located in the URL field.

The URL that you enter into the URL field must be absolute (for example, `http://someURL.com/movies/mymovie.swf`, not `/movies/mymovie.swf`). If you don't place the source movie in the location you set or enter the correct URL, the destination movie will not be able to locate and draw the necessary symbols.

6. Repeat the process for each symbol that will ultimately be included in the final Shared Symbol Library.

Although the process of tagging each symbol with a unique identifier is tedious (especially if you've got a lot of symbols), it is necessary. Any of the symbols that aren't assigned a unique identifier will not work properly in the final Shared Symbol Library.

7. When you've finished identifying all the desired symbols as assets within the runtime Shared Symbol Library, click OK.

Now that you've set up all the properties of the source movie (which will act as the runtime Shared Symbol Library), save and publish it by following these steps:

1. Save the currently open Flash file (that contains the Library you've been working on) by choosing File → Save. Alternatively, you can use the shortcut Cmd/Ctrl+S. So you don't get confused, make sure you give the file a distinct name that will identify it specifically as containing a runtime Shared Symbol Library.
2. From here, publish the FLA file as a SWF file (Flash's web format). All you need to do is select File → Publish Settings. Make sure that Flash is the only format chosen in the Formats tab. Beyond that, you don't need to set any of the other options.
3. Once you've finished adjusting the necessary Publish Settings, click OK. Then choose File → Publish (or use the shortcut Shift+F12). The SWF file will be saved to the same directory where you saved the original FLA file that you were working on.
4. Take the SWF file (which includes the tagged symbols) and upload it to the server with the URL that corresponds to what you entered into the URL field.

For a far more detailed and thorough look at publishing your Flash movie, see Chapter 30.

Linking to a Runtime Shared Symbol Library

Now that you've gone through the process of creating a runtime Shared Symbol Library (the source movie), you can use its assets in another Flash movie (the destination movie).

The effect of linking to a runtime Shared Symbol Library is something like opening external Flash Libraries. The most important difference is that when you change a symbol in your runtime Shared Symbol Library (source movie), it changes all the linked symbols in the destination movies that use it.

Unless you've got a copy of the symbol you want to draw from the runtime Shared Symbol Library (source movie) in the destination movie's Library, you won't be able to do any linking at all. Although this seems a little counterproductive (what's the point of drawing the symbol from a runtime Shared Symbol Library if it's already in the destination movie?), it's important to realize that the symbol won't actually get exported if it has been linked to a runtime Shared Symbol Library. It really just acts as a placeholder in your movie.

Now follow these steps to link to a Library:

1. Create a new Flash movie (or open an existing one) to act as the destination movie.
2. Open the movie's Library by choosing Window → Library.
3. Make sure that copies of the symbol that you want to draw from the runtime Shared Symbol Library (source movie) are in the Library of the newly created movie. You can do this by opening both Libraries and then dragging the symbols from the source movie into the target movie.
4. Select the Graphic, Movie Clip, or Button symbol, and open the Symbol Properties dialog box by choosing Properties from the Library's Options menu. If the Symbol Properties dialog box isn't expanded, click the Advanced button.
5. Select the Import for Runtime Sharing option.
6. Enter the unique identifier for the symbol you're drawing from the runtime Shared Symbol Library (source movie) into the Identifier field.

If you don't put the exact identifier into the field, the destination movie will not be able to locate the symbol you want to use.

7. Enter the exact location for where the runtime Shared Symbol Library (source movie) is posted into the URL field.

If you don't put in the exact URL, the destination movie won't be able to locate the source movie or access the required symbol.

8. When you've finished, click OK.

Severing the Link to a Runtime Shared Symbol Library

Although linking to a runtime Shared Symbol Library is definitely a handy technique, you might find yourself in a position where you want to sever the link between your destination movie and the source movie. Here's how:

1. Open the destination movie that contains the symbol you want to de-link.
2. Open the Library by choosing Window → Library.
3. Select the Movie Clip, Button, or Graphic symbol you want to de-link.
4. Open the Symbol Properties dialog box by selecting Properties from the Library Options menu.
5. De-select the Import for Runtime Sharing option.
6. Click OK.

Linking to an Author-Time Shared Symbol Library

As mentioned before, when you employ an author-time Shared Symbol Library, you can replace any symbol in the movie that you're currently working on with a symbol from another movie entirely.

When you're working with an author-time Shared Symbol Library, the source movie must be directly accessible at the time you publish your movie, either over a local network or on your own computer.

Remember that although the symbol in the destination movie retains its original name and properties, its contents are replaced by those of the symbol you're drawing from the source movie.

Any asset that the symbol from the source movie uses (such as an audio file in a Movie Clip) is also copied over to the destination movie.

To replace a symbol in a destination movie with one drawn from an author-time Shared Symbol Library, just follow these steps:

1. Open the movie that has the symbol you want to replace.
2. Open the Library by choosing Window → Library.
3. Select the symbol you want to replace.
4. Open the Symbol Properties dialog box by selecting Properties from the Library Options menu.
5. Click the Browse button located at the bottom of the Symbol Properties dialog box.
6. When the Locate Macromedia Flash Document File dialog box opens, navigate to where the FLA file containing the symbol you want to use as the replacement is located, select it, and click Open.

Figure 7.10
The Select Source
Symbol dialog box



7. When the Select Source Symbol dialog box (see Figure 7.10) opens (which lists all the symbols in the FLA file you picked), select the symbol that you want to use as the replacement.
8. Click OK.
9. When you are returned to the Symbol Properties dialog box, click OK again. Note that the contents of the symbol have been replaced.

To replace one symbol for another in the same movie, click the Symbol button in the Source section of the Symbol Properties dialog box to access the Select Source Symbol dialog box.

Working with Symbol Instances

You've already learned that when you drag a symbol from the Library to the Stage, you're not really adding the symbol *per se*. Instead, you're creating a copy (called an *instance*) that you can alter without changing the parent symbol (which remains in the Library). As such, an instance can be changed to look quite different than its parent symbol. Although editing the symbol in the Library updates all its instances, editing an instance of a symbol updates only that instance.

In the next section, you'll investigate how you can manipulate the various visual characteristics of symbol instances.

Modifying the Appearance of Instances with the Property Inspector

Instances have a series of properties that you can manipulate to change the instance's visual character. You'll find that many, if not all, of these instance properties are particularly important when it comes to animation. You can animate each property—including transparency, brightness, and color—to create a change in the instance's visual appearance over time. For example, you could animate an instance's transparency so it appears to fade in over time.

For more information on how to animate an instance's visual properties, see Chapter 10.

This section of the chapter will make you familiar with the tools to change the overall appearance of any given symbol instance.

When it comes to the visual character of an instance property, you'll rely exclusively on the Color drop-down menu in the Property Inspector (see Figure 7.11) to make any changes.

1. Select the symbol instance that has the visual properties you want to manipulate.
2. If the Property Inspector isn't already open, choose Window → Properties.

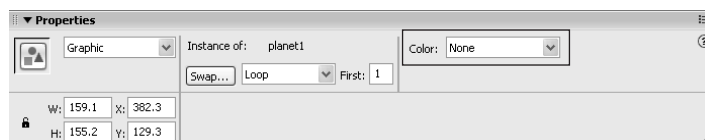


Figure 7.11
The Property Inspector's Color drop-down menu

3. Select one of the options from the Color drop-down menu (each of which has unique properties).

The alterations you make to a symbol instance's visual properties in the Property Inspector will be dynamically applied to the selected symbol instance.

Let's take a look at each option individually:

None Leaves the symbol instance unaltered. Applying the None effect is a good way to revert a symbol instance back to its original form after you've fiddled with its visual properties.

Brightness Changes the relative brightness of the selected symbol instance when you input a value (in percent) into the Brightness field (or use the Brightness slider) (see Figure 7.12). The value can range from 100 percent (white) to -100 percent (black).

Setting the Brightness to 0 percent retains the original color of the symbol instance.

Tint Offers several ways to alter the color of the selected symbol instance. You can click the Tint swatch (which opens the Color Picker) and choose a color. Alternatively, you can mix your own RGB color by entering a value into the R, G, and B fields (see Figure 7.13).

Remember, you can also use the sliders (accessed by clicking the down arrow just to the right of each field) to adjust the value of the individual RGB channels.

Finally, you can adjust the amount of tint by entering a value (in percent) between 0 and 100 into the Tint field.

Figure 7.12
Brightness properties

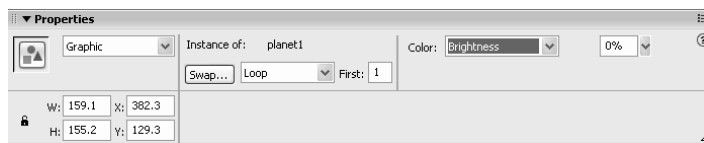
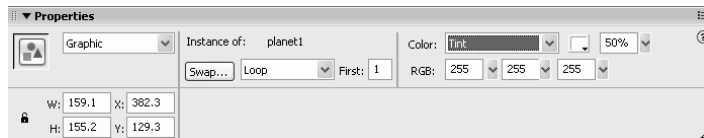


Figure 7.13
Tint properties



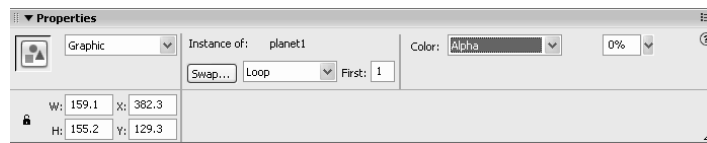


Figure 7.14
Alpha properties

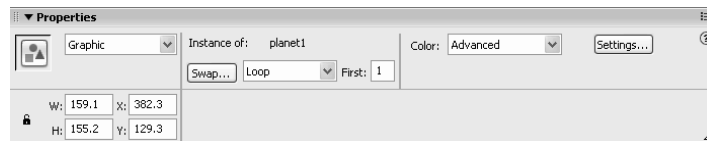


Figure 7.15
Advanced properties

Like the R, G, and B fields, you can manipulate the Tint amount by adjusting the slider (accessed by clicking the down arrow just to the right of the field).

Alpha Lets you adjust the transparency of a selected symbol instance. Enter a value into the Alpha field or use the slider (accessed by clicking the down arrow to the right of the field) to adjust the value. The value can range from 0 (totally transparent) to 100 (no transparency) (see Figure 7.14).

Advanced When you select Advanced from the Color drop-down menu (see Figure 7.15) and click the Settings button in the Property Inspector, you open the Advanced Effect dialog box (see Figure 7.16), where you can change the color and transparency of an object simultaneously.

In the Advanced Effect dialog box, you'll notice that there are two sets of Tint and Alpha controls. The four controls on the right (R, G, B, and Alpha) change the color and transparency of a selected symbol instance by an absolute value. The four controls on the left change a symbol instance's Tint and Alpha value by a relative amount (percentage).

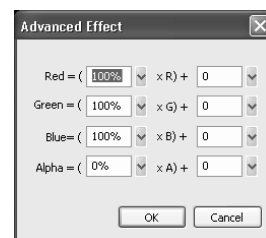


Figure 7.16
Advanced Effect dialog box

Replacing One Instance with Another

You might find yourself in a situation where you want to swap one instance for another. This is particularly useful when you have a complex scene where you want to swap one symbol instance with another but want to make sure the new one is placed in the exact same location as the old one. To do this, follow these steps:

1. Select the symbol instance you want to replace.
2. Choose Window → Properties.

3. Click the Swap button .
4. When the Swap Symbol dialog box (see Figure 7.17) opens, select the symbol that you want to swap, and click OK.

Transforming an Instance on the Stage

Symbol instances are exact copies of their parent symbols in the Library. You can transform them without changing the parent. As such, you're probably going to transform symbol instances on a regular basis. To do this, follow these steps:

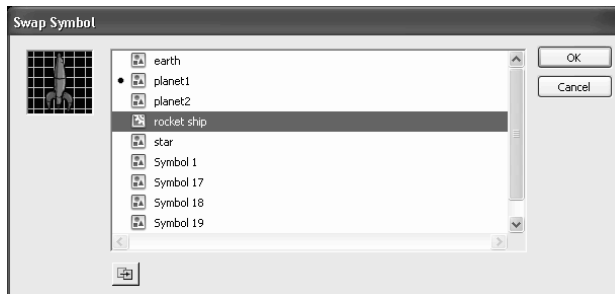
1. On the Stage, select the symbol instance you'd like to edit.
2. From here, use any of Flash's transformation tools to manipulate the symbol instance.

To transform every aspect of the symbol instance, you'll probably have to break it down into its parts—a process accomplished by selecting the symbol instance and then choosing **Modify** → **Break Apart**.

If you want to edit the parent symbol of a given instance, you can open the Library, select the symbol, and choose **Edit** from the Library Options menu. Alternatively, you can double-click the symbol, which opens the Symbol Editor, where you can make any changes you want.

Remember, when you change the parent symbol, all its associated symbol instances also change.

Figure 7.17
The Swap Symbol
dialog box



Inspirational Design Model

Originally located in a small, remote village in the southeastern region of Huizhou in China's Anhui province, Yin Yu Tang is an elegant, rural home. Built in the late Qing dynasty (1644–1911), the house was home to the Huang family for more than 200 years. In 1997, the remaining members of the Huang family, most of which had long since moved away from their ancestral home, authorized the house to be transported across the world to the Peabody Essex Museum in Salem, Massachusetts.

The exhibit itself is complemented by a spectacular interactive experience (mounted on the Web at www.pem.org/yinyutang and on kiosks through the exhibit itself) designed to allow users to explore the house and discover a rare example of the region's architecture and learn about the daily life of the Huang family. Created by Second Story Interactive (www.secondstory.com), the heart of the interactive experience (see Figure 7.18) is composed of five thematic sections that provide a unique lens through which the house can be examined: Orientation, Construction, Ornamentation, Belongings, and Preservation. Content is segmented into distinct "scenes" within each theme; as visitors navigate between scenes, a persistent 3D model view of Yin Yu Tang reacts to reveal different features of the house. A visual interactive family tree, dozens of audio interviews, hundreds of historical and contemporary photographs, and many videos are sprinkled throughout the experience to make this Chinese home "a living house."

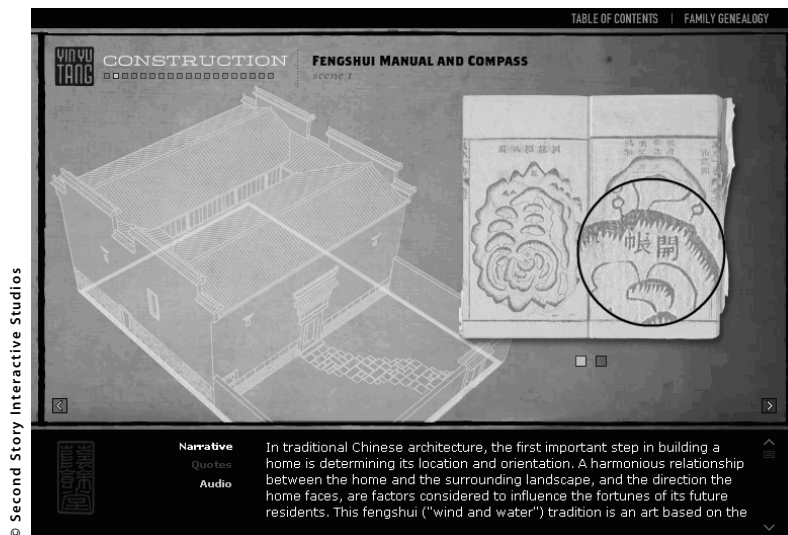


Figure 7.18
Yin Yu Tang

