# 第20章　按钮和导航条

通过学习图像映像和切片技术，我们已经了解到，在 Fireworks中不仅可以编辑图像，而且还可以同时在图像上绑定 HTML代码。然而无论切片也好、图像映像也好，它们在网页中的表现都是静态的，换句话说，它们都作为一幅标准的图像显示在网页中，尽管单击相应的热区和切片可能会跳转到其他地方，但是图片本身没有什么变化。

然而实际在Internet中，我们可以看到大量的具有交互特性的图像。其中最常见的就是按钮，我们经常可以看到这样的情形：当将鼠标移动到一个按钮上方时，按钮图片显示为一种状态；而将鼠标移离按钮上方时，按钮又显示成另一种状态。如果您了解 HTML，您一定知道利用JavaScript可以实现这些特性，然而，如果希望手工编写这些代码，同时将多种状态的图片同代码相关联，工作量是很大的。

我们说过，在Fireworks中不仅可以生成图像，而且可以绑定 HTML代码。连代码都可以生成，那么还有什么同图像相关的操作不能由 Fireworks实现呢？实际上，在阅读了本章和第21章之后，您就会发现，利用 Fireworks可以非常方便地构建多种复杂的按钮行为，以体现用户同按钮的交互性。通过将这些操作同普通的图像操作结合起来，您就可以实现千变万化的动态效果。

这一章，我们主要介绍如何在Fireworks中实现按钮和导航条等Web常见元素。

## 20.1　概述

按钮是网页中常见的元素。我们首先需要了解按钮的一些基本特征。

### 20.1.1　按钮的基本特征

单击按钮可以实现某种行为，或是进行某种操作。例如，单击一个按钮可以进入到一个网站中，或是跳转到另一个页面上。利用按钮在页面中实现导航，是按钮最常见的用途。

最简单的按钮就是一幅图片（当然，为了美观，可能会将图片做成矩形的形状，然后使用诸如内部斜面边等特效来形成立体效果，再在上面显示文字）。但是单击这种图片按钮时，按钮图片本身并不发生任何变化，用户无法了解按钮是否被按下，或是确定单击操作是否生效。特别重要的是，用户不知道这幅图片到底是不是按钮，是不是带有链接，因为将鼠标移动到按钮上时，除了鼠标指针可能变成手的形状之外，没有其他任何特征。

为了解决这种问题，网页中经常使用多个图片来表示按钮的不同状态。例如，可以使用一幅图片来表示按钮正常时的弹起状态，用另一幅图片表示鼠标移入按钮区域的状态，用第三幅图片来表明按钮按下时的状态。通过使用 JavaScript代码来实现按钮图片的这种动态交替，就可以形成非常丰富的动感效果。图 20-1显示了这种情形。

通常，我们将按钮的这种根据鼠标和动作变化而改变状态的特性称作“轮替”(Rollover)，可以说，轮替是按钮最重要的特征。在本章中所讨论的按钮都是具有轮替特性的按钮。为了便于同原始的单图片风格的按钮在概念上相区别，我们还是将那种简单的单图片按钮称作

“图片”，最多是“关联了行为或链接的图片”，而只将具有轮替效果的按钮称作真正的“按钮”。


图20-1 按钮的多种状态

### 20.1.2 Fireworks对按钮的支持

在Fireworks中，即便您对JavaScript一无所知，您也能够轻松创建出多种轮替按钮。它提供了称作“按钮编辑器”的工具，可以引导您轻松地完成对按钮的创建。在按钮编辑器中，您可以像对待单独的对象那样绘制按钮的不同状态，而将这些不同状态的图片组合起来形成轮替按钮的所有操作会全部由 Fireworks在后台完成。而一旦您根据需要修改了按钮中的文字或形状，Fireworks会在您的允许下自动改变其他的状态图片，使不同状态下的按钮保持风格一致。当然，尽管您可以在 Fireworks的帮助下维持按钮中各个状态图片的风格一致，但是您也可以设计使按钮的每个状态图片都保持相互的独立。

在Fireworks中，可以分别设计如下 4个状态的按钮图片，每种状态都对应不同的鼠标动作：

弹起（Up）状态　弹起状态是按钮默认的状态，当鼠标指针没有指向按钮时，按钮就显示为这种状态。

移入（Over）状态　移入状态是当用户将鼠标指向按钮但没有按下鼠标时的状态。

按下（Down）状态　按下状态是用户将按钮按下时显示的状态。导致按钮被按下的原因，可能是用户使用鼠标单击按钮，也可能是通过将输入焦点移动到按钮上，然后按下回车键。

按下时的移入（Over While Down）状态　按下时的移入状态指的是在按钮被按下后，在其上移动鼠标时的状态。

在网页中，大多数的按钮都只有两种状态——弹起状态和移入状态，利用按钮的移入状态，可以提醒用户当前鼠标指针所在的位置有一个按钮，单击鼠标可以进行某种操作。

按钮的按下状态通常用于在网页中显示切换按钮的场合，单击这种按钮时，不是跳转到某个地方，而是表明某种选项被确认，或是表示某种操作已经执行。在导航条中经常使用这种按钮状态。

按下时的移入状态则通常用于告诉用户他的单击操作没有导致任何结果，因为按钮已经被按下了。

在Fireworks中，通过按钮编辑器所构建的按钮，实际上是一种特殊类型的符号。而出现在文档窗口中的按钮，就是该符号的一个实例。在构建按钮之后，按钮符号会出现在文档的库面板中，通过在库面板中将符号拖动到文档窗口中，就可以构建多个按钮实例。关于符号和实例的概念，可以参看本书前面介绍符号和实例的有关章节。

尽管按钮中实际上包含了多个图片，但是这些图片被封装到一起，成为了一个整体。因此，如果您在文档中改变了按钮实例的位置，该实例中所有的状态图片位置都会相应改变。

同样，如果您改变了按钮的某些属性，所有的状态图片属性都会相应变化。

通过将多个按钮组合起来，可以构建所谓的导航条（ Nav Bar)。导航条可以看成是一系列的按钮，用于在一系列具有相同级别的网页间进行跳转。在 Fireworks中，您可以通过复制一个按钮来迅速创建包含多个相同风格按钮的导航条。

在导出按钮时， Fireworks会自动生成用于在浏览器中实现轮替效果所必需的所有代码，包括JavaScript，以及指向各个状态图片的链接。通过将这些代码放入网页中需要按钮或导航条的地方，不需要进行任何修改就可以在网页中实现对轮替效果的支持。

## 20.2 创建和导出按钮

在了解了按钮的基本概念之后，我们可以开始介绍如何在 Fireworks中创建按钮。创建按钮主要在按钮编辑器中进行，通过绘制不同状态下的按钮图片，就可以实现对按钮的创建。

### 20.2.1 新建按钮

要在文档中新建一个按钮，您可以按照如下方法进行操作：

1) 打开“ Insert ”菜单，选择“ New Button ”（新建按钮）命令，这时就会启动按钮编辑器，如图20-2所示。其中显示一个十字线，表明按钮的中心位置。

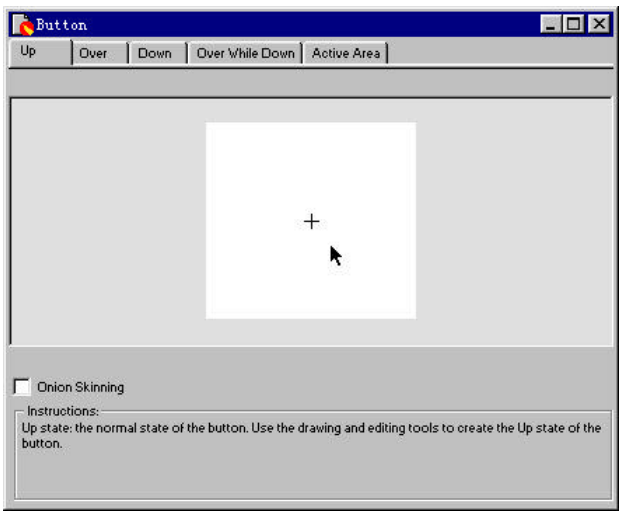


图20-2 按钮编辑器

2) 在按钮编辑器中的“ Up ”选项卡中，首先绘制按钮弹起状态下的图片。例如，可以绘制一个矩形对象作为按钮的基本形状，然后在上方放置一个文本对象，如图 20-3所示。在图中，我们绘制了一个圆角矩形作为按钮主体，并在其上应用了内部斜面边和投影特效生成立体效果。最后将一个红色的“ 进入 ”文本对象放在其上，作为按钮文字。

3) 进入“ Over ”选项卡，绘制按钮的移入状态图像，如图 20-4所示。您可以重新绘制矩形和文本对象，但更好的方法是单击“ Copy Up Graphics ”（复制弹起图像）命令，将“ Up ”选项卡中的按钮图像完整地复制到当前选项卡中，然后在其基础上再进行编辑。例如，这里我们将按钮主体的颜色由亮蓝改为暗蓝。

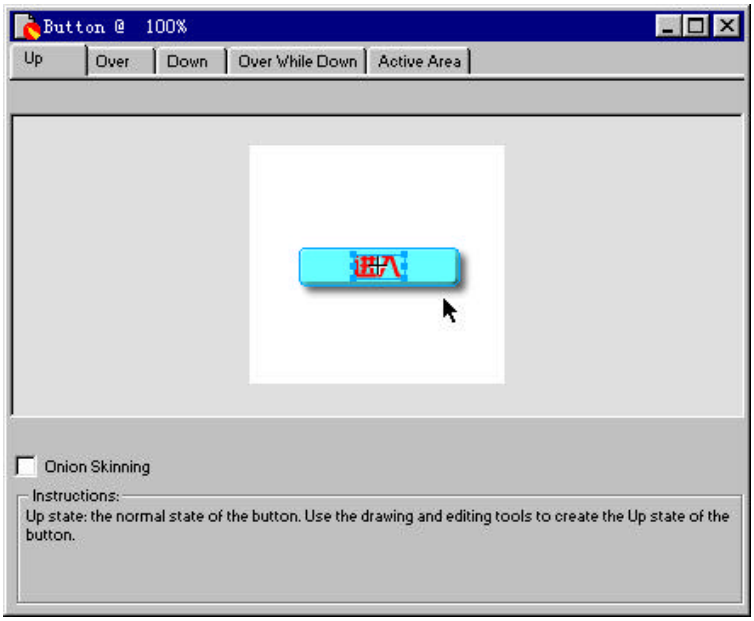


图20-3 绘制按钮的弹起状态

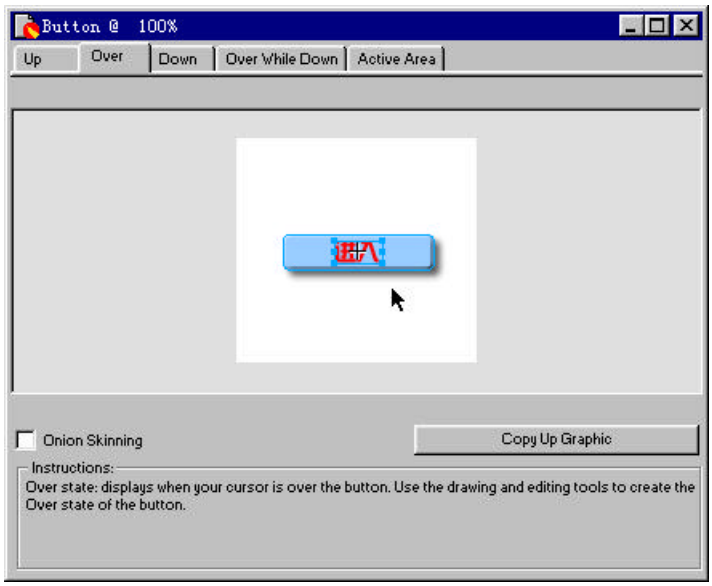


图20-4 绘制按钮的移入状态

4) 继续进入到“ Down ”选项卡中，绘制按钮的按下状态图像。同样，可以通过单击“ Copy Over Graphics ”（复制移入图像）按钮将“ Over ”选项卡中的按钮图像复制到其中，然后再在其基础上进行编辑。例如，我们在这里为了表示按钮的按下状态，取消了按钮矩形对象的投影特效，并将按钮和文本对象一同往右下角移动几个像素的位置，如图 20-5所示。

5) 继续进入到“ Over While Down ”选项卡中，绘制按钮按下时的移入状态图像。同样，

通过单击“ Copy Down Graphics ”（复制按下图像）按钮，可以将“ Down ”选项卡中的图像完整复制到当前选项卡中，然后再在其基础上进行编辑。例如，这里我们将按钮的颜色改为红色，同时将文字的颜色改为暗蓝，如图 20-6所示。

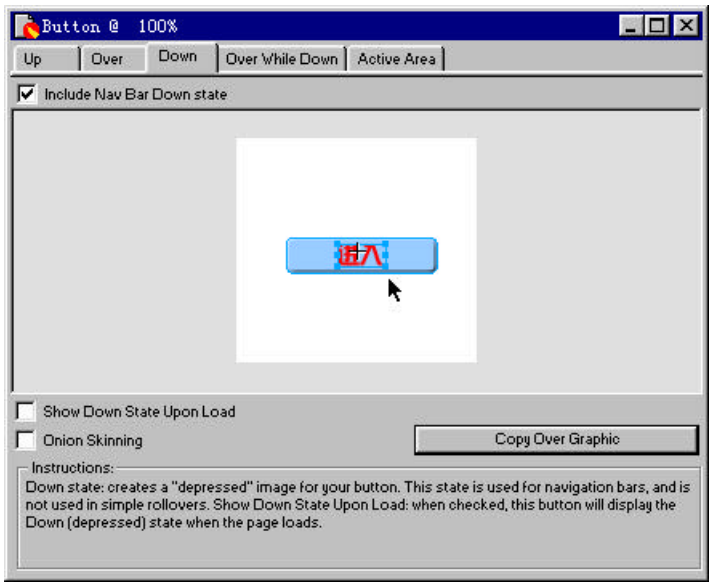


图20-5 绘制按钮的按下状态

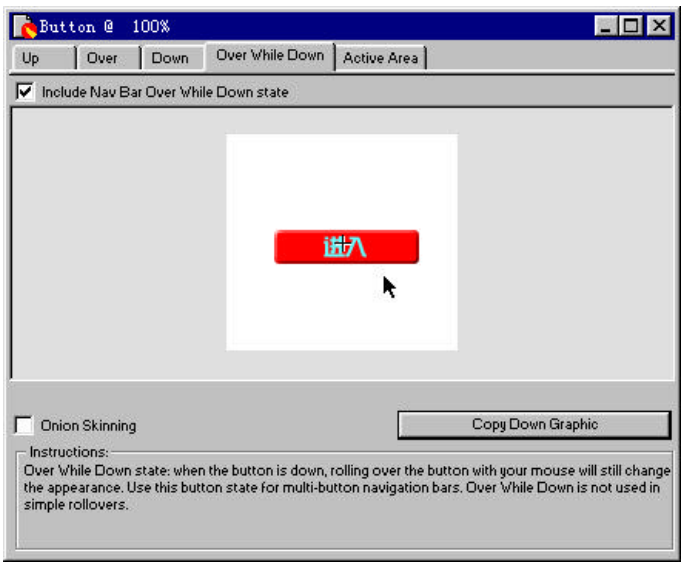


图20-6 绘制按钮的按下时的移入状态

6) 关闭按钮编辑器，这时在文档窗口中就出现新创建的按钮。如图 20-7所示，可以看到该按钮实际上是一个实例。从库面板上可以看到它对应的符号。另外，在文档中添加按钮后，Fireworks会自动将按钮实例制作成一个切片。

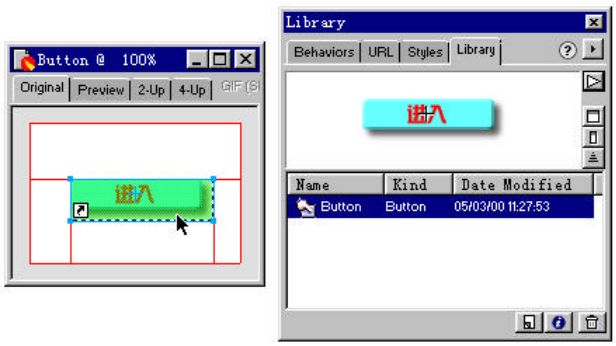


图20-7 文档中新建的按钮以及对应的库面板

**注意** 如果希望在按钮编辑器中的一个选项卡中看到所有按钮状态，可以选中其中的“Onion Skinning”（洋葱皮）按钮。这时当前状态的按钮图像被正常显示，而其他状态的按钮图像以半透明的形式显示。

在“Down”选项卡中，如果选中“Include Nav Bar Down state”（包含导航条的按下状态）复选框，则表明该按钮中可以存在按下状态。这在生成导航条按钮时非常有用。通常用按钮的按下状态表明当前正在访问按钮对应的页面。

在“Down”选项卡中，如果选中“Show Down Upon Load”（载入时显示按下状态）复选框，则默认时按钮处于按下状态。这也主要应用于导航条按钮，通常有一个导航条按钮对应默认显示的当前页面，当载入该默认页面时，相应的导航条按钮在载入之后不用单击，就应该处于按下状态。

在“Over While Down”选项卡中，如果选中“Include Nav Bar Over While Down state”（包含导航条的按下时移入状态）复选框，则表明该按钮中可以存在按下时的移入状态。通常这种状态也应用于导航条按钮中它用于表明当前按钮已经按下，无法再接受单击操作。

如果希望预览按钮效果，可以进入文档窗口的“Preview”等预览窗格，然后用鼠标在按钮上移动或单击以进行测试。必要时，可以将切片隐藏以正确显示按钮。对于上面的例子，可以生成如图20-8所示的结果。


图20-8 生成的按钮效果

在预览按钮时，一旦按钮被按下，则不会自动弹起，除非您按下了另外一个按钮。换句话说，按钮的按下状态主要用于显示切换状态。大多数情况下，设置按钮的弹起状态和移入状态就足够了。这种包含两种状态的按钮，称作简单轮替按钮。

在按钮编辑器中绘制按钮时，帧面板上会出现4个帧，这表明在Fireworks中，实际上是通过将按钮的不同状态的图片分别放入不同帧的方式进行管理的。这种多帧的情形只出现在打开按钮编辑器时，一旦退出了按钮编辑器，则帧面板上会再次恢复原先的状态，换句话说，

您无法利用帧面板直接编辑各个状态图像，而只能使用按钮编辑器。

您可以在按钮编辑器的“ Active Area ”（活动区域）选项卡中设置按钮的活动区域，我们在本章后面介绍相关操作。

### 20.2.2 编辑按钮

如果已经在文档中插入了按钮，而需要修改按钮，这时可以按照如下方法进行操作：

方法一：在文档窗口中，双击要编辑的按钮，重新打开按钮编辑器，然后对按钮的各个状态进行编辑。编辑完毕，关闭按钮编辑器，即可实现对按钮的更新。

方法二：在文档窗口中，选中要编辑的按钮，然后打开“ Modify ”（修改）菜单，选择“ Symbol ”（符号），再选择“ Edit Symbol ”（编辑符号）命令，同样可以打开按钮编辑器来对按钮进行重新编辑。

**注意** 在对按钮进行重新编辑时，如果修改了某个选项卡中的文本对象，会出现如图20-9所示的对话框，提示您选择是否同时更新其他选项卡中对应的文本对象。单击“ Yes ”，则对所有选项卡中对应的文本对象都同时进行修改；而单击“ No ”，则只修改当前选项卡的文本对象。

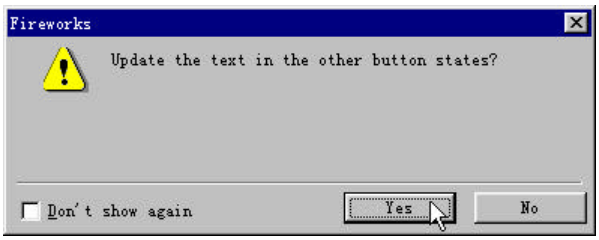


图20-9 更新文本对象的提示

如果仅仅希望修改按钮上的文字，而不是修改按钮的形状，则不必打开按钮编辑器，直接在对象面板上的“ 按钮文字 ”文本框中输入新的文字即可，如图 20-10所示。

按钮实际上就是“ Button ”类型的符号，因此，可以通过从库面板中将按钮符号拖动到文档窗口中的方法绘制多个按钮。要注意的是，如果在文档中生成了一个符号的多个按钮实例，双击某个实例对之进行编辑的时候，就会显示如图 20-11所示的对话框，提示您选择是希望对所有的实例同时进行编辑，还是仅仅只编辑当前的实例。如果希望通过编辑符号来改变文档中所有的实例，可以单击“ All ”（全部）；如果仅仅希望在修改符号后只影响当前双击的实例，可以单击“ Current ”（当前）按钮。

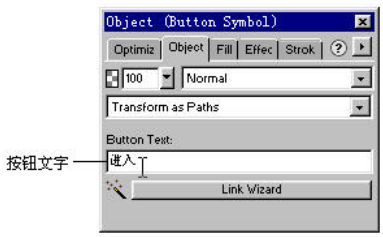


图20-10 修改按钮文字


图20-11 选择要编辑的实例

您可能觉得奇怪，修改符号之后，不是会影响所有的实例吗？的确如此，按钮也不例外，所以，在单击了“ Current ”按钮后， Fireworks就会在文档中根据该实例自动生成一个新的符号，并将该实例同该新建符号关联。所谓对当前实例的编辑，就转变为对该新建符号的编辑，当然，编辑的结果只影响该实例本身，图 20-12解释了这个原理。

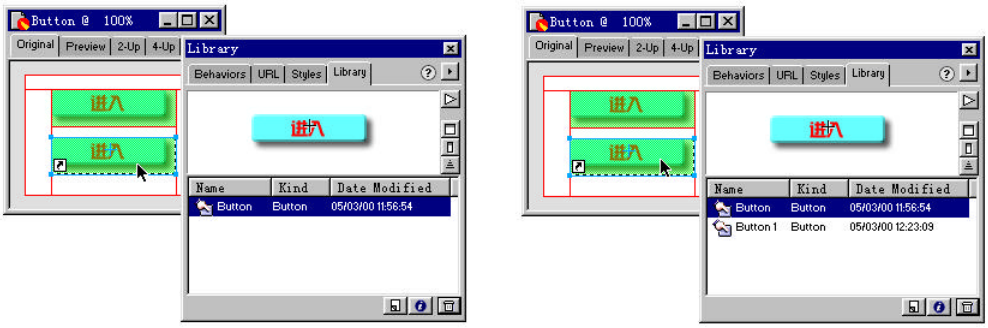


图20-12 编辑单个实例的内幕

### 20.2.3 设置按钮的活动区域

所谓按钮的活动区域，指的就是在按钮上使鼠标移入和单击操作生效的区域。例如，绘制了一个较大的按钮，却只设置当鼠标移动到其中的某个局部区域时才触发轮替效果，这个局部区域就称作活动区域。

假设我们将前面设计的按钮的活动区域设置为右边一半，则可能产生如图 20-13所示的效果。


图20-13 将按钮的右半边设置为活动区域

这看上去有些奇怪，因为没有人会在一个按钮上这样做，但是在某些特殊场合，可能确实需要这种特性。在阅读了第23章之后，您就会明白活动区域的作用。默认状态下，在创建按钮时会将整个按钮作为活动区域。当然，您也可以在按钮编辑器的“ Active Area ”选项卡中重新指定活动区域。可以按照如下方法进行操作：

1) 在按钮编辑器中，进入“ Active Area ”选项卡，如图20-14所示。在新建按钮时，可以直接进入该选项卡。如果希望修改现有按钮，则必须首先打开按钮编辑器，然后再进入。

2) 清除“ Set Active Area Automatically ”（自动设置活动区域）复选框。

3) 活动区域实际上就是切片对象的覆盖区域，您可以按照绘制和修改切片的方法从工具箱上选择切片工具，然后绘制新的活动区域；或是使用相应的方法对现有的活动区域进行编辑。

4) 编辑完毕，关闭按钮编辑器，即完成了对按钮活动区域的设置。

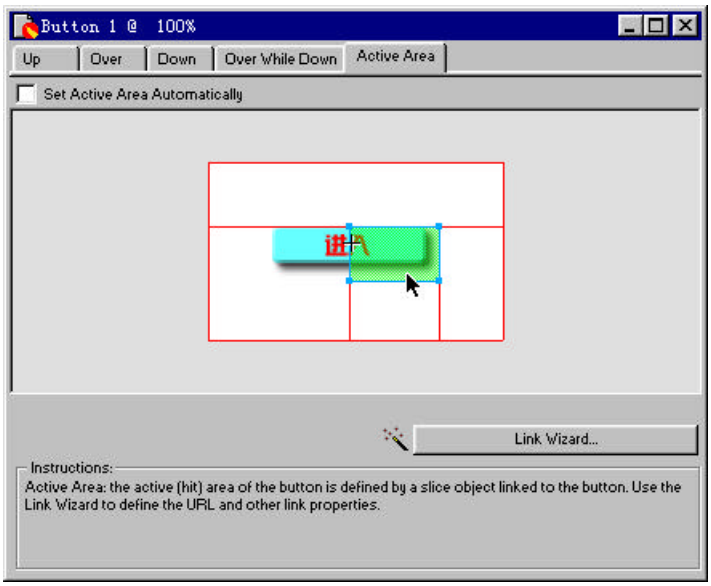


图20-14 设置按钮的活动区域

### 20.2.4 为按钮添加URL链接

按钮不是为了好看才放到网页中的，通过单击按钮，应该可以执行某种操作。按钮上最常见的操作是链接，在Fireworks中，您可以为按钮分派URL地址。

要为按钮分派链接，您可以按照如下方法进行操作：

1) 在文档窗口中选中要添加URL的按钮实例。

2) 在对象面板上，单击“Link Wizard”（链接向导）按钮，这时会显示如图20-15所示的对话框。

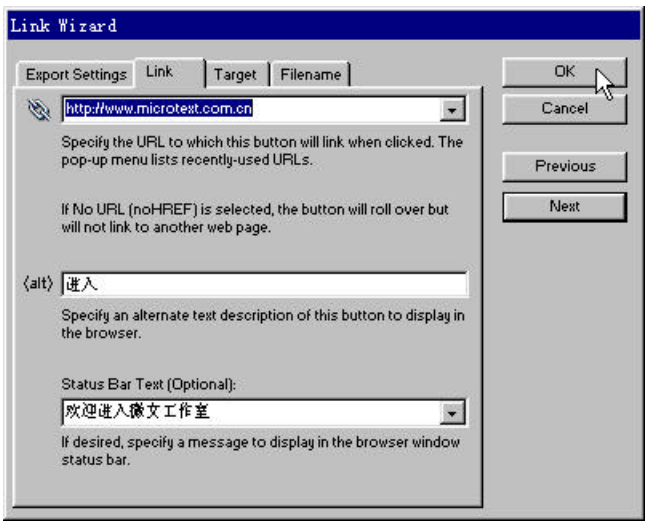


图20-15 设置链接地址

3) 在“Link”选项卡中允许您设置链接地址。

- 在第一个下拉列表中输入需要的URL地址。可以选择现有的URL，也可以直接在其中输入需要的URL。
- 在第二个文本框中输入该按钮的替换文字。
- 在第三个下拉列表框中输入鼠标移动到该按钮上时在浏览器状态行上显示的文字。可以选择现有的状态行文字，也可以直接在其中输入需要的状态行文字。如果保持该项为空，则在浏览器中指向该按钮时，浏览器状态行上显示当前链接地址。

4) 在“Target”选项卡中，允许您指定链接目标的打开的方式。它们的具体含义，请参考介绍热区的章节，如图20-16所示。

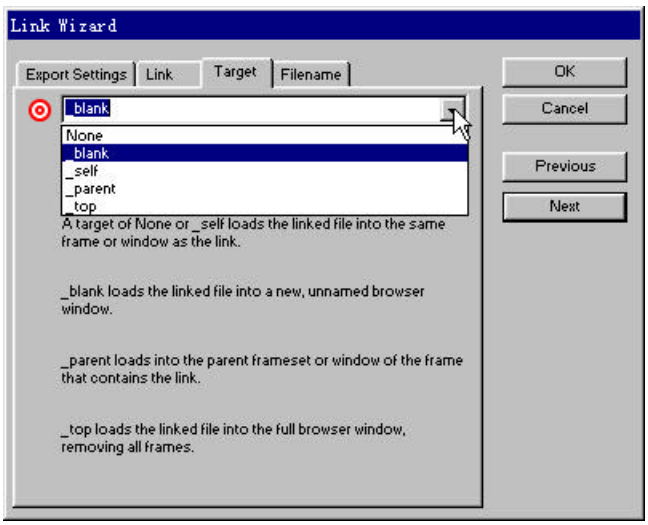


图20-16 设置链接目标

5) 设置完毕，按下“OK”按钮，确定操作，即完成了对按钮的地址分派，同时也指定了链接目标的打开方式。

**注意** 对按钮的地址分派主要是针对按钮上的活动区域而言的。实际上，在按钮编辑器的“Active Area”选项卡中，也可以单击“Link Wizard”按钮，对活动区域进行URL的分派。

在图20-15所示的“Link Wizard”对话框中，单击“Previous”（上一步）按钮可以自动进入到上一个选项卡中；单击“Next”（下一步）按钮可以自动进入到下一选项卡中。当然，我们可能感觉通过直接单击选项卡的标签来进入选项卡更为自然。

## 20.2.5 导出按钮

在文档中完成了按钮的创建之后，最终需要将之导出，才能真正在Web中使用。您可以按照常规的导出切片的方法，将按钮导出。

这里有几个需要补充的地方：在对象面板上单击“Link Wizard”按钮，打开链接向导对话框时，在“Export Settings”（导出设置）选项卡中可以为按钮选择预设的优化方案，如图20-17所示。单击“Edit”按钮，还可以直接打开导出预览对话框，允许您进一步控制对按钮的优化设置。

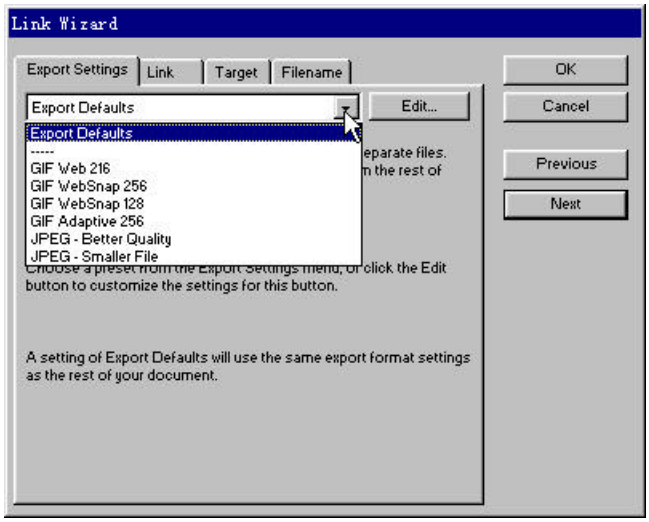


图20-17 设置按钮图片的导出选项

在该对话框的“ Filename ”（文件名）选项卡中，可以设置按钮被导出时各个状态图片的名称，如图20-18所示。选中“ Auto-Name ”复选框，则有系统自动为按钮图片命名；如果清除该复选框，则可以在下方的文本框中自行指定其名称。

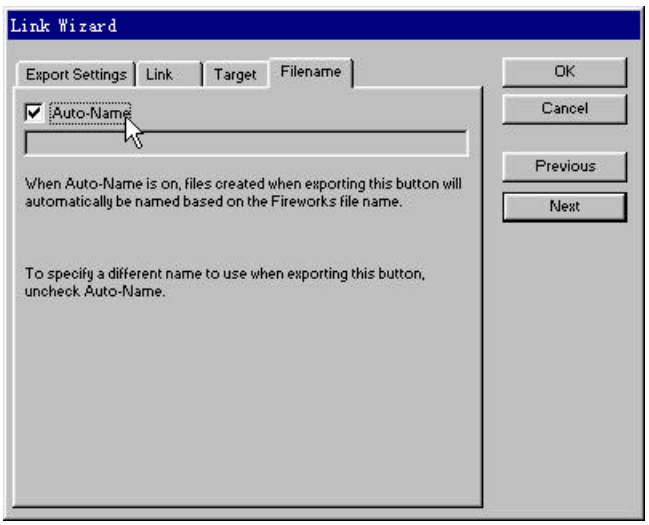


图20-18 设置按钮图片的命名方式

## 20.3 绘制导航条

所谓导航条，实际上就是一组分别指向不同链接地址的按钮。在网页中，单击相应的按钮，就可以实现相应的跳转。一般来说，无论在站点中如何跳转，导航条按钮始终出现在屏幕上（通常将它们放入另外的框架中，以便保留在屏幕上），因此，无论何时，您都可以通过单击相应按钮而在两个页面之间直接切换。由于这些按钮起到了很好的导航作用，因此被称

作导航条。

通常情况下，您可以在文档中通过复制按钮的方法来快速构建导航条。在复制了按钮之后，分别为这些按钮指派不同的URL地址，就可以完成导航条的绘制。

创建导航条

一般来说，导航条上各个按钮的形状应该相同，但是文字可以不同，创建导航条的核心在于如何实现对按钮形状的同时更改，而不涉及按钮上的文字。这可以通过将按钮主体作为一个符号，并将按钮主体的实例连同文字作为另一个符号的方式来实现。

要创建简单的导航条，您可以按照如下方法进行操作：

1) 首先绘制一个按钮，但是不要在按钮中包含文字。该按钮将用作导航条上的按钮。完成按钮的绘制后，它会出现在库面板中。假设这里我们绘制了如图 20-19所示的按钮，并将之命名为“MyNavButton”。

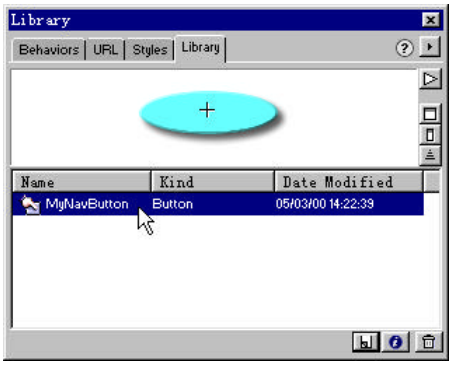


图20-19 将用于导航条上的按钮

2) 通过打开“Insert”，选择“New Button”命令，打开按钮编辑器。

3) 从库面板上，将刚刚创建的导航条按钮拖动到刚刚打开的空白的按钮编辑器中，如图 20-20所示。现在，在文档中出现了两个图层，其中新出现的图层被共享，且按钮已放入到共享图层中。从图层面板中您可以看到这种变化，如图 20-21所示。

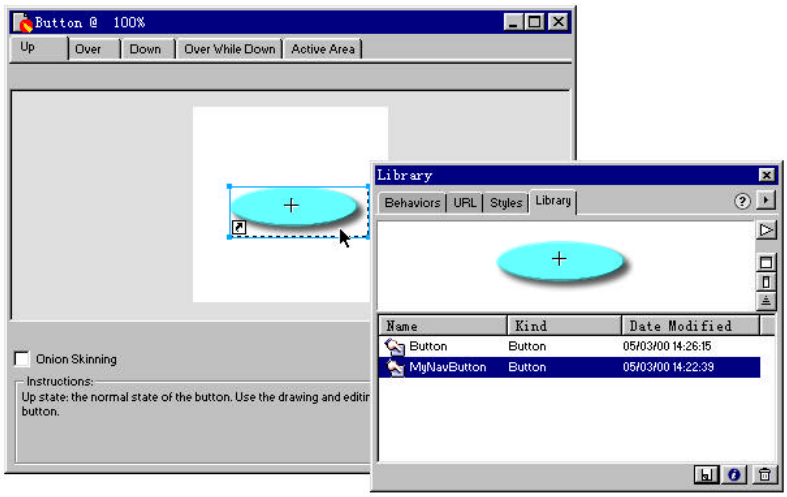


图20-20 将导航条按钮从库面板上拖动到新的空白的按钮编辑器中

4) 在图层面板中，选中没有被共享的图层，然后在按钮编辑器中添加文本对象（用于设置按钮上的文字），如图20-22所示。还可以进入其他的几个选项卡，分别设置该文字在相应按钮状态下对应的外观。

5) 如果需要，可以在按钮编辑器的“Down”和“Over While Down”的各个选项卡中，分别选中位于上方的“Include Nav Bar Down State”复选框和“Include Nav Bar Over While Down State”复选框，表明在按钮中包含了按下和按下时移入两个状态。

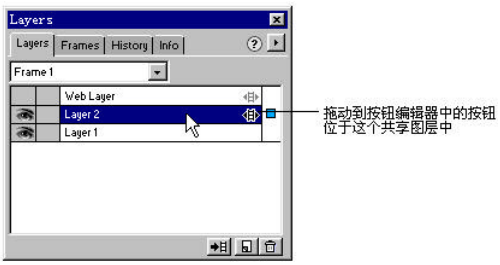


图20-21 图层面板中的共享图层

6) 关闭按钮编辑器，这时新创建的按钮将出现在库面板中，如图20-23所示。这个按钮，就是我们最终需要的导航条按钮。

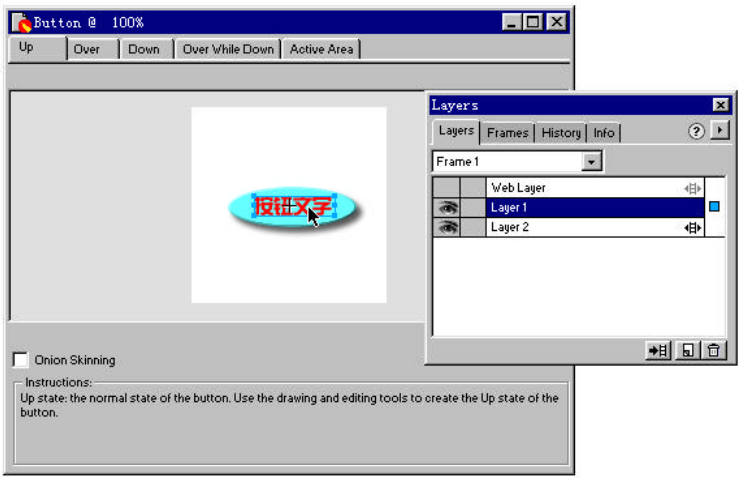


图20-22 设置按钮文字

7) 在文档窗口中，可以通过从库面板上将新设计的导航条按钮拖动到文档窗口中的方式，复制出多个导航条按钮实例，并根据需要修改每个按钮上的文字，如图 20-24所示。


图20-23 完成导航条按钮的编辑

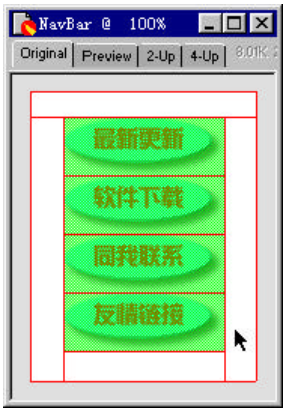


图20-24 新建的导航条

在这个例子里，我们得到的导航条效果最终如图 20-25所示。

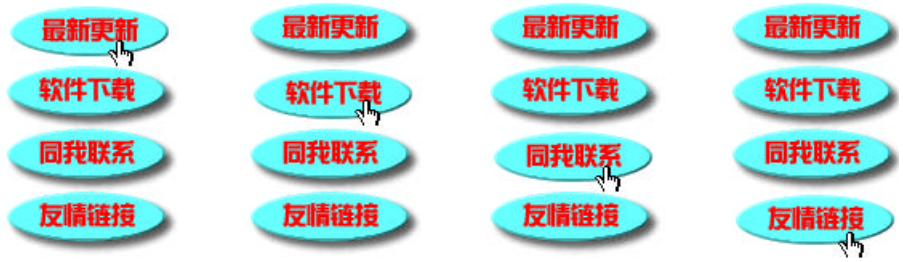


图20-25 创建的导航条

**注意** 双击文档中的各个导航条按钮以编辑其中文字时，会出现如图20-11所示的对话框，提示您选择修改所有实例还是修改当前实例。请单击“ Current ”修改当前实例，这时会将当前按钮制作成为一个新的符号，并出现在库面板中，如图20-26所示。

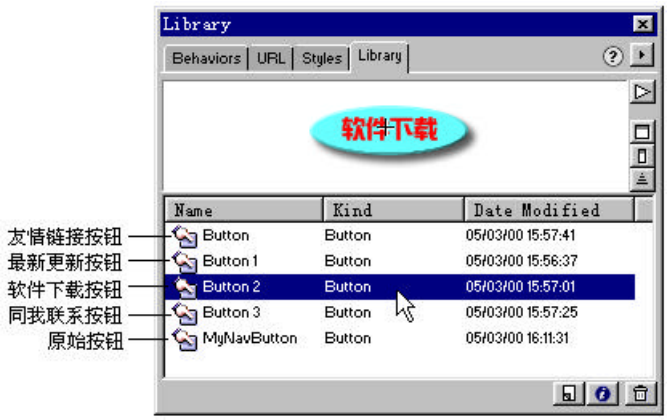


图20-26 当前的库面板

有人可能会说，这么麻烦，直接设计多个按钮符号，然后分别生成实例以构建各导航条按钮岂不更简单？然而这种分别从不同符号派生按钮的方法，不利于对导航条按钮进行统一的更新操作。我们前面介绍的操作核心，在于将导航条按钮的按钮主体（也即不带文本的按钮）作为一个符号插入，这样使得直接修改导航按钮主体符号，就可以快速改变所有导航条中按钮的形状。例如，这里我们对库面板上名为“ MyNavButton ”的符号（它是我们第一个绘制的按钮符号，也即那个没有文字的按钮符号）修改成如图20-27所示的样子，修改的结果就会影响所有的导航条按钮，如图20-28所示。

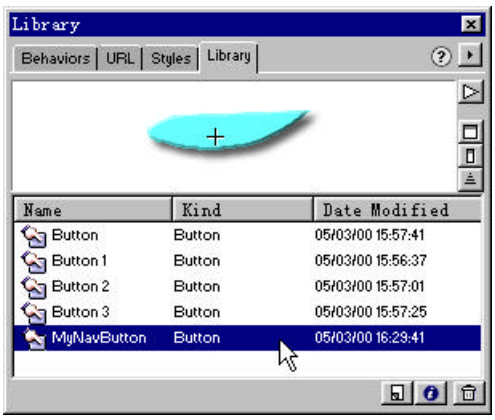


图20-27 修改原始的按钮

在导航条按钮中，按下状态是比较重要的。因为通常导航条按钮会始终出现在屏幕上，当

**您单击某个按钮时，该按钮就保持按下状态，表明已经访问了某个地方，直到在导航条中单击另一个按钮，原先按下的按钮才会弹起。**

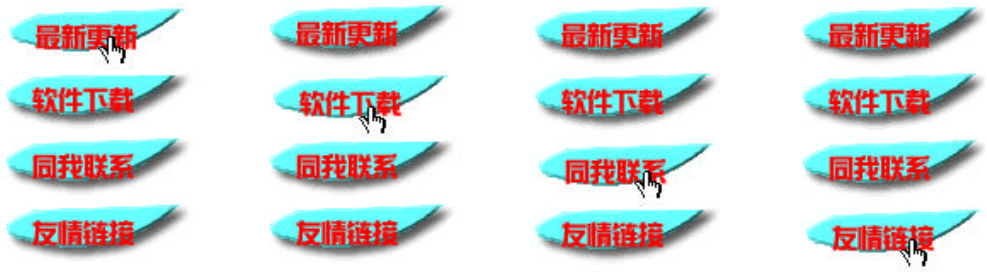


图20-28 同时改变所有的导航按钮