

news
news
pulse
current
news



news

pulse

news



http://www

http://www

Flash ActionScript 权威指南

Phillip Kerman 著

陈冰 译

SAMS



中国电力出版社

726

726

Flash ActionScript

权威指南

Phillip Kerman 著

陈冰 译

中国电力出版社

内 容 提 要

本书内容共分为三部分。第一部分系统地、由浅入深地讲述 ActionScript 编程的基础知识。在这个部分中有大量的范例来说明这些知识和技巧。第二部分是专题讨论,它们并不很难,但却能够为你解决问题提供一种思路。内容全面翔实,可读性较高。第三部分为附录,介绍了 Flash 5 不再支持的 ActionScript 组件的替代方法等。

本书适合网页设计、制作人员阅读,也可作为相关人员的培训教材使用。

图书在版编目(CIP)数据

Flash ActionScript 权威指南/(美)科曼 著;陈冰译. -北京:中国电力出版社,2002.4

ISBN 7-5083-0984-7

I . F... II . ①科...②陈... III . 动画-设计-图形软件, Flash ActionScript IV . TP391.41

中国版本图书馆 CIP 数据核字(2002)第 016411 号

著作权合同登记号 图字:01-2002-1437 号

本书英文版原名: ActionScripting in FLASH

**Authorized translation from the English language edition, published by Sams
Publishing Copyright©2001**

All rights reserved.

本书中文版由美国培生集团授权出版,版权所有。

中国电力出版社出版、发行

(北京三里河路6号 100044 <http://www.infopower.com.cn>)

北京市地矿印刷厂印刷

各地新华书店经售

*

2002年5月第一版 2002年5月北京第一次印刷

787毫米×1092毫米 16开本 32印张 729千字

定价 55.00 元

版 权 所 有 翻 印 必 究

(本书如有印装质量问题,我社发行部负责退换)

前 言

升级到 Flash 5 将把一个已经吸引了众多期待目光的媒体（Flash 动画）带入到一个真正令人吃惊的境界。内含 ActionScript 语言的新版本意味着 Flash 5 现在没有什么不能实现了。如果你能组织你的目标并将其翻译成程序员所使用的语言，那么让 Flash 按你希望的方式工作并不是一件难事，但在这里有一个问题：将目标翻译成一种编程语言并不是每个人都能够做到的。

本书瞄准了这样的一些读者，他们能够装配基本的 Flash 电影并清楚他们想要达到的目标。我将帮助你目标分成能够被翻译成 ActionScript 代码的一个个单独的任务。自然，这将涉及教你如何“编程”（像一个程序员那样来思考问题）。但不管怎么说，本书不是一本普通的编程书，因为其中的每一个主题都与 Flash 密切相关并最终都将应用到 Flash 上。当然，如果你已经是一个经验丰富的程序员的话，你可能会感觉本书中的某些章节对你来说只是一种知识的回顾。但对于作为程序员的你来说，我将展示如何应用你以往的编程知识来完成 Flash 的任务。当然，话说回来，如果你没有将这些知识应用到 Flash 中的话，那么所有的编程技巧（不管是我教给你的还是你自己掌握的）都不会对你有任何帮助。

毫无疑问，本书没有逃避 Flash 编程的高级话题。但不管怎么说，本书也没有详细地论述了 ActionScript 语言的每一个细节。事实是你几乎无法论述完所有那些高级的编程话题，看看那些关于高级编程话题的书和各种资源有多少你就明白了，这不是本书的目的。本书的目的在于给你必要的技巧以便你能够将任何你所产生的想法应用到 Flash 上。当你用我所涉及的这些知识武装起来的时候，你将有能力接受任何挑战。在你的开发过程中，你可能会涉及研究一个深奥的物理公式或应用一个特别的数学计算。如果那意味着必须去研究一个特别的话题的话，本书为你提供的技巧将帮助你想出将其应用到 Flash 中的方法。

我开始写这本书正是在完成了《Sams Teach Yourself Macromedia Flash 5 in 24 Hours》一书之后。但本书并不是前面那本书的延续，两者之间也没有什么重复的内容。我前面说过了，你需要能够装配简单的 Flash 电影（例如：我不会教你如何在 Flash 中进行绘图）。本书的第 1 章，“Flash 基础”就已经越过了这个前提条件。因此，具备这些基本的知识对你来说是很重要的。别担心，我不会跑得太快以至你跟不上我的脚步，但阅读本书的每一位读者都应该具备这样的起点。

本书由两部分组成。基础部分各章看起来就像是一本教科书。大量的例子散布其中，但你不需要不停地在 Flash 中运行它们。（当然我确实怀疑一旦灵感产生，你会经常想要尝试运行某些东西。）专题讨论的各章都是一些内行的讲座。这为你应用在基础部分中学到的知识提供了一个机会。如果你喜欢的话，当需要某些问题的进一步的阐述和说明时，直接跳到相关的专题讨论将对你大有帮助。在专题讨论中的某些部分，我甚至会带领你沿着错

误的路线走下去，以便我们可以讨论沿着这样方向的解决方案会出现怎样的一种情形。我发现这比某些其他风格的讲座会更加贴近现实的世界，那些讲座看起来只能证明“可能”只需要很少的几步就能够达到一个特定的结果。但真实的生活经常是充满了挫折感的，或许这些专题讨论所模拟出的情景能够在继续自己的工作时帮助你避免受挫。

在我们开始前进之前还有最后的一点需要说明。由于 Flash 5 和 Flash 4 相比有了相当大的改变，因此我决定仅讨论 Flash 5。在本书中，你会发现包括附录 A “等价物”在内的部分章节对 Flash 4 有所涉及。但总的来说，本书是为 Flash 5 而写的。第一个专题讨论就提供了关于如何确信你的用户是否使用了正确的 Flash 播放器的信息。当然，你也将获得使你的用户升级的相关知识，升级以便他们都能够看到你的 Flash 5 作品。

现在，准备迎接从一个 Flash 用户到一个 ActionScript 程序员的转变吧！

目 录

译者序
前言
关于作者

第一部分 基 础

第 1 章	Flash 基础.....	3
1.1	Timeline 的体系结构.....	3
1.2	脚本的位置.....	9
1.3	永远都是电影剪辑.....	11
1.4	坐标系统.....	12
1.5	诀窍.....	13
1.6	小结.....	17
第 2 章	Flash 5 中有些什么新东西.....	18
2.1	使编程变得轻松的先进功能.....	18
2.2	需忘记的先前知识.....	23
2.3	小结.....	27
第 3 章	程序设计师的方法.....	28
3.1	说明书.....	28
3.2	原型设计.....	29
3.3	好的风格.....	31
3.4	分离代码和数据.....	34
3.5	小结.....	35
第 4 章	Flash 的基本编程.....	36
4.1	术语、特别字符和格式.....	36
4.2	数据类型和变量.....	45
4.3	使用变量.....	50
4.4	点语法.....	55
4.5	小结.....	55

第 5 章	编程结构.....	57
5.1	语句、表达式和操作符.....	57
5.2	语句中的简单对象.....	69
5.3	条件和循环语句.....	75
5.4	实用的表达式编写方法.....	86
5.5	小结.....	89
第 6 章	调试.....	90
6.1	接近无 Bug 生活的普通途径.....	90
6.2	使用调试器.....	96
6.3	小结.....	111
第 7 章	Movie Clip 对象.....	112
7.1	剪辑的特性.....	112
7.2	剪辑中的变量（或“自创特性”）.....	115
7.3	剪辑的方法.....	117
7.4	引用剪辑和瞄准.....	120
7.5	小结.....	126
第 8 章	函数.....	127
8.1	如何使用函数.....	127
8.2	创建自创函数.....	131
8.3	函数的概念.....	143
8.4	小结.....	146
第 9 章	选择文本、捕获键盘输入及操纵字符串.....	148
9.1	String 对象形式.....	149
9.2	探究 String 对象的方法.....	151
9.3	使用 HTML 文本.....	157
9.4	Selection 对象.....	159
9.5	Key 对象.....	162
9.6	小结.....	169
第 10 章	数组.....	170
10.1	数组概述.....	170
10.2	数组的创建和操纵.....	173

10.3	联合数组.....	188
10.4	小结.....	191
第 11 章	对象.....	192
11.1	对象的正式规则.....	192
11.2	捆绑声音.....	193
11.3	Color.....	198
11.4	Date.....	203
11.5	捆绑 Movie Clip.....	209
11.6	小结.....	210
第 12 章	自创对象.....	211
12.1	基本对象.....	211
12.2	自创对象的实际例子.....	221
12.3	小结.....	229
第 13 章	Smart Clip.....	230
13.1	标准 Smart Clip.....	231
13.2	用定制 UI 替代 Clip Parameters 面板.....	239
13.3	小结.....	244
第 14 章	连接外部数据.....	245
14.1	外部脚本.....	245
14.2	外部数据文件.....	247
14.3	服务器脚本.....	255
14.4	XML.....	257
14.5	JavaScript.....	266
14.6	Director 和 Authorware.....	276
14.7	小结.....	285

第二部分 专 题 讨 论

A	初级专题讨论.....	291
第 15 章	确保用户拥有 Flash 5 播放器.....	293

第 16 章	伪造视频.....	300
第 17 章	创建定制光标.....	307
第 18 章	创建水平滑块.....	320
18.1	创建快速而令人讨厌的滑块.....	320
18.2	将滑块转换成 Smart Clip	324
18.3	小结.....	331
第 19 章	建立幻灯片放映.....	333
第 20 章	映射.....	341
第 21 章	使用不规则可点击区域.....	350
第 22 章	修改内建的 Smart Clip	357
B	中级专题讨论.....	365
第 23 章	创建货币兑换计算器.....	367
第 24 章	创建工具提示 Smart Clip	385
第 25 章	创建数字计时器.....	397
第 26 章	创建模拟计时器.....	405
第 27 章	创建倒计时计时器.....	411
第 28 章	使用 Math 创建圆形滑块	417
第 29 章	开发基于时间的动画.....	424
第 30 章	创建多状态按钮.....	429

C 高级专题讨论	439
第 31 章 脱机生产	441
第 32 章 创建动态幻灯片演示	445
第 33 章 创建 JavaScript Cookie	458
第 34 章 编写 JavaScript 来建立电影间的通信	465
第 35 章 修改损坏的脚本	472
35.1 纸牌翻转	472
35.2 纸牌叼咬	474
35.3 圆周运动	475
35.4 多选择	476
35.5 旋转盒子	476
35.6 字浮动	478
35.7 黄色盒子	479
35.8 点击和按住	480
35.9 对象	481
35.10 移动多个	483
35.11 小结	484

第三部分 附 录

附录 A 等价物	487
A.1 对遭到反对的代码的替代	487
A.2 不再稀奇古怪	490
A.3 本书中老掉牙的一些东西	491
附录 B 为 Macromedia Exchange 网站构造 Flash 扩展	493
B.1 它如何工作	493
B.2 MXI 文件格式	494

第一部分

基 础

下面的 14 章几乎涵盖了 ActionScript 的所有细节。尽管内容看起来像是一本教科书，但我在其间也贯穿了一些例子。你可以暂时中断阅读去尝试本书中任何你感兴趣的主体，但不管怎么说，这就是本书中你要阅读的第一部分。在本书的后半部分（专题讨论），你将看到 21 个详细的讲座。

下面是本书的这一部分中要探究的所有主题的一个核心纲要。

第1章 “Flash 基础”中包含了一些阅读本书的先决条件。我期待每一位读者都读一下。即使你认为你是一个 Flash 专家，你也应该读一下本章以便做一个回顾，并了解我贯穿全书将使用的一些术语。

第2章 “Flash 5 中有些什么新东西”中介绍了一些与 ActionScript 有关的 Flash 5 的关键特征。除了编程语言的变化以外，那些将对你创建 Flash 电影产生影响的特征也将得到讨论。本章中也将包含一个关于容易忽略的先前知识的快速纲要（那些在 Flash 的旧版本中已经习惯的知识）。

第3章 “程序设计师的方法”是非常普通而概括的一章，它将为你的编程生涯打下地基。本章中的话题将涉及编程规范、标准和正确的风格。

第4章 “Flash 的基本编程”为你介绍了 ActionScript 的术语和基本元素，例如数据类型和变量。讨论这些元素而不展示它们是如何工作是说不过去的，但这一章的目的只是让你知道所有这些将在以后的章节中被组织成一个有机的整体的片断究竟是些什么东西。

第5章 “编程结构”是非常庞大的一章，该章阐述了构造 ActionScript 的所有方法。如果说第 4 章是建筑的原料，那么本章就是建筑的构架和建筑的风格。你也可以浏览一下 Math 和 Number 对象，因为它们对结构元素必不可少。

第6章 “赋值”为你在经历了第 5 章的紧张刺激之后喘口气提供了一个机会，同时你将花些时间学习如何确保高质量编程的多种方法，以免你在错误的方向上的走得太远。新的 **Flash Debugger** (Flash 调试器) 将在这里讲解，一些通常的用于避免和排除错误的编程技巧也将在这里提到。

第7章 “**Movie Clip** 对象”介绍了一个我们所熟知的 Flash 组件，这也将有助于你理解在后面的章节中将会到来的其他对象。通过这种方法，可以利用现有的知识举一反三掌握更高级的话题。

第8章 “函数”将展示给你如何使用 Flash 内建的函数，以及如何编写自创函数。结果表明自创的函数远比 Flash 自带的函数要复杂得多。这恐怕是最有价值的一章了，因为它将为你节省大量的时间。

第9章 “选择文本、捕获键盘输入及操纵字符串”介绍了控制屏幕上的文本（例如 **HTML** 文本和输入文本）及从屏幕中删去的文本（例如在用户看到它们之前清理字符串）的多种方法。你也将学习如何使用 **Key** 对象来“捕获”用户通过键盘输入的信息。

第10章 “数组”探究了如何创建、访问和操纵数组。它是一种用来组织复杂信息的简单而有效的方法。

第11章 “对象”介绍了对象的构成。为你展示了如何使用内建的 **Sound**、**Color** 和 **Date** 对象，并传授给你一种使用 **Movie Clip** 对象的新方法。

第12章 “自创对象”展示如何应用你已经掌握的知识来创建更为复杂的对象。如果说数组是一种保存复杂信息的简单方法，那么对象就是一种用来保存那些真正可以称得上是“复杂信息”的途径了。

第13章 “**Smart Clip**”罗列了创建和使用 **Smart Clip** (聪明剪辑) 的所有方法。从完成标准的聪明剪辑的方法做起，你可以创建 **Custom User Interfaces** (定制用户接口)。

第14章 “连接外部数据”展示给你 Flash 与外部应用程序对话的多种方法。包括读取文本文件、与服务器应用程序进行交互、交换 **XML** 结构的数据、由 **JavaScript** 调用和进行控制、与 **Authorware** 和 **Director** 这样的宿主应用程序进行交互。我没有时间再为你展示所有外部应用程序或工具的具体使用，但它们与 Flash 的联系我想几乎没有什么遗漏。

要说的实在太多了，我不可能叙述完关于 Flash 的所有的一切。如果我真试图那样做的话，本书将可能厚达现在的两倍甚至更多。但无论如何，我可以确信只要你领悟了我在这 14 章中所组织的内容，你将有能力面对出现的各种新情况。

第 1 章

Flash 基础

Flash 专家是一个奇特的群体。其中的一些有着传统的动画制作和图形技巧背景。另一些则具有编程的背景。还有一些非常年轻的没有任何专业背景的年轻人，他们刚从大学毕业甚至年龄更小。但你来自哪里并不重要，因为 Flash 平易近人且功能强大。

由于本书是中级 Flash 和高级编程间的桥梁，因而读者必须有相应的水平。中级 Flash 用户，是指那些熟悉绘图、过渡和声音效果并正准备着手编程的朋友；正准备将已有的编程技巧应用于 Flash 的富有经验的程序设计师。但无论是谁，我们都需要从同样的水平开始我们的旅程。

本章把以后将要论述到的一些关键性的东西连接起来。它不是一个对绘图、过渡和声效的摘要。相反，它是一个特别针对 Flash 脚本编写的基础知识的快速概述。如果这些知识对你来说很熟悉，那么很好。但如果不是的话，在你向本书的其余部分发动攻击前请确信这些知识对你来说都已经变得有意义。

本章将涵盖：

- 嵌套 Movie Clip（电影剪辑）的 Timeline 体系结构
- Stage（舞台）坐标系统
- 诸如不可见按钮和空电影剪辑这样的传统 Flash 技巧
- 脚本出现的位置

我更愿意将本章中的大部分视为是对过去知识的回顾，但不管怎么说，它们中的某些内容对你来说可能是新东西。现在你可以开始按照我为你列出的顺序接近这些概念了。举个例子，如果你还从未听说过不可见按钮，那么好吧，你只需要尝试着去使用它。

1.1 Timeline 的体系结构

对于理解 Flash 很重要的一个关键性的概念是嵌套的 Timeline。任何时候当你选择了一个形状，然后选择 Insert > Convert to Symbol (F8)，这个形状都将被放置到 Library（库）



中并将在舞台上保留该符号的一个实例。你也可以选择一个符号的实例，然后选择 **Insert > Convert to Symbol**，这将在一个新符号的内部放置被选择符号的一个实例。重要的事情是要记住按 **F8** 键可以将任何被选择的事物（即使你选择的是一个符号实例）作为一个新符号放置到库中。这些嵌套的符号已经蕴含了编程和动画结构。

动画和文件大小的含义

要明白嵌套符号是如何应用于动画的，考虑如何创建一个包含有运动车轮的汽车符号。首先，绘制一个车轮然后将其转换成一个 **Graphic Symbol**（图形符号）。要创建一个旋转的车轮符号，必须将这个车轮图形符号的一个实例转换成一个 **Movie Clip**（电影剪辑）。然后对这个电影剪辑的内容进行编辑，具体说就是要通过创建一个简单的 **Motion Tween**（运动过渡）使车轮符号旋转起来。由于只有符号实例能够在 **Motion Tween** 中使用，因此将车轮符号放置到“运动的车轮”符号中的额外步骤是必须的。最后，使用两个“运动的车轮”剪辑，并将它们放置到第三个剪辑“汽车”中，然后就可以移动汽车使其穿过主 **Timeline** 了。

当工作结束时，汽车是一个包含两个“运动的车轮”剪辑实例的符号，而其中每一个“运动的车轮”都包含一个普通的车轮实例。当建立这样的嵌套符号时，通常应该按照“由内而外”或“由个别到整体”的顺序创建。但不管你以怎样的途径来创建，必须一直跟踪在舞台上方的地址栏，如图 1.1 所示。地址栏以当前正在编辑的场景名开始，然后显示如图中那样的符号嵌套层级，地址栏指出了正在编辑的“Wheel（车轮）”符号是位于“Wheel in Motion（运动的车轮）”中的，而“Wheel in Motion”又是嵌套在“Car（汽车）”中的。当然，你完全可以创建出比具有旋转的车轮更复杂的嵌套符号，你所要做的只是时刻留意正在做的是什。

除了用于创建复杂的效果以外，通过重复利用图形组件，嵌套符号也能够减小电影的文件大小。但如果将这个优势利用得太过分的话，它也可能对你产生负面影响。举一个荒唐的例子吧，你可以将一个只有一个像素的符号不断地重用并嵌套来创建一条直线，然后再用四条这样的直线来完成一个正方形等等。但不管怎么说，通常情况下符号能够重用的优势的确具有重要意义。

除了节省文件体积外，小心谨慎地使用符号（特别是电影剪辑）经常能够提高生产力。例如：当对基本符号进行改变时，这些改变将在该符号的所有实例上得到体现。当你复制一个还没有被转换成符号的形状时，将遭遇到两个问题：文件大小增加明显和 **Flash** 不能做出相应的调整。这是因为你可能将图形放置到了一个非整数的位置（即含有小数的位置），而 **Flash** 经常需要在绘制像素时进行舍入。最终导致的结果将是具有同样大小的两个不同的形状，你可以明显地看到它们之间的差别（尽管不大）。相反，同一符号的多个实例将总是完全相同的（除非你对它们进行了不同的缩放）。



第 1 章

“当前剪辑”（包含脚本的那个剪辑）将被瞄准。如果你想要瞄准一个不同的剪辑，你必须指明要瞄准哪个剪辑。

想像瞄准的一个方法是考虑计算机中的文件夹（和子文件夹）结构。你的那些一个套一个的文件夹看起来就像一个套一个的符号。（我将使用这个类比贯穿整个描述，并且在这部分的最后我将把它和 Flash 联系上。）假如你正在浏览一个文件夹并且想要打开某个子文件夹中的某个文件，你只需要简单地“瞄准”你想要浏览的子文件夹。当你决定瞄准另一个文件或文件夹时，你有两种基本的方法来做这件事：相对方法和绝对方法。在图 1.2 中从文件夹“Flash 5”处做一个快速的检索，如果你想要进入文件夹“Help”，然后进入其中的子文件夹“UsingFlash”，相对瞄准可以被表示为进入文件夹“Help”，然后进入子文件夹“UsingFlash”。考虑一次只能浏览或进入一个文件夹的情况，在这种情况下，当你进入文件夹“Help”开始时，你正在假设当前的文件夹中有一个被称作“Help”的文件夹。这就是一种相对瞄准，因为它是相对于你在那里而言的。

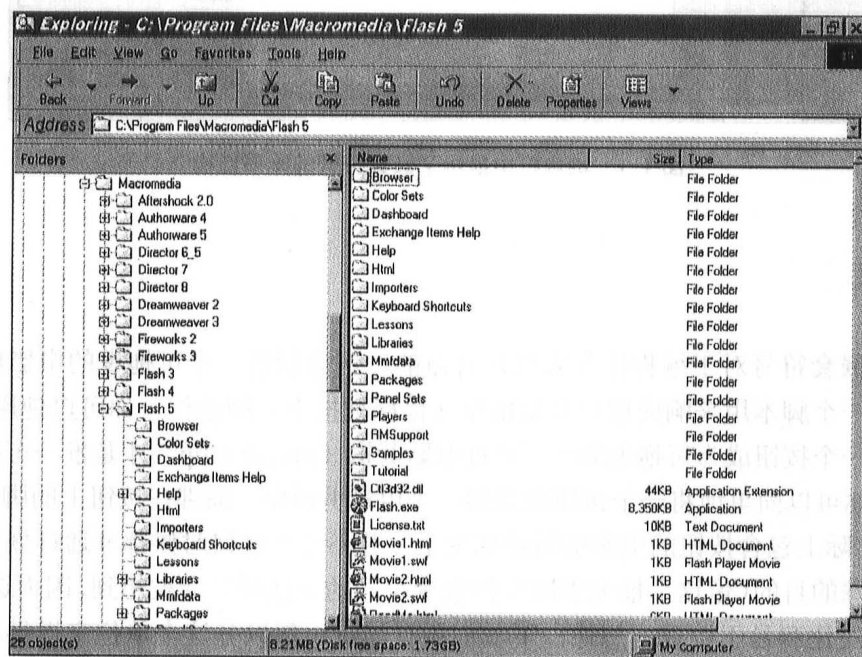


图 1.2 相对和绝对瞄准类似于在计算机上浏览文件和文件夹

除了这种“下潜”到子文件夹中的相对瞄准外，相对瞄准也可以上升。简言之，如果你正位于一个文件夹（或者电影剪辑）中，可以引用你所处的这个文件夹。例如：如果你正在浏览位于“Macromedia”文件夹中的“Flash 5”文件夹，你可以使用这样的相对瞄准来引用“Macromedia”文件夹：“上升一级”（可以在图 1.2 中点击“Up”）。在 Flash 中，一个相对引用将涉及术语 `_parent`，而在 HTML 中，字符 `../` 将被使用。你将在本章的后面学习这些，但重要的概念是要知道相对引用是可以“上”或“下”的。

绝对瞄准可以作为瞄准的另一选择。绝对瞄准指定了你正在瞄准的项目的完整地址。

这样，绝对引用将不受你当前正位于哪个文件夹的影响。在浏览文件夹的事例中，一个绝对路径将包括驱动器盘符。例如：`C:\Program Files\Macromedia\Flash 5` 就是一个绝对路径。你也可以将短语“下一户”或“下一条街”比作一个相对瞄准，而将“北京市三里河路 6 号”视为一个绝对瞄准。

使用哪一种引用（相对还是绝对）完全取决于你的主观判断。但通常情况下，相对引用更理想一些，因为你不会受制于层级体系的改变。想像一下当你将“Flash 5”文件夹移动到另一个硬盘时的情形，一瞬之间，绝对瞄准 `C:\Program Files\Macromedia\Flash 5` 就不能工作了。但如果你使用的是相对瞄准的话，例如 `Macromedia\Flash 5`（既从我所在的地方进入“Macromedia”文件夹，然后再进入“Flash 5”文件夹），它没有与任何特定的驱动器或文件夹捆绑在一起。相对引用的惟一不利因素是它们需要你处于某个特定的文件夹中（在这里就是“Program Files”文件夹）。绝对引用 `C:\Program Files\Macromedia\Flash 5` 不论你身处何处都能工作（只要这个文件夹没有被移动）。

尽管在相对和绝对瞄准中做出判断完全是一种主观行为，但你会发现当你不得不通过“上升”和“下潜”来发现目标时，相对瞄准将变得相当的复杂和纠缠不清。例如：如果你想要瞄准名为“Flash 3”的文件夹（它位于你的“Macromedia”文件夹中），而你此刻正位于“Flash 5”文件夹中，则这个相对瞄准将先要上升一个文件夹，然后再下潜到“Flash 3”中。使相对引用做方向改变不是不可能的，但在这种情况下，一个绝对瞄准将更加轻松。

现在，别让我们从 Flash 中飘出太远了，让我们将目光瞄准 Flash。瞄准剪辑实例（和它们的 Timeline）。使用先前的汽车例子，如果我正位于一个“车轮”中，我可以使用一个相对瞄准来瞄准包含我的汽车。命名每一个嵌套的电影剪辑将会很有帮助。例如：如果我要瞄准汽车的前轮，我需要确信汽车和汽车的轮子都有一个实例名。考虑在舞台上“汽车”符号有两个实例，由于一次只能瞄准一个实例——因此必须指定你正要瞄准的是哪一个汽车实例。

在 Flash 中的相对瞄准是很简单的。只需要使用你正要瞄准的剪辑实例名就可以了。如果这个实例名是“BigCar”，那么就使用它好了！要瞄准位于一个实例内部的实例，你只需简单地用一个点（.）来分隔每一个实例名。`BigCar.FrontWheel` 将瞄准“BigCar”实例中的“FrontWheel（前轮）”实例。如果你正在“FrontWheel”实例中编写要瞄准“BigCar”实例的脚本，你可以使用 `_parent`。术语 `_parent` 意指包含我所在的剪辑的剪辑（即我所在的剪辑的父剪辑）。你也可以使用 `_parent._parent` 等一次上升多级。你也可以瞄准处于“上一层”的某个实例。这么说吧，从“FrontWheel”实例中，你想要瞄准“BackWheel（后轮）”（该实例被包含在“BigCar”中——即包含“FrontWheel”的同一个剪辑），你可以使用 `_parent.BackWheel`。

Flash 中的绝对瞄准总是以 `_root` 开始的，它瞄准了主 Timeline。例如：`_root.BigCar.FrontWheel` 将瞄准主 Timeline 中实例“BigCar”中的名为“FrontWheel”的实例。绝对瞄准使用 `_root` 的惟一例外是当瞄准多级（levels）时。使用 Load Movie（装载电影），你可以在剪辑内部或某个级深度中播放 .swf 文件。要绝对地瞄准一个经由 Load Movie



第 1 章

装载的剪辑(或整个.swf),你需要以_level1(对于 level 2 使用_level2,对于 level 3 使用_level3 等等)来作为开始。

瞄准剪辑听起来像是一个没有多大意义的练习。但实际上的确有一些原因使你需要这样做。你可以瞄准一个剪辑来改变它的某些特性。例如:你可以瞄准汽车的前轮来改变它的 Alpha(透明)特性。脚本 `_root.BigCar.FrontWheel._alpha=50` 将把前轮的透明度设置为 50%。不仅仅只是改变属性,你还可以让动作来影响个别的实例。例如: `_root.BigCar.FrontWheel.gotoAndStop(10)` 将使电影进到第 10 帧(在 FrontWheel 实例中)并停止在那里。当以这种方法将一个动作应用到一个实例时,该动作被称为这个实例的一个方法(method)。我们将在第 4 章 Flash 的基本编程和第 7 章的 Movie Clip 对象中讨论关于改变特性和使用方法。在这里,你只需要明白瞄准剪辑为什么具有重要的意义就行了。

这一部分提供了有关瞄准的基础知识。但不管怎么说,你可能想要进行实践了。通过 Flash 的 Target Path Editor(目标路径编辑器)来学习这一知识是一个高明的方法。任何时候只要你能看到 Insert Target Path(插入目标路径)按钮(显示在图 1.3 中),就可以探测全部文件,而 Flash 将自动为你写出正确的目标引用。还可以分别使用相对和绝对瞄准来试验。

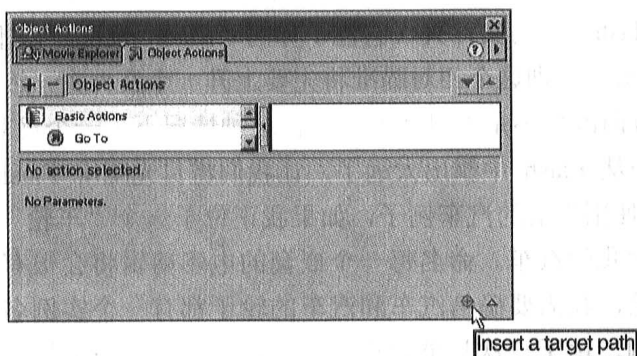


图 1.3 Insert Target Path 按钮帮助你编写目标路径

最后要说的一点是,尽管旧的“斜线”引用仍被支持(具有 HTML 编写经验的人会很熟悉这种东西),但我推荐你只使用点符号。你可以在图 1.4 中看到 Target Path Editor。

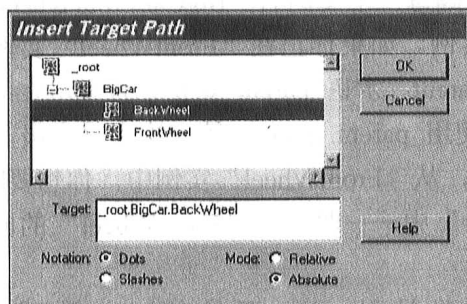


图 1.4 Target Path Editor 使你能够在使用相对还是绝对路径和使用斜线还是点符号之间做出选择