

The Essential Guide to Flex 3

Flex 3 程序设计

[美] Charles E. Brown 著
张骥 涂颖芳 等译

- ▶ Amazon畅销书
- ▶ Adobe技术专家力作，深入浅出
- ▶ 手把手教你使用Flex 3



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Web开发系列

The Essential Guide to Flex 3

Flex 3程序设计

人民邮电出版社
北京

图书在版编目(CIP)数据

Flex 3 程序设计 / (美) 布朗 (Brown, C. E.) 著; 张
骥等译. —北京: 人民邮电出版社, 2009.8
(图灵程序设计丛书)
书名原文: The Essential Guide to Flex 3
ISBN 978-7-115-21071-5

I. F… II. ①布…②张… III. 软件工具—程序设计
IV. TP311.56

中国版本图书馆CIP数据核字(2009)第100528号

内 容 提 要

本书通过简明易懂的示例向读者展示了如何使用 Flex 和 ActionScript 3.0 创建强大的富因特网应用程序。书中首先介绍了相关软件的安装及 Flex 和 ActionScript 的基本知识, 然后结合各种练习深入讲解了 Flex 的各种强大的功能: 容器、事件与组件、打印和图表功能等。本书最后还提供了专业的案例研究, 展示了如何构建完整的 Flex 应用程序。

本书是一本适合初、中级读者阅读的 Flex 教程。

-
- ◆ 著 [美] Charles E. Brown
译 张 骥 涂颖芳 等
责任编辑 傅志红
执行编辑 王军花
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 29.25
字数: 691千字 2009年8月第1版
印数: 1—3 000册 2009年8月北京第1次印刷
著作权合同登记号 图字: 01-2008-5176号
ISBN 978-7-115-21071-5/TP
-

定价: 65.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *The Essential Guide to Flex 3* by Charles E. Brown, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2008 by Charles E. Brown. Simplified Chinese-language edition copyright © 2009 Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

1954年Fortran语言的发明,使软件业跨入了高级语言时代;1972年Smalltalk的发布,标志着“面向对象”语言时代的到来;2004年Adobe公司推出的Flex框架,预示着富因特网应用程序(RIA)浓墨重彩地登上了历史舞台,从此网络应用程序的表示层只能基于单调的HTML页面的时代一去不复返了。

Flex从诞生到现在,已经历了5年时间,版本从1.0发展到了现在的3.3,功能也从仅支持J2EE (Java 2 Platform, Enterprise Edition)应用,到现在支持几乎所有的动态网页技术。可以说, Flex已经成为了开发富因特网应用程序的首选工具。Flex框架无缝集成了Flash、ActionScript以及MXML,并提供了丰富的可扩展用户界面及数据访问组件,使开发人员能够快速构建出具有丰富数据表现、强大客户端逻辑和集成多媒体的富因特网应用程序,从而极大地提高了Web界面的用户体验和人机交互性。

作为一本面向初中级读者的Flex教程,本书通过一些简单却典型的示例向读者展示了Flex的方方面面。作者在其多年教学经验的基础上,总结出一套独到的授课方式。一些晦涩的专业术语在作者的笔下变得通俗易懂,书中的示例在讲明要点的同时也尽量简单。此外,作者还为读者留下了大量思考和练习的空间。作者的这些精心安排一定会使读者在较短的时间内获得最佳的学习效果。希望读者能够借助此书成为RIA的开发先锋。

本书的大部分章节由张骥翻译,涂颖芳完成了部分章节的初译和全部的文档整理工作,熊炜、胡沙、史维、蒋宇轩、陈兴道、顾崇元、王谦、张颖、芦彤彤和周正歌对部分译文亦有贡献。在整个翻译和统稿过程中,译者尽可能地保证术语翻译的准确和统一,但错误和疏漏恐难避免,欢迎并感谢读者斧正。

译者

2009年5月

前 言

令人难以置信，我们现在有了第二代Flex。在刚刚完成本书第1版写作的几个星期内，我们就看到了Flex 3不断推出的各个beta版。在这期间，本书的很多章节都重写了三四遍。

首先我要感谢那些花时间在Amazon.com等地方写下热情评论的读者。我几乎阅读了每一条建议并把它们整合到了这一版。我削减了有关ActionScript的一些技术讲解，把重点放在了Flex本身的特性上。

我做过多年的技术培训工作，（期间只有少数几次涉及大型宽泛的主题），我学会了用更简短、更明确的概念解释来代替比较宽泛的技术化的（且常常是不易理解的）讲解方式。换句话说，我常常喜欢不兜圈子、一语中的。

在阅读本书的时候，请记住几件事情。首先，大家会发现我所展示的技术体现了我的编程和设计风格。当然，条条大路通罗马。但任何一本书都不可能涵盖所有编程风格，尤其是在讲解这种大型主题的时候。如果大家找到不同方法，只要管用，就尽可使用。

第二，为了阐明要点，我特意用些简单的示例。我不希望读者仅仅囿于那些食谱式的操作说明，它们所起的作用不过是测试一下大家阅读理解和操作的能力。虽然本书有一个案例研究贯穿始终，但每一章都是独立的，不会依赖于之前章节所做的练习。因此，大家可以翻至任何一章学习其内容。

第三，我会假定大家对面向对象编程概念至少已经有粗略的了解。虽然我会在各章随时阐释这些概念，但那只是非常基础的简单介绍。OOP是一个非常大的主题，相关论著已经汗牛充栋。

好，提示和声明就到此为止。

我希望本书起到的作用是：让大家对Flex和ActionScript 3.0环境有足够的体验，以便日后有能力解决自己所遇到的独特问题。我花了大量时间来讨论如何通过使用ActionScript 3.0 Language Reference来寻求帮助。

对于采用哪种服务器技术来展示Flex的动态一面，我必须做出决定。因为我在自己的工作中使用的是ColdFusion，所以我决定使用该技术。我那位了不起的技术编辑David Powers是PHP方面的权威，写过多本PHP图书。他热情地编写了一个在Flex中使用PHP的示例来作为ColdFusion的他选，为此我深表谢意。

希望大家读过本书后对Flex 3像我一样满怀激情。真心鼓励大家用本书中的示例多做尝试。要把这本书看成是开始，而不是结束。

让我们开始学习吧。

版式约定

为了让本书尽可能地清晰易读，书中使用了下列版式。

重要的词语或概念通常会在首次出现的时候用楷体突出显示。

代码用等宽字体显示。

新添代码或更改之后的代码通常用加粗的等宽字体显示。

菜单命令以“菜单→子菜单→子菜单”的形式写出。

在想要引起大家注意的地方，会采用如下样式突出显示。

嘿，别说我没警告过你。

有时候，代码无法排在书中的单独一行里。这时，我会使用箭头记号：➡。

This is a very, very long section of code that should be written all ➡
on the same line without a break.

致谢

没有众人的帮助，我是无法写成本书的。

每当我自以为写成了无懈可击的一章时，我那位了不起的技术编辑David Powers都会把我拉回到现实。他的聪明才智和悉心指导让本书的写作方向与第1版略有不同。我还要感谢他写的关于PHP技术的章节。

我必须感谢我的项目经理Sofia Marchant，她为频频更改的生产进度费了不少心。beta测试环境下的工作并不容易做，她却善于让一切进展顺利。

我要感谢我的所有朋友和同事（包括我培训班上的一些学生），他们为本书提供了宝贵的建议和看法。

最后，我要感谢很多热情的读者，他们在Amazon.com和其他一些地方（包括给我发电子邮件）写下了许多鼓励我的话。他们的很多留言为本书提供了一些不错的想法。

目 录

第 1 章 Flex 基础知识..... 1	
1.1 因特网今昔..... 1	
1.2 Flex与RIA..... 5	
1.3 Flex、Flex Builder和ActionScript 3.0..... 6	
1.4 安装Flex Builder 3..... 7	
1.4.1 将Flex Builder作为Eclipse插件 安装..... 8	
1.4.2 安装Flex Builder 3..... 10	
1.5 小结..... 17	
第 2 章 Flex 与 Flex Builder 3..... 18	
2.1 从Flex Builder 3出发..... 19	
2.1.1 创建一个Flex项目..... 21	
2.1.2 创建一个Flex应用程序..... 26	
2.1.3 更改属性..... 28	
2.2 剖析Flex应用程序..... 30	
2.2.1 MXML..... 31	
2.2.2 寻求帮助..... 34	
2.3 走进幕后..... 37	
2.3.1 部署文件..... 38	
2.3.2 查看生成的ActionScript代码..... 39	
2.4 小结..... 42	
第 3 章 ActionScript..... 43	
3.1 ActionScript编程概念..... 44	
3.1.1 理解什么是类文件..... 44	
3.1.2 和以前版本的兼容性..... 45	
3.1.3 开始使用ActionScript 3.0..... 46	
3.1.4 MXML与ActionScript..... 48	
3.2 混合MXML和ActionScript代码..... 50	
3.2.1 注释..... 55	
3.2.2 使用trace()函数..... 55	
3.2.3 函数..... 58	
3.2.4 传递参数..... 61	
3.2.5 处理事件..... 63	
3.2.6 使用[Bindable]标签..... 66	
3.2.7 添加交互性..... 66	
3.3 访问修饰符..... 68	
3.4 重构..... 69	
3.5 Flex调试..... 74	
3.6 小结..... 76	
第 4 章 容器..... 77	
4.1 应用程序容器..... 77	
4.2 布局管理器..... 78	
4.3 布局容器..... 80	
4.3.1 HBox和VBox容器..... 81	
4.3.2 Form容器..... 86	
4.3.3 Panel容器..... 90	
4.3.4 ControlBar容器..... 93	
4.4 导航容器..... 94	
4.4.1 ViewStack容器..... 95	
4.4.2 使用ActionScript实现导航 功能..... 103	
4.4.3 TabNavigator和Accordion容器..... 104	
4.5 状态..... 106	
4.5.1 更改状态..... 109	
4.5.2 状态与代码..... 114	
4.5.3 状态与鼠标悬停效果..... 115	
4.5.4 向项目中导入资产..... 116	
4.6 过渡..... 124	
4.6.1 构建容器..... 124	
4.6.2 构建状态..... 126	

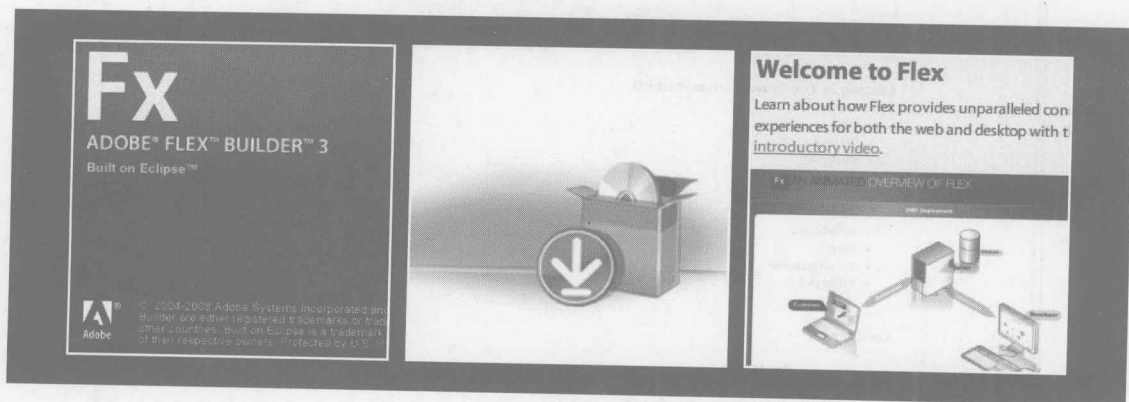
4.6.3 创建过渡	128	7.1.3 Flex 3与CSS	220
4.7 小结	131	7.2 验证与格式化	226
第5章 事件与组件	133	7.2.1 NumberValidator	227
5.1 事件	133	7.2.2 ActionScript与验证	230
5.1.1 事件对象	136	7.2.3 StringValidator	235
5.1.2 addEventListener	139	7.3 小结	237
5.2 组件	141	第8章 Repeater 组件	238
5.2.1 添加组件	145	8.1 理解Repeater组件	239
5.2.2 组件与数据	147	8.2 使用XML数据	244
5.3 自定义事件	151	8.3 小结	252
5.3.1 声明自定义事件	151	第9章 拖放操作	253
5.3.2 创建事件	153	9.1 理解拖放操作的概念	254
5.3.3 发送事件	153	9.2 拖至DataGrid组件	255
5.3.4 传递数据	156	9.3 拖至List控件	258
5.4 小结	159	9.4 用程序化的方式添加拖放功能	262
第6章 Flex 与 XML	160	9.5 小结	267
6.1 数据源	160	第10章 案例研究 I	268
6.2 XML简介	161	10.1 案例研究: 出版社的网站	269
6.3 在Flex中使用XML	163	10.1.1 创建项目环境	269
6.3.1 使用HTTPService标签	165	10.1.2 建立初始布局	271
6.3.2 显示数据	167	10.1.3 开始搭建结构	274
6.4 XML与ActionScript 3.0	168	10.1.4 创建组件	275
6.5 代码出错的时候	171	10.1.5 组装组件	291
6.6 Flash Player的安全性	174	10.1.6 在项目中使用时CSS	301
6.7 E4X入门简介	175	10.2 小结	302
6.8 Model标签	179	第11章 案例研究 II	303
6.9 DataGrid控件	180	11.1 连接数据源	303
6.10 修改DataGrid列	181	11.2 测试代码	309
6.11 DateFormatter类	183	11.3 处理图书封面图片	313
6.12 编辑和显示数据	185	11.4 CoverDetails组件	315
6.13 DataGrid容器中的图像	193	11.5 改变状态	320
6.14 Tree控件	197	11.6 给购物车添加功能	327
6.15 AdvancedDataGrid组件	207	11.7 小结	343
6.16 小结	210	第12章 Flex 与数据	344
第7章 格式化和 CSS	211	12.1 ColdFusion服务器	345
7.1 Flex与CSS	211	12.1.1 安装数据源	345
7.1.1 CSS基础知识	212	12.1.2 连接Flex与ColdFusion 8	349
7.1.2 在Flex中使用CSS	213		

12.1.3 使用数据	352	第 14 章 图表功能	395
12.1.4 换一种做法	367	14.1 使用PieChart组件	395
12.2 LCDS	369	14.2 使用ColumnChart组件	402
12.3 连接Flex与LCDS	374	14.3 制作图表动画	407
12.4 Flex与PHP	377	14.4 使用BarChart组件	409
12.5 使用PHP或ASP.NET作为应用程序 服务器	377	14.5 小结	410
12.5.1 准备数据库	377	第 15 章 AIR	411
12.5.2 构建应用程序	377	15.1 理解AIR	411
12.6 小结	382	15.2 Hello World	413
第 13 章 打印	383	15.3 Application Descriptor	420
13.1 打印过程	383	15.4 部署AIR应用程序	421
13.2 创建单独的Print容器	386	15.5 转换已有的Flex应用程序	427
13.3 打印与组件	391	15.6 结语	431
13.4 小结	394	附录 A 安装 ColdFusion 8	432
		索引	446

第1章

Flex基础知识

1



先做两个假设：

- 你所了解的因特网设计仅限于HTML页面。
- 你对Flex是什么尚一无所知。

基于这两点假设，我们就可以从头开始学起了。在这一章，我们要看看Flex在因特网发展过程中所处的位置。接着，将研究Flex到底是什么，它与传统的Web技术有何不同。

最后，在卷起袖子干活儿之前，需要先安装Flex及其相关技术，本章将带着大家完成这一过程。

1.1 因特网今昔

我们需要大致了解因特网到现在为止的历史，才能认识到Flex的优点。之所以要讲历史，是因为我们今天看到的各种各样的技术是在因特网发展的不同时间段上出现的。如前所述，为了弄明白Flex的价值所在，了解这个发展过程是非常重要的。

HTML 和动态网页

最早的网站只是传送文本数据，常常会包含到其他网页的超链接。因为因特网的连接速度非常慢（大家还记得28KB的连接速度吗），所以图片的数量会被控制到最小。目前仍然能够找到一

些这样的例子，如本书出版商的联系页面：www.friendsofed.com/contact.html。

图1-1显示了这个联系页面。

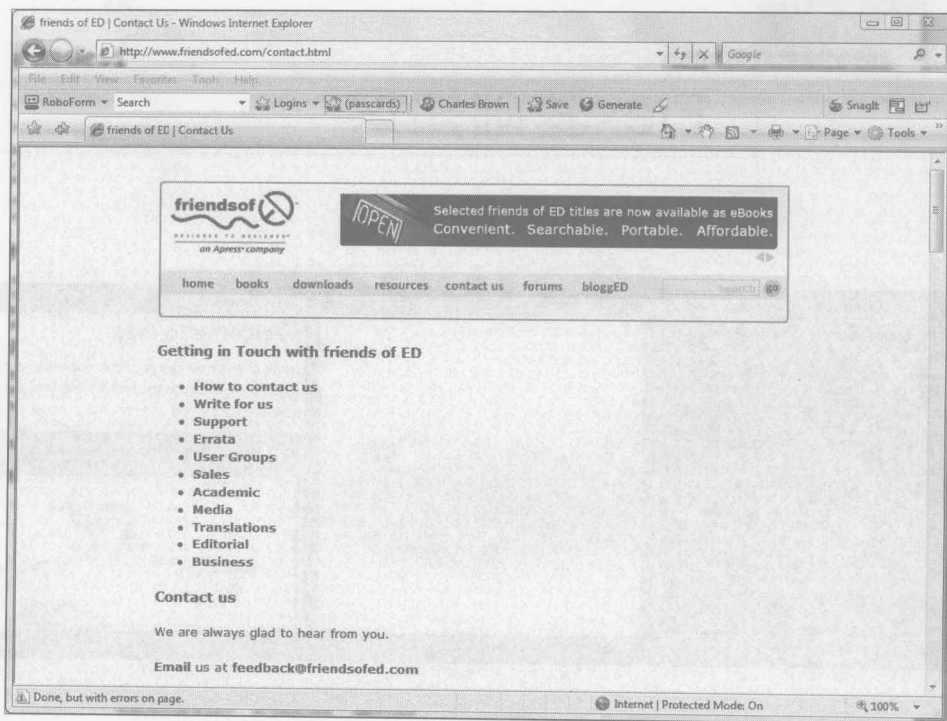


图1-1 friends of ED出版社网站的联系页面

这就是传统的HTML（Hypertext Markup Language，超文本标记语言）网站。注意除了文本和超链接外，网页上只有几幅简单的图片（早期网站上的图片甚至要更少）。还要注意文件扩展名.html，这种网页永远不会改变，除非有人真正进去改变它。

HTML页面（如图1-1所示的页面）就是指静态的或不会变化的页面。不会变化也许是个不恰当的说法。更确切地说，只有进入XHTML代码并手动作出修改的时候它才会改变。

2

下面讲讲静态页面是如何调用的。

在浏览器中键入www.friendsofed.com后，该请求会通过万维网上的一系列路由器传送出去，最终到达主Web服务器。Web服务器会搜索被请求的HTML页面所在的根目录，找到并将该页面打包，在包上面注明返回地址，然后把它发送回浏览器。接着，浏览器会读取HTML代码，显示出你所看到的页面。“在因特网上”查看Web页面——这是一个普遍的误解，我在培训班上仍能听到这种说法。Web页面是下载到你的计算机上，并通过计算机看到的。一旦Web服务器把HTML页面发送给你，它的工作就完成了。更确切地说，你是在客户端机器上查看页面的。作为Web页面的使用者或查看者，你就是客户。

当然，这其实是一种简单的解释。本书不准备详细讨论怎样构建和分发HTML页面。讨论这

些细节的书有很多。我推荐由Craig Grannell编著的*The Essential Guide to CSS and HTML Web Design* (friends of ED, 2007)。

让我们看看因特网的下一步演变。

请前往下列网站: www.adobe.com/cfusion/webforums/forum/index.cfm?forumid=60。

这个Web地址会把我们带到Adobe Flex Support Forums页面, 如图1-2所示。

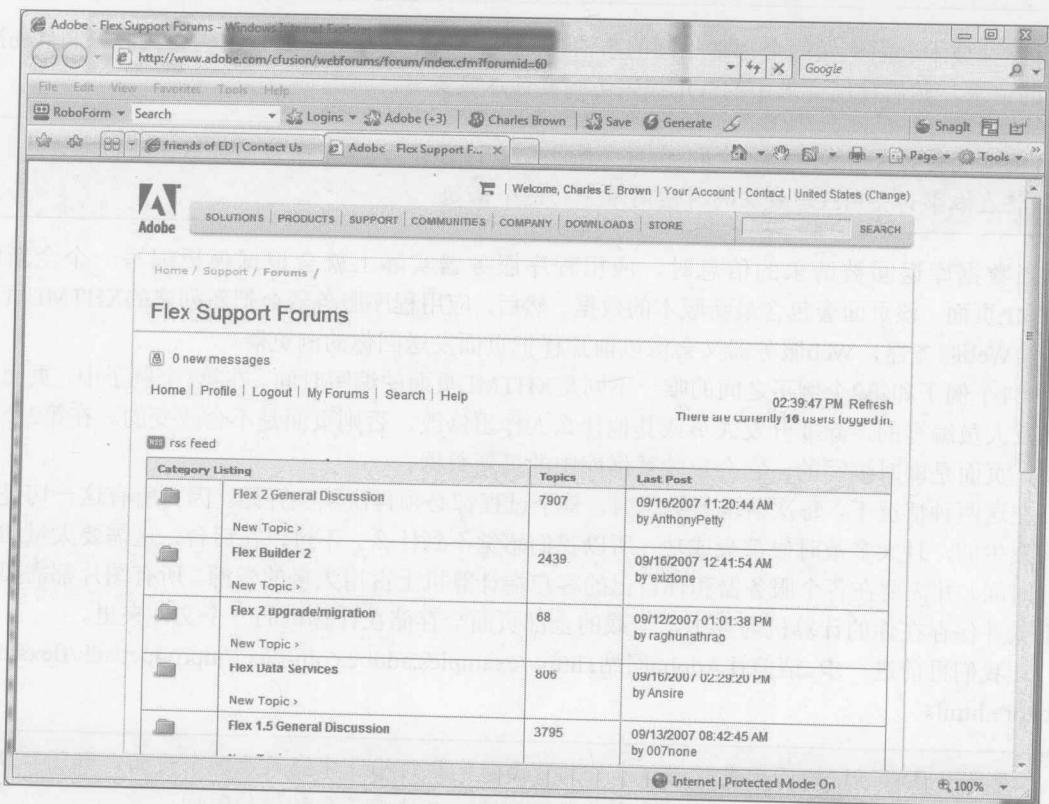


图1-2 Adobe Flex Support Forums 页面

这是一个传统的动态网站的例子。我们来看看其运行机制。它们向前面的静态页面又添加了两个步骤。

在浏览器中键入URL (Web地址) 之后, 请求同样会马上通过万维网上的一系列路由器传出去, 最后找到Adobe的Web服务器。事情就是在这里有了一些变化。

注意地址中的字母cfm。这些字母会让Web服务器把请求发送到另一个名为应用程序服务器的软件。处理动态技术的应用程序服务器有5种 (实际上, 还有其他几种, 但这5种是最流行的):

- ❑ CFM: ColdFusion;
- ❑ ASP: Classic Microsoft Active Server Pages;
- ❑ ASPX: Microsoft .NET Active Server Pages;

4

□ JSP: Java Server Pages;

□ PHP: 一种脚本语言, 其字母并不代表什么。

我的技术编辑David Powers对“PHP并不代表什么”这句话提出了异议。虽然有些网站称PHP为“PHP: Hypertext Preprocessor”的递归缩写, 但这仍然没有说清楚PHP到底代表什么。

这5种技术表面上做的全都是相同的事情, 只是难易程度各不相同。它们会接收来自Web服务器的请求, 然后使用请求中的SQL代码访问数据库服务器。

SQL代表Structured Query Language (结构化查询语言), 它是向访问数据库的标准方法。我们将在本书讨论Flex与数据的时候简单介绍这个话题。

当数据库返回被请求的信息时, 应用程序服务器实际上就会根据模板编写一个全新的XHTML页面。该页面会包含最新版本的数据。然后, 应用程序服务器会把新创建的XHTML页面返回给Web服务器, Web服务器又会像以前那样把页面发送回你的浏览器。

第1个例子和第2个例子之间的唯一不同是XHTML页面的编写时间。在第1个例子中, 页面是由开发人员编写的, 除非开发人员或其他什么人作出修改, 否则页面是不会改变的。在第2个例子中, 页面是即时编写的, 它会反映数据库中的最新数据。

在这两种情况下, 每次请求新数据时, 整个过程都必须再次从头开始。因为所有这一切是在瞬间发生的, 且大多数时候都会成功, 所以我们感觉不到什么。不过, 在后台, 这需要大量的服务器时间, 并需要在各个服务器和你自己的客户端计算机上占用大量的资源。所有图片都需要分别下载并保存在你的计算机内存中, 下载的全部页面会存储在计算机的一个文件夹里。

让我们再前进一步。请前往Adobe网站: <http://examples.adobe.com/flex2/inproduct/sdk/flexstore/flexstore.html>。

需要用Flash Player 9或更新的版本作为Web浏览器的插件才能显示这个页面, 如图1-3所示。如果没有这个插件, 系统就会提示我们下载它。这只需花数秒钟的时间。

5

看看这个网站, 它与前两个页面的差别非常明显。注意在单击选项卡时, 我们会从一个页面平滑地移动到另一个页面, 且没有在前面的例子中看到的重载过程。另外, 在Products选项卡中, 如果更改手机的价格范围, 就会看到手机自动重新排列的动画。

这是Flex网站的原型, 其内部机制当然就是本书的主题。不过, 就其最简单的形式来说, 我们真正加载的只有一个文件, 即Flash SWF (读为“swif”)文件。这之后, 当信息需要改变时, 我们刷新的是所改变的内容而不是整个页面。这就意味着潜在错误更少、数据显示更快, 用户体验更完美。另外, 大家还会在书中看到它所占据的资源也会更少, 因而成为今天新兴的便携式因特网设备的理想之选。

这种技术到底哪里与众不同的呢?

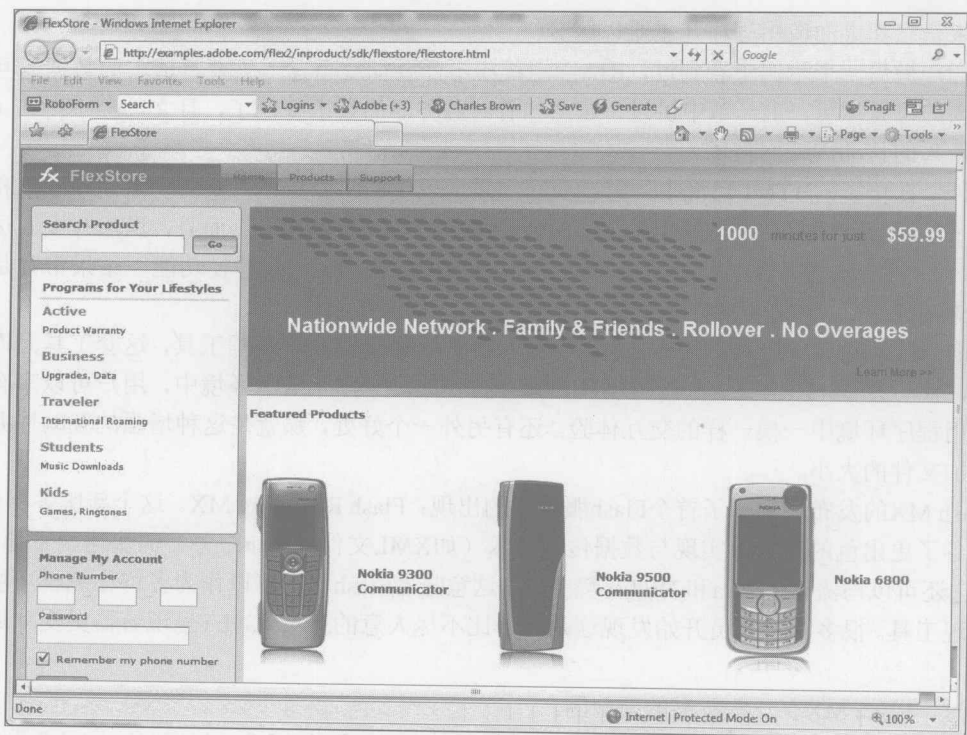


图1-3 Flex网站的原型

1.2 Flex 与 RIA

为了能够彻底理解最后一个示例是怎么回事，你可能需要略微改变一下自己的思路。

大家在前两个例子中看到过，传统的Web页面是通过向服务器发送请求并经历刚才所讲的过程才从一个页面转到另一个页面的。对于动态页面，Web服务器会把收到的请求发送给前面所讲的5个应用程序服务器之一，后者又会把请求发送给数据库服务器。然后，应用程序服务器会把数据汇集起来，并写好新的HTML页面发送给Web服务器，最后再发送给你的Web浏览器显示。如果你去了某个网站（比如Amazon.com）的5个不同的页面，整个过程就会经历5遍。我想大多数人都会同意，这并不是一个很高效的做事情的方式。

而且，我认为大多数人都可以轻松地区分出因特网应用程序（如前两个示例）和桌面应用程序（如Microsoft Word）之间的差别。整体的界面外观是不同的。

如果整个过程运行得更高效一些，岂不妙哉？如果桌面应用程序和Web应用程序的观感或多或少地相同一些，不是更好吗？

最后一个示例中的Flex原型像一个因特网应用程序吗？还是更像一个桌面应用程序？

为了解决这些问题，Macromedia（现为Adobe）借Flash MX的推出引入了一个新的术语：RIA（Rich Internet Application，富因特网应用程序）。这项基于Flash的技术冲破了传统HTML的很多局

限，结果是它和桌面应用程序几乎难以区分。

我们在最后一个示例中看到过，RIA应用程序不需要彻底重建。只要将请求的数据返回并插入到需要的位置即可。上一节中讲过，这意味着对服务器的要求降低了，且文件会小得多（这非常适合新兴的移动因特网技术）。

另外，在传统的HTML环境中，用户交互只限于表单和几个按钮。桌面功能，如菜单和不同版面的平滑过渡，则常常很不好用，且它们会显著地增加文件的大小。而且，虽然开发人员会使用JavaScript实现这些功能，但浏览器安全程序常常会阻止JavaScript行使功能，结果常常是丢失了更多的功能。

Flash MX通过为Web开发人员提供了一整套解决这些问题的新编程工具，这套工具允许更好的交互，且不会有HTML/JavaScript被阻止的问题。忽然之间，在RIA环境中，用户可以享有和在桌面应用程序环境中一模一样的交互体验。还有另外一个好处，那就是这种增强的互动性并不会显著增加文件的大小。

Flash MX的发布还预示了首个Flash服务器的出现：Flash Remoting MX。这个新服务器使RIA环境具备了更出色的能力来实现与数据传输技术（如XML文件和Web服务）的快速、平稳交互。此外，它还可以与流行的Java和.NET环境交互。这意味着Flash现在可以作为多样化编程环境下的一个呈现工具。很多开发人员开始发现这是一个比不尽人意的Java和.NET呈现容器更受欢迎的选择。

不过，Flash MX有一些特有的新问题。

Flash MX发布之后，Macromedia推出了升级版的ActionScript 2.0。ActionScript 1.0是一门相当简单的过程式语言，用来辅助创建Flash动画。为了解决RIA的新需要，ActionScript 2.0变成了一门半面向对象编程（OOP）的语言。

如果你是Flex或编程环境的初学者，可能就会不熟悉OOP、ActionScript或过程式语言这些术语。不熟悉也不要紧，这里只是对因特网历史的讨论。我会在本书的后面仔细地说明这些术语。

虽然它遵循了OOP语法的一些原则，但它也必须支持以前的非OOP ActionScript 1.0。事情并非总是尽如人意，很多人抱怨没有调试工具。

很多开发者还抱怨，为了开发RIA，他们需要了解Flash环境的很多复杂元素（时间轴、场景等）。

为了解决这么多的问题，Macromedia于2004年推出了Flex。Flex没有Flash那么多的复杂设计元素，而是为开发人员提供了一个更为传统的编程环境。它甚至有自己的类似Dreamweaver的开发工具，名为Flex Builder。不过，由于ActionScript 2.0的局限性，没人指望它能够流行。

它显然需要一次大的革新。

1.3 Flex、Flex Builder 和 ActionScript 3.0

Flex 2是在2006年的夏天推出的。它不只是对原来的Flex的升级，而是一次彻头彻尾的革新。

首要的改变是ActionScript 3.0的推出。

本书后面将会介绍到, ActionScript 3.0是一门类似于C++和Java的成熟的开源编程语言。虽然你可能还会把ActionScript与Flash联系起来, 但这两者现在只是偶尔发生关系。换句话说, 如果愿意, 只用ActionScript而不用Flash就可以构建一个完整的应用程序。

如果要求你用两个词汇来描述Flex是什么, 你可以不费力地说它是一个呈现服务器(presentation server), 第2章会详细分析这个概念。不过, 你现在只需要知道就其最简单的形式来说, 它胜过前面所讲的任意一种应用程序服务器, 并在呈现数据方面取代了XHTML/JavaScript。因此, 它不是把数据作为XHTML来呈现, 而是可以使用Flash (SWF) 文件的动态能力来呈现。

为了适应这套新的强大的开发工具, Adobe决定不升级类似Dreamweaver的Flex Builder 1。取而代之的是, Adobe选择了很多程序员熟悉的开发环境: Eclipse。

Eclipse 与 Flex Builder 3

Eclipse是一个有众多程序员, 特别是Java程序员使用的免费编程开发环境, 我们称之为IDE或Integrated Development Environment (集成开发环境)。它允许开发人员同时在多个编程环境中工作。大家可以在如下地址找到Eclipse: www.eclipse.org。

8

虽然Eclipse被Java开发人员广泛使用, 但其真正的强大之处是能够容纳各种编程语言的插件。例如, C++和PHP都有插件, 甚至连进行ColdFusion开发的免费插件也日益流行起来。我们将在本书后面集成Flex与ColdFusion时使用这个插件。

虽然Eclipse的很多插件是免费的, 但Flex Builder 2并不免费。不过, Flex Builder 2允许开发人员在带有很多强大的编程和调试工具的传统IDE中工作。

Flex Builder 3具有更多的强大工具, 大家将会在本书后面看到它们, 这些工具允许开发人员驾驭Adobe公司其他的设计和开发工具的强大功能。部分改进之处包括:

- 改进之后的Design视图可以利用强大的Adobe CS3设计工具, 这可以改进设计师与开发人员之间的工作流程;
- 新改进的数据组件让数据源与服务器的连接方式变得更加简单;
- 可以在Flex Builder IDE中构建和部署新的AIR (Adobe Integrated Runtime, Adobe集成运行时) 工具。

在写作本书时, Adobe公司宣布其正在让Flex完全开源。这意味着其他开发人员可以为Flex开发具有竞争力的IDE。不过对于本书来说, 我们将使用Flex Builder 3。

第2章将详细研究Flex Builder 3。不过现在, 我们必须先安装这个工具才能使用它。做好了安装准备之后, 就让我们继续吧。

1.4 安装 Flex Builder 3

Flex Builder由3个单独的组件组成。