

在线服务：视频库、源代码库、专业论坛、专家实时支持

Flex⁺ ASP.NET Web 应用开发 实战详解

许勇 王黎 等编著



55段全程配音语音教学视频
全书实例源代码，使学习、分析、调试程序更方便

在线服务方式

在线服务网站：www.itzcn.com

QQ群在线服务：45368980、33925615、107423140



清华大学出版社

在线服务：视频库、源代码库、专业论坛、专家实时支持

Flex⁺ ASP.NET Web 应用开发 实战详解

Flex⁺
ASP.NET Web

许勇 王黎 等编著

清华大学出版社

北京

内 容 简 介

Flex 是开发富互联网应用程序 (Rich Internet Application, RIA) 的利器, 它提供了丰富的可扩展用户界面及数据访问组件。本书共分为 5 篇, 分别是 Flex 基础知识篇、ASP.NET 编程篇、Flex 组件应用篇、Flex 数据交互篇和综合实例篇。全书对 Flex 和 ASP.NET 两种流行技术进行了归纳和总结, 内容覆盖了 Flex 和 ASP.NET 技术的知识和应用场景, 力求通过实例使读者更形象地理解 ActionScript 的编程思想, 快速掌握 Flex 的组件开发。

本书可作为 Flex 开发人员的重要学习资料, 也可作为网站开发和 Flex 开发人员的职业培训教程。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目 (CIP) 数据

Flex+ASP.NET Web 应用开发实战详解 / 许勇, 王黎等编著. —北京: 清华大学出版社, 2010.10

ISBN 978-7-302-22316-0

I. ①F… II. ①许… ②王… III. ①软件工具—程序设计②主页制作—程序设计
IV. ①TP311.56②TP393.092

中国版本图书馆 CIP 数据核字 (2010) 第 055946 号

责任编辑: 夏兆彦

责任校对: 徐俊伟

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京密云胶印厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 190×260 印 张: 34.75 字 数: 862 千字

附光盘 1 张

版 次: 2010 年 10 月第 1 版

印 次: 2010 年 10 月第 1 次印刷

印 数: 1~5000

定 价: 65.00 元

产品编号: 031595-01

FOREWORD

前言

Adobe Flex 是为满足希望开发富互联网应用程序的企业级程序员的需求而推出的表示服务器和应用程序框架。Flex 是开发富互联网应用程序 (Rich Internet Application, RIA) 的利器, 它无缝整合了 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演示、强大客户端逻辑和集成多媒体的 RIA 应用程序。目前已广泛应用于各种商业领域, 如电子商务、行政管理、企业业务流程自动化等方面。

1. 本书内容

本书对 Flex 和 ASP.NET 两种流行技术进行了归纳和总结, 内容覆盖了 Flex 和 ASP.NET 技术的知识和应用场景。本书共分为 5 篇, 分别是: Flex 基础知识篇、ASP.NET 编程篇、Flex 组件应用篇、Flex 数据交互篇和综合实例篇。各篇主要内容如下所示。

第 1 篇: Flex 基础知识篇 (第 1~5 章)。主要介绍 Flex 中脚本语言 ActionScript 的知识, 像 ActionScript 的运算符、控制语句、面向对象编程和内置类等。其中, 第 1、2 章向读者介绍 Flex 环境的配置以及 Flex Builder 3.0 的基本操作。第 3、4 章主要介绍 ActionScript 的语法和面向对象的支持, 像常量、变量、数据类型、运算符、对象、类、属性和方法等。第 5 章介绍如何使用函数、处理字符串、处理数组以及处理日期和时间。

第 2 篇: ASP.NET 编程篇 (第 6~8 章)。主要针对 ASP.NET 初学者介绍开发常规网站所必备的基础知识, 包括 ASP.NET 提供的环境的配置, C# 3.5 语法, ADO.NET 提供的对数据的查询、增加、删除、修改操作以及数据显示控件, XML 数据的解析、显示和生成, ASP.NET 的内置对象、Web 服务和文件操作。

第 3 篇: Flex 组件应用篇 (第 9~13 章)。第 9、10 章详细介绍 Flex 中的基础类组件、布局类组件、导航和容器类组件。第 11 章主要介绍如何使用行为、对象状态和动画效果制作出绚丽的界面效果。第 12 章介绍 Flex 的事件机制, 重点是使读者理解事件的工作流程、了解 Event 对象并掌握如何使用自定义事件。第 13 章介绍如何创建用户自定义组件, 为组件定义 CSS 样式和主题, 以及在组件之间进行参数传递。

第 4 篇: Flex 数据交互篇 (第 14~15 章)。主要针对数据存储和交互这两个方面进行介绍, 其中第 14 章向读者介绍在 Flex 3.0 中如何对数据进行处理, 包括数据模型、数据绑定、DataGrid 组件、数据验证和数据格式化等内容。第 15 章介绍在 Flex 中如何与服务器端进行数据交互, 包括常用的数据传输方式、使用 HTTPService 和 WebService 组件进行交互, 并制作了留言本和域名查询实例。

第5篇：综合实例篇（第16~17章）。这一篇包括两个实例，第16章使用 Flex 制作一个功能齐全的 FLV 播放器，实现色彩调整、视频列表、播放控制等功能。第17章则是一个视频展示网站，在服务器端使用 ASP.NET 和 SQL Server，前台实现了用户注册和登录，视频分类、搜索、收藏、播放及视频列表等功能；后台实现了类别、视频的添加和管理操作。

2. 本书特色

书中采用大量的实例进行讲解，力求通过实例使读者更形象地理解 ActionScript 的编程思想，快速掌握 Flex 的组件开发方法。本书难度适中，内容由浅入深，实用性强，覆盖面广，条理清晰。

- **知识点全** 本书紧紧围绕利用 Flex 与 ASP.NET 进行 RIA 程序开发展开讲解，具有很强的逻辑性和系统性。
- **实例丰富** 书中各实例均经过作者精心设计和挑选，它们都是由作者在实际开发中的经验总结而来，涵盖了在实际开发中所遇到的各种问题。
- **应用广泛** 对于精选案例，给出了详细步骤，结构清晰简明、分析深入浅出，而且有些程序能够直接在项目中使用，避免读者进行二次开发。
- **基于理论，注重实践** 在讲述过程，不仅仅介绍理论知识，而且在合适位置安排综合应用实例，或者小型应用程序，将理论应用到实践当中，来加强读者实际应用能力，巩固 Flex 开发基础知识。
- **随书光盘** 本书为实例配备了视频教学文件，读者可以通过视频文件更加直观地学习 Flex 和 ASP.NET 的使用方法。
- **网站技术支持** 读者在学习或者工作的过程中，如果遇到实际问题，可以直接登录 www.itzcn.com 与我们取得联系，作者会在第一时间内给予帮助。

3. 读者对象

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识及丰富的实例来指导读者透彻地学习 Flex 各方面的知识。本书可以作为 Flex 开发人员的重要学习资料，也可以作为网站开发和 Flex 开发人员的职业培训教程。

本书适合以下人员阅读学习。

- Flash 开发人员。
- Flex 应用开发人员。
- 网站建设及网络开发人员。
- Flex 开源项目爱好者。
- RIA 应用爱好者。

除了封面署名人员之外，参与本书编写的还有杨辉、胡家宏、于永军、张秋香、李乃文、张仕禹、夏小军、赵振江、李振山、李文才、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林、郭新志、牛丽平、唐守国等。在编写过程中难免会有疏漏之处，欢迎读者与我们联系，帮助我们改正提高。

CONTENTS

目 录

第 1 篇 Flex 基础知识篇

第 1 章 Flex 3.0 入门	2
1.1 Flex 概述	2
1.1.1 RIA 发展	2
1.1.2 Flex 简介	4
1.1.3 Flex 架构	5
1.2 Flex 3.0	7
1.3 MXML 概述	9
1.3.1 MXML 命名规范	10
1.3.2 MXML 文件结构	11
1.4 ActionScript 3.0 概述	13
1.4.1 ActionScript 3.0 简介	13
1.4.2 在 Flex 中 ActionScript 的使用方式	14
1.5 部署 Flex 3.0 开发环境	17
1.5.1 获取 Flex 3.0	17
1.5.2 安装 Flex Builder 3	17
1.5.3 第一个 Flex 程序	21
第 2 章 熟悉开发环境 Flex Builder 3	25
2.1 熟悉 Flex Builder 3 的工作区	25
2.1.1 Editors	25
2.1.2 其他窗格	28
2.2 编译与运行 Flex 3.0 程序	33
2.3 调试 Flex 3.0 程序	33
2.3.1 添加断点	33
2.3.2 调试程序	35
2.3.3 监视变量	36
2.4 Flex 3.0 项目概述	37
2.4.1 Flex Project	37
2.4.2 ActionScript Project	42
2.4.3 Flex Library Project	44
2.5 Flex Builder 3 中的常用快捷键	45
2.6 使用 Flex 帮助文档	46
第 3 章 ActionScript 3.0 语法	48
3.1 常量和变量	48
3.1.1 常量	48

3.1.2 变量	50
3.2 数据类型	51
3.2.1 基本数据类型	51
3.2.2 复合数据类型	54
3.2.3 数据类型检查	57
3.2.4 is 和 as 运算符	60
3.2.5 数据类型转换	61
3.3 运算符	66
3.3.1 运算符的分类	66
3.3.2 常用运算符	67
3.3.3 运算符的优先级	69
3.4 流程控制语句	70
3.4.1 条件语句	70
3.4.2 循环语句	72
第 4 章 ActionScript 3.0 面向对象	77
4.1 类和对象	77
4.1.1 面向对象概述	78
4.1.2 类的基本概念	80
4.1.3 类成员修饰符	81
4.1.4 定义方法	83
4.1.5 定义属性	89
4.2 包和命名空间	90
4.2.1 包	90
4.2.2 命名空间	94
4.3 枚举类	101
4.4 继承	103
4.4.1 继承概述	103
4.4.2 属性的继承	104
4.4.3 方法的继承和覆盖	107
4.5 接口	109
第 5 章 ActionScript 3.0 中常用	
数据处理	112
5.1 函数	112
5.1.1 定义函数	113
5.1.2 调用函数	114
5.1.3 函数的返回值	114
5.1.4 函数的作用域	114
5.1.5 函数的参数	116
5.2 字符串	121

5.2.1 创建字符串	121
5.2.2 String 类的属性和字符串中的字符	122
5.2.3 在字符串中查找子字符串和模式	124
5.2.4 替换子字符串和模式	127
5.2.5 字符串的连接与比较	129
5.3 数组	131
5.3.1 数组简介	131
5.3.2 索引数组	132
5.3.3 关联数组	140
5.3.4 多维数组	143
5.3.5 克隆数组	145
5.4 日期和时间	145
5.4.1 创建 Date 对象	146
5.4.2 获取时间单位值	147
5.4.3 执行日期和时间运算	147
5.4.4 控制时间间隔	148

第 2 篇 ASP.NET 编程篇

第 6 章 ASP.NET 的简单应用	152
6.1 ASP.NET 3.5 概述	152
6.1.1 .NET Framework 3.5 简介	152
6.1.2 开发环境简介	155
6.2 C# 3.5 语法概述	159
6.2.1 控制语句	159
6.2.2 面向对象实现	167
6.2.3 结构	172
6.2.4 枚举	173
6.2.5 数组和集合	175
6.3 配置应用程序	178
6.3.1 ASP.NET 配置概述	179
6.3.2 Web.config 结构	180
6.3.3 在 Flex 中生成 Web.config	181
第 7 章 ASP.NET 数据显示	185
7.1 ADO.NET 概述	185
7.1.1 ADO.NET 命名空间	186
7.1.2 ADO.NET 组件	187
7.1.3 ADO.NET 对象	187

7.2 数据显示控件	191
7.2.1 ListView 控件	191
7.2.2 DataList 控件	195
7.2.3 GridView 控件	199
7.2.4 Repeater 控件	202
7.3 XML 命名空间和控件	205
7.4 显示 XML	206
7.4.1 XML 控件读取	208
7.4.2 DOM 技术读取	208
7.4.3 DataSet 对象读取	209
7.4.4 XmlTextReader 类读取	210
7.5 生成 XML	211
7.5.1 使用 DataSet 创建	211
7.5.2 使用文本方式创建	212
第 8 章 ASP.NET 高级应用	214
8.1 ASP.NET 内置对象	214
8.1.1 Response 对象	214
8.1.2 Request 对象	215
8.1.3 Server 对象	217
8.1.4 Application 对象和 Session 对象	218
8.1.5 Cookie 对象	220
8.2 Web 服务	222
8.2.1 Web 服务概述	222
8.2.2 创建 Web 服务	225
8.2.3 使用 Web 服务	228
8.3 处理文件	230
8.3.1 System.IO 命名空间	231
8.3.2 操作驱动器	231
8.3.3 操作文件夹	233
8.3.4 操作文件	237
8.3.5 读写文件	238

第 3 篇 Flex 组件应用篇

第 9 章 使用组件	244
9.1 Flex 组件概述	244
9.2 Flex 常用组件	245
9.2.1 文本组件	245
9.2.2 CheckBox 和 RadioButton 组件	249

9.2.3 ComboBox 和 List 组件	253
9.2.4 按钮组件	256
9.2.5 Image 组件	258
9.2.6 日期组件	259
9.3 导航类组件	263
9.3.1 ToggleButtonBar 和 TabBar 组件	263
9.3.2 MenuBar 组件	266
9.3.3 PopUpButton 和 PopUpMenu Button 组件	268
第 10 章 使用容器布局页面	272
10.1 管理程序的布局	272
10.1.1 控制 Application 组件的布局	272
10.1.2 ApplicationControlBar 组件	274
10.1.3 HBox、VBox 和 Canvas 组件	276
10.1.4 HDividedBox 和 VDividedBox 组件	279
10.2 窗口布局	282
10.2.1 Panel 组件	282
10.2.2 TitleWindow 组件	285
10.3 表单布局	287
10.4 动态控制对象的布局	288
10.4.1 Tile 组件	289
10.4.2 Grid 组件	290
10.5 导航容器	293
10.5.1 ViewStack 组件	293
10.5.2 Accordion 组件	294
10.5.3 TabNavigator 组件	296
第 11 章 使用行为对象和动画效果	299
11.1 认识行为对象	299
11.1.1 行为对象简介	299
11.1.2 创建行为对象	300
11.2 行为和组件	301
11.2.1 组件的行为和动画效果	301
11.2.2 为组件添加行为——执行监听动画	302
11.3 常见动画效果	304
11.3.1 模糊效果	304
11.3.2 淡入淡出效果	306

11.3.3	发光效果	307
11.3.4	彩虹效果	309
11.3.5	溶解效果	310
11.3.6	移动效果	312
11.3.7	尺寸调整效果	313
11.3.8	旋转效果	315
11.3.9	声音效果	316
11.3.10	缩放效果	318
11.3.11	擦除效果	319
11.3.12	复合效果	321
11.4	行为和状态	323
11.4.1	使用 State 对象	323
11.4.2	使用 Transition 对象	327
第 12 章	事件机制	331
12.1	观察者模式	331
12.2	ActionScript 3.0 的可视化对象架构	335
12.3	事件机制的工作流程	337
12.3.1	事件流	337
12.3.2	Event 对象概述	341
12.3.3	创建自定义事件	342
12.3.4	扩展自定义事件	348
12.4	事件机制的高级应用	351
第 13 章	自定义组件	357
13.1	创建组件	357
13.1.1	使用 MXML 创建组件	358
13.1.2	使用 ActionScript 创建组件	359
13.2	在组件文件中添加项目	361
13.2.1	在 MXML 文件中添加项目	361
13.2.2	在 ActionScript 文件中 添加项目	365
13.3	使用 CSS 样式	367
13.3.1	CSS 样式语法	368
13.3.2	创建 CSS 文件	370
13.3.3	引用 CSS 样式	371
13.3.4	使用主题	377
13.4	参数传递	381
13.4.1	属性的传递	381
13.4.2	方法的传递	383
13.4.3	事件的传递	384

第 4 篇 Flex 数据交互篇

第 14 章	Flex 中的数据处理	390
14.1	数据模型	390
14.1.1	使用<mx:Model>组件	390
14.1.2	使用<mx:XML>组件	393
14.1.3	使用<mx:Object>组件	394
14.1.4	使用 ActionScript 脚本	395
14.1.5	使用类	396
14.2	数据绑定	400
14.2.1	简单绑定方式	400
14.2.2	使用<mx:Binding>组件	403
14.2.3	使用 ActionScript 脚本	405
14.3	DataGrid 组件	406
14.3.1	显示数据	406
14.3.2	获取行数据	408
14.3.3	自定义列	409
14.3.4	编辑数据	411
14.4	数据验证	414
14.4.1	数据验证组件概述	414
14.4.2	使用数据验证组件	415
14.4.3	验证触发方式	417
14.4.4	验证失败处理	420
14.4.5	自定义验证组件	422
14.4.6	数据验证应用实例	425
14.5	数据格式化	428
14.5.1	格式化组件概述	428
14.5.2	货币格式化组件 <mx:CurrencyFormatter>	429
14.5.3	日期格式化组件 <mx:DateFormatter>	431
14.5.4	数字格式化组件 <mx:NumberFormatter>	432
14.5.5	电话格式化组件 <mx:PhoneFormatter>	433
14.5.6	邮编格式化组件 <mx:ZipCodeFormatter>	434
第 15 章	数据传输与服务器交互	437
15.1	数据传输的方式	437

15.1.1 内部数据传输·····	437	16.4.1 视频播放和控制器·····	486
15.1.2 文件流方式传输·····	440	16.4.2 播放列表·····	490
15.1.3 XML 方式传输·····	441	16.4.3 调节器·····	491
15.1.4 其他方式传输·····	444	16.5 主程序设计·····	492
15.2 使用 HTTPService 与服务器端交互·····	445	第 17 章 视频展示网站 ·····	501
15.3 HTTPService 应用实例——留言本·····	447	17.1 系统概述·····	501
15.3.1 编写 ASP.NET 程序·····	447	17.1.1 需求分析·····	501
15.3.2 创建虚拟目录·····	451	17.1.2 结构设计·····	502
15.3.3 留言本界面与功能实现·····	452	17.2 数据库和数据库类设计·····	503
15.4 使用 WebService 与服务器端交互·····	464	17.2.1 数据库设计·····	504
15.5 WebService 应用实例·····	466	17.2.2 数据库类设计·····	505
15.5.1 编写服务器端程序·····	466	17.3 服务器端程序设计·····	509
15.5.2 编写 Flex 程序·····	468	17.3.1 处理用户程序文件·····	509
第 5 篇 综合实例篇		17.3.2 处理视频分类和视频列表 程序设计·····	512
第 16 章 功能齐全的 FLV 播放器 ·····	472	17.4 前台设计·····	515
16.1 系统概述·····	472	17.4.1 事件处理类设计·····	515
16.1.1 需求分析·····	473	17.4.2 用户模块设计·····	517
16.1.2 结构设计·····	473	17.4.3 分类模块设计·····	521
16.2 数据源、主题设计和色彩矩阵类的 创建·····	475	17.4.4 搜索模块设计·····	523
16.2.1 数据源文件及其格式设计·····	475	17.4.5 视频列表模块设计·····	526
16.2.2 程序主题设计·····	476	17.4.6 收藏夹及个人信息模块设计·····	530
16.2.3 色彩矩阵类的创建·····	479	17.4.7 整合主程序·····	531
16.3 编写事件类·····	484	17.5 后台设计·····	536
16.3.1 视频控制器事件类·····	484	17.5.1 添加类别·····	536
16.3.2 视频调节器事件类·····	484	17.5.2 修改和删除类别·····	538
16.3.3 播放列表事件类·····	485	17.5.3 添加视频·····	539
16.4 自定义组件设计·····	485	17.5.4 修改和删除视频·····	542
		17.5.5 整合主程序·····	543

第 1 篇 Flex 基础知识篇

第1章

Flex 3.0 入门



内容摘要 | Abstract

Flex 是一个提供开发设计和运行支持的架构, 可以使开发人员创建利用 Adobe Flash Player 9 作为前台的 RIA (Rich Internet Application, 富客户端互联网应用程序), 以满足用户更为直观和极具交互性的在线体验的要求。

本章以最新的 Flex 3.0 为例向读者详细讲解什么是 Flex、Flex 的组成部分、如何配置 Flex 开发环境以及开发第一个实例程序。



学习目标 | Objective

- 了解 RIA 和 Flex 的概念
- 理解 Flex 架构及其各部分的作用
- 了解 Flex 3.0 的新特性
- 熟悉 MXML 的命名规范和文件结构
- 熟悉 ActionScript 的使用方式
- 掌握 Flex 3.0 的环境配置方法
- 开发第一个 Flex 程序

1.1 Flex 概述

在 Ajax 和 Microsoft WPF 出现之前, Macromedia (现为 Adobe) 就曾推出基于 Flash 的 RIA 解决方案, 用于创建具有富交互和多功能的 Web 应用程序 (又称为 RIA)。现在, Adobe 对 Flash 进行了增强, 使其具有超越 Web 的功能, 从而成为一个完整的开发环境, 这就是 Flex。

1.1.1 RIA 发展

用户与目前的 Web 应用程序交互时, 其体验并不能令人满意, 主要体现在, Web 模型是基于页面的模型, 而且缺少客户端智能。因此即使是相对简单的事务处理 (例如网上购物) 也让人混淆。而且, 它几乎无法完成复杂的用户交互 (如传统的客户端/服务器应用程序和桌面应用程序中的用户交互)。这样的技术使得 Web 应用程序难以使用、支持成本高, 并且在很多方面无法发挥作用。

为了提高用户体验,出现了一种新类型的 Internet 应用程序,那就是 Rich Internet Application (以下简称 RIA)。这些应用程序结合了桌面应用程序反应快、交互性强的优点与 Web 应用程序传播范围广及容易传播的特性。RIA 简化并改进了 Web 应用程序的用户交互,这样,用户开发的应用程序可以提供更丰富、更具有交互性和响应性的用户体验。

Macromedia 是公认的新兴 RIA 市场的领导者。今天 98% 的浏览器上都使用 Macromedia Flash 客户端软件。因此几乎每个人都可以使用基于 Flash 的 RIA。Macromedia Flex 是 Macromedia 的新服务器产品,它使企业应用程序开发人员能够全面访问 RIA 的功能。Flex 具有基于标准的架构,与当前企业开发人员的工具、方法和设计模式互补。

1. 过渡

从最初的 HTML 到现在,服务器端系统架构经历了很多次重要转变。在此过程中,客户端的表现功能也有一些转变,并且每个阶段的计算功能所带来的应用程序体验也有变化,直到 RIA 出现为止。下面列举其中重要的 3 个方面。

□ 基于主机的应用程序

由基于主机的计算发展而来的交互式应用程序。推动此阶段计算的商业需求来自于企业组织内部业务自动化,例如工资表。应用程序在内部的专用网络间进行本地发布,用户界面的丰富性仅限于文本范围内。

□ 客户端/服务器应用程序

这种模式发展得很快,主要是因为需要对企业组织内部的信息及应用进行部门级别的访问。仍然是在企业组织内部本地访问应用程序,但随着图形用户界面的出现及客户端处理的应用,应用程序的丰富性大大提高。

□ Web 应用

Web 应用程序的全局性应用,以及基于主机集中管理应用程序的模式,突破了客户端/服务器模式的主要限制。但对于用户来说,这需要很大的投入。从处理的角度来看,Web 应用程序模式将客户端转变为虚拟终端。用于提供最佳用户体验的主要交互问题消失了,这些主要问题包括直接控制、客户端处理及局部存储等。

2. RIA 出现

Macromedia 公司于 2001 年初首先提出了 RIA 的概念。当时走在前沿的 Flash 开发者们都已经开始在实际应用开发中使用类似的模型来架构他们的程序。这些应用与传统的基于 HTML 的 Web 应用相比,扩展了设计的自由度,突破了用户的交互局限。

虽然 RIA 的优势很明显,但要求新技术要与现有的基础结构及处理过程相适合。这就需要满足如下要求。

□ 提供一个用户熟悉的编程模型

基于熟悉的面向对象的语言(像 Java 或者 C#)进行业务逻辑的后台开发,及利用基于标记的语言(像 XHTML)进行用户界面开发。新产品必须以现有的模型为基础进行创建,以充分利用现有的技术并确保较低的成本。

□ 利用现有的基础结构

很多企业组织在服务器应用技术上进行了大量投资,而且大部分企业组织将在内部使用

J2EE 及 .NET 技术开发产品。多数的企业组织也需要使用并符合该基础结构。

□ 采用标准的协议及 API

Web 的众多优点之一就是在整个技术中采用更宽的标准。这些标准包括但不局限于行业标准,如 HTML/HTTP、XML、SOAP、Web Service、CSS,还有 J2EE 及 .NET 等。

□ 保留现有的工具

对于采用表示层解决方案的开发人员来说,关键问题是能否确保使用现有的编辑器或集成开发环境 (IDE) 来编写应用程序。开发人员希望利用主要 IDE (如 Eclipse、Visual Studio) 和主要的 Web 应用程序开发产品 (如 Dreamweaver) 来编写代码。

□ 高效的生产工具

生产工具帮助人们使用新技术进行高效率的工作。它们还帮助人们学习新技术中新的语言、框架和结构,并指导人们进行最佳操作。

□ 支持主要的设计模型

企业在创建应用程序时使用设计模型进行企业开发。在过去的几年中,类似模型——视图-控制器 (MVC) 的模型在 J2EE 和 .NET 企业应用程序开发中越来越普遍。表示层解决方案必须适应现有的这些基于模型的架构。

由于架构一致性与基于标准的解决方案的要求,RIA 一直无法为多数的企业应用程序开发人员所利用。这种情况直到最近才有所改变。

1.1.2 Flex 简介

Flex 是 Adobe 的新产品,它满足了某些 IT 开发人员的需要。他们希望开发一种应用程序,这种程序既有桌面应用程序的响应性与丰富性,又具有 Web 传播范围广的特性,即 RIA。

Flex 驻留在组织的 N 层应用程序模型的表示层,使用在客户端运行的可执行代码,作为当前 HTML 的补充。Flex 具有基于标准的、用户熟悉的编程方法及 workflows,强大的类库可创建表示层,从而提供更有效、更真实的终端用户体验。

Flex 应用程序与传统的 HTML 应用程序的主要区别在于 Flex 应用程序处理最适合在客户端运行,如字段校验、数据格式、分类、过滤、工具提示、合成视频、行为及效果等。Flex 可使开发人员更好地执行应用程序,这种应用程序使用户可以迅速反应、在不同状态与显示时流畅过渡,并提供毫无中断的连续性工作流。

1. Flex 开发模型

Flex 开发模型对于使用 JSP、ASP/ASP.NET 或其他类似的脚本语言的开发人员来说并不陌生。基本的模型相同:建立一个包含应用程序源代码的文本文件,然后将此文件部署到服务器上;服务器在收到第一个请求时,将此源码编译成为应用程序,后续的请求将经过缓存处理。与发送一系列包含数据与布局的 HTML 页面不同,Flex 发送可在 Flash Player 虚拟机上运行的富客户端用户界面。需要时, Flex 应用程序将与服务器交换数据,以响应客户端上终端用户的操作。

Flex 开发者使用扩展的用户界面组件库与基于 XML 标记的语言定义用户界面,利用面向对象的脚本语言 (ActionScript) 来处理程序逻辑。

除了在现有的表示层上进行添加外, Flex 并不需要对当前的业务层与整合层进行任何改变。Flex 在应用服务器内运行, 并为 Flex 应用程序提供整合与管理能力。Flex 整合的能力可以轻松地通过 Web 服务、Java 对象访问或 XML 使用现有的代码及信息。Flex 还可以与一些现有的表示层技术与框架结构(如 JSP 及 Struts 等)进行集成。

2. Flex 的部署和管理

在 J2EE 平台上部署 Flex 服务器非常简单, 因为 Flex 是本地 Java 应用程序。在 J2EE 平台上部署 Flex 应用程序是通过 Java Web 包(WAR)文件处理的。从管理角度看, Flex XML 方案和基于文件的应用程序模型意味着, Flex 应用程序可以与现有的管理工具和应用程序生命周期工具轻松集成。在 Flash Player 中执行时, Flex 应用程序可以与服务器端的功能交互, 这些功能包括 Java 对象、SOAP Web 服务和其他服务器端服务。

1.1.3 Flex 架构

在图 1-1 中显示了 Flex 的基本架构, 可以看到 Flex 主要由 Flex 应用程序框架与 Flex 运行时服务构成。下面将针对架构中的重要部分进行简单介绍, 首先是 Flex 应用程序框架。

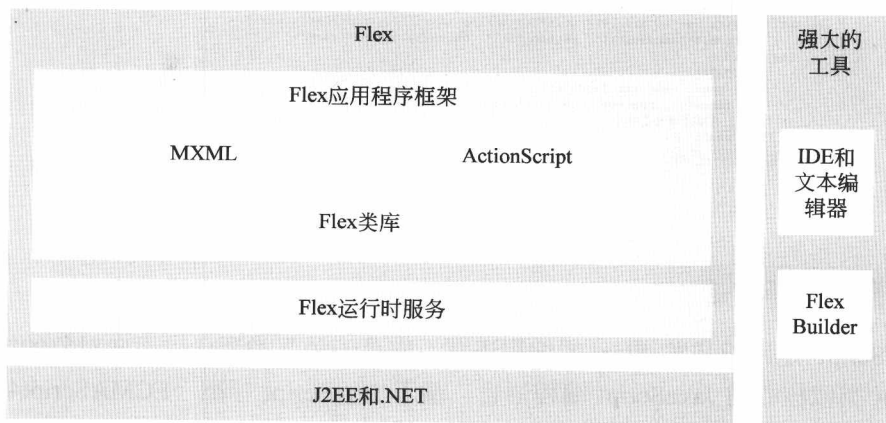


图 1-1 Flex 基本架构

1. Flex 应用程序框架

如图 1-1 所示, Flex 应用程序框架由 MXML、ActionScript 及 Flex 类库构成。开发人员利用 MXML 及 ActionScript 编写 Flex 应用程序。利用 MXML 定义应用程序用户界面元素, 利用 ActionScript 定义客户逻辑与程序控制。Flex 类库中包括 Flex 组件、管理器及行为等。利用基于 Flex 组件的开发模型, 开发人员可在程序中加入内置的组件、创建新组件或是将内置的组件加入复合组件中。

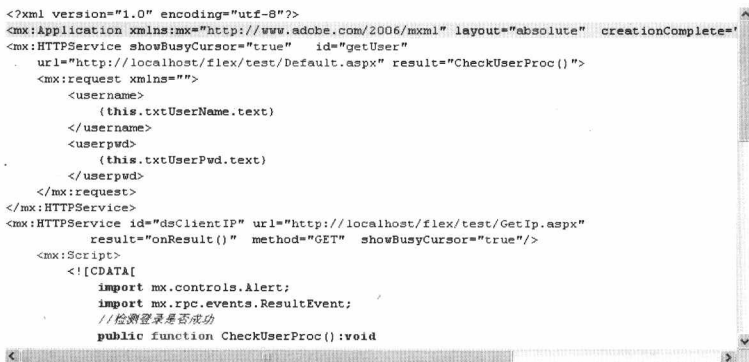
2. MXML: Flex 标记语言

MXML 和 HTML 一样, 是标记语言, 它描述了反映内容与功能的用户界面。与 HTML

不同的是, MXML 为表示层用户界面和服务器端数据绑定提供了声明的抽象。MXML 可将表示与业务逻辑的问题彻底分开, 以实现最大程度的提高开发人员的生产率及应用程序的重复使用率。

MXML 的开发基础是在迭代过程上。这与其他类型的 Web 应用程序文件相同, 如 HTML、JSP、ASP 及 ColdFusion 标记语言 (CFML)。开发 MXML 应用程序就像打开一个文本编辑器一样简单, 只要输入一些标签, 保存文件, 然后在 Web 浏览器上打开文件 URL 即可。

另外, MXML 文件同时也是普通的 XML 文件, 所以可以选择多种开发环境。可以在简单文件编辑器、专用 XML 编辑器或是支持文件编辑的集成开发环境 (例如 Flex Builder) 中进行开发。由于 MXML 符合 W3C XML 方案的定义, 也可以使用结构化编辑, 如代码着色和代码提示 (取决于编辑器的功能)。图 1-2 所示为片段 MXML 在开发环境 Flex Builder 中的效果。



```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" creationComplete="
<mx:HTTPService showBusyCursor="true" id="getUser"
  url="http://localhost/flex/test/Default.aspx" result="CheckUserProc()">
  <mx:request xmlns="">
    <username>
      {this.txtUserName.text}
    </username>
    <userpwd>
      {this.txtUserPwd.text}
    </userpwd>
  </mx:request>
</mx:HTTPService>
<mx:HTTPService id="dsClientIP" url="http://localhost/flex/test/GetIp.aspx"
  result="onResult()" method="GET" showBusyCursor="true"/>
<mx:Script>
  <![CDATA[
    import mx.controls.Alert;
    import mx.rpc.events.ResultEvent;
    //检测登录是否成功
    public function CheckUserProc():void
```

图 1-2 片段 MXML

3. ActionScript

ActionScript 是用于开发 Flex 的编程语言, 它是一种强类型化、面向对象的编程语言。ActionScript 类似核心的 JavaScript 编程语言, 基于 JavaScript 标准 (ECMAScript4)。Flex 开发者使用 ActionScript 来描述客户端逻辑。例如, 开发者使用 ActionScript 来定义事件监听器及句柄, 设置或获取组件属性的值及处理回调函数等。目前最新版本为 ActionScript 3.0。

4. Flex 类库

Flex 类库包括 RIA 的类库, 其中包含了 Flex 组件 (容器与控件)、数据绑定、行为及其他功能。Flex 类库为开发者提供一致的视觉提示、交互模式和应用程序导航惯例。

5. Flex 组件

基于组件的模型使 Flex 应用程序的开发简化。开发者可利用 Flex 中包含的内置组件和扩展组件来添加新的属性及方法, 创建新的组件。Flex 组件非常灵活, 可为开发者提供大量控件, 对组件外观、如何响应用户交互、文本的字体与字号、运行时的大小以及很多其他特征进行控制。

Flex 组件具有下列特点。

- ☐ **事件** 应用程序或需要组件响应的用户操作。
- ☐ **行为** 由应用程序或用户操作触发的可以看见或听见的变化。
- ☐ **纹理** 控制组件外观的图形。
- ☐ **样式** 各种特点的集合, 如字体、字号及文本的对齐等。
- ☐ **尺寸** 组件的高度与宽度 (所有的组件都有默认的尺寸)。

Flex 类库提供两种类型的组件: 容器和控件。开发者在使用 Flex 创建应用程序时, 使用控件与容器描述用户界面。控件是一种用户界面组件, 处理用户互动操作及显示可供用户直接通过该控件处理的数据, 像 DataGrid 与 TreeControl。容器定义了 Flash Player 绘图表面的区域, 控制容器内所有内容的布局, 包括其他的容器与控件。容器包括用于数据输入的表单容器、对话框及网格等。

Flex 组件具有下列特点。

- ☐ **MXML API** 用于声明控件及其属性值和事件等。
- ☐ **ActionScript API** 用于调用控件的方法及在运行时设置其属性、事件。
- ☐ **可利用样式、纹理、字体来定制外观与视觉效果。**

6. Flex 行为

Flex 类库也提供内置的行为, 可以使开发者在应用程序中容易地添加动画和声音等来为用户的操作提供相应的环境。例如, 开发者可以使用行为让对话框在收到焦点时轻微弹起。

行为是触发器与效果的结合体。触发是一个操作, 如鼠标单击组件或是组件显示出来。效果是组件在一段时间内 (单位为毫秒) 可见的变化。效果的例子有淡入淡出、移动、改变大小、模糊、暂停或擦除过渡等。

开发者可以为单一触发定义多重效果, 也可以按特定应用程序的需要定制效果或合成效果。

1.2 Flex 3.0

自从 Macromedia 推出 Flex 的第一个版本后, Flex 的发展速度令人惊讶。2004 年 3 月 Macromedia 正式推出 Flex 1.0 和 Flex Builder 1.0。现在看来, 这两个产品还是很成熟的。事实上, 当时 1.0 的推出并没有引起很多人的关注, 究其原因主要有两点: 一是价格非常昂贵; 二是运行环境只支持 J2EE, 需要 Java 的服务器环境才能运行。对于大部分的 Flash 程序员而言, Java 的学习门槛太高, 特别是没有网络编程经验的程序员, 很难接受一门学习周期很长的新语言。

当然, 软件本身的不成熟也是一个重要的原因。Flex Builder 1.0 的界面极其类似于网页编辑工具 Dreamweaver, 用户很容易上手。但用户普遍反馈编辑工具的速度太慢, 最后生成的 SWF 文件体积大。不过, Flex 的特性还是给很多人留下深刻的印象, 用纯粹的 XML 描述语言也可以生成 SWF, 这让很多人都觉得很神奇。

不久 Macromedia 便发布了升级补丁。2004 年 11 月, Macromedia 对 1.0 进行更大规模的