



免费提供实例源代码

信息科学与技术丛书

程序设计系列

ASP.NET 2.0

动态网站设计实例

郭馨君 孟清华 郭俊 等编著

- ◎ 涵盖 ASP.NET 2.0 技术
- ◎ 手把手教你实现 Web 应用
- ◎ 网上新闻发布系统开发实例
- ◎ BBS 系统开发实例
- ◎ 网上书店开发实例



机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书

程序设计系列

ASP.NET 2.0 动态网站设计实例

郭磬君 孟清华 郭 骏 等编著

机械工业出版社

ASP.NET 是 Microsoft 公司推出的 .NET 框架中用于 Web 应用的一种技术。本书介绍了 ASP.NET 2.0 的基础知识, 包括其发展历史和语法基础, 重点介绍 ASP.NET 2.0 中的各控件组及其特性, 并以作者开发的新闻发布系统、BBS 系统、网上书店三个完整的案例, 详细介绍了 ASP.NET 2.0 的应用开发过程。

本书可作为网络爱好者的自学教材, 也可作为网络开发工程师的参考书。

图书在版编目 (CIP) 数据

ASP.NET 2.0 动态网站设计实例 / 郭磐君等编著. —北京: 机械工业出版社, 2007.1

(信息科学与技术丛书. 程序设计系列)

ISBN 7-111-20230-9

I. A... II. 郭... III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2006) 第 128208 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 刘亚军

责任印制: 李 妍

唐山丰电印务有限公司印刷

2007 年 1 月第 1 版 · 第 1 次印刷

184mm×260mm · 22 印张 · 544 千字

0001—5000 册

定价: 33.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68326294

编辑热线电话 (010) 88379739

封面无防伪标均为盗版

出版说明

随着信息科学技术的迅速发展，人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书，来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术受到不断挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着因特网延伸到人类世界的层层面面，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容，注重理论与实践相结合，内容实用，层次分明，语言流畅，是信息科学与技术领域专业人员不可或缺的图书。

现今，信息科学技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

ASP.NET 是 Microsoft 公司推出的 .NET 框架中用于实现 Web 应用的一种技术。对于 Web 应用的开发人员来说,利用 ASP.NET 提供的丰富的编程模型可以轻松地进行开发,大大节省开发时间,提高开发质量。自正式发布 .NET Framework 1.0 到 .NET Framework 1.1 以来,其开发环境也从 Visual Studio 2002 发展到了 Visual Studio 2003。这期间,有不少的 Bug 得到了修正,不少的功能得到了改进,但仍然有不少的缺陷给开发人员带来困扰。例如,复杂的页面生命周期,大量冗余的 ViewsState 代码,对控件开发的可视化支持不够好,等等。

2005 年是值得所有 ASP.NET 应用的开发人员兴奋的一年,Microsoft 公司推出了其新一代的旗舰产品——Visual Studio 2005。这个版本支持的 ASP.NET 2.0 提供了大量用于提高开发速度的新控件,并将冗余的代码量尽可能地降低,其目标是将代码量降低 95%。实际上,虽然离这个目标还有一段距离,但与 Visual Studio 2003 相比确实有了质的飞跃。

本书的作者都是网站开发与 Web 应用技术研究方面的专家和技术权威。本书是作者多年的 Web 开发经验的总结。

本书共分 11 章:第 1、2 章介绍了 ASP.NET 2.0 的基础知识,包括其发展历史和语法基础;第 3~8 章介绍 ASP.NET 2.0 中的各控件组,包括标准控件组,站点导航控件组,安全设置、个性化和登录控件组,验证控件组,WebParts 控件组,数据控件组,MasterPage 和主题以及各控件组的特性;第 9~11 章以作者开发的新闻发布系统、BBS 系统、网上书店三个完整案例,详细介绍了 ASP.NET 2.0 的应用开发过程。

本书作者提供书中各实例的完整源代码,需要的读者可以登录机械工业出版社网站 www.cmpbook.com 下载。

参加本书编写的还有左昉、李广荣、刘昌丽、侯芳、赵黎、陈丽芹、王敏、袁涛、李鹏、周广芬、许洪、张玺、吴高阳等。

由于作者水平有限,书中不妥之处在所难免,恳请读者批评指正,并能联系 win760520@126.com 提出宝贵意见。

作 者

目 录

出版说明

前言

第1章 初识 ASP.NET 2.0 1

1.1 ASP.NET 2.0 概述 1

1.1.1 .Net 简介 1

1.1.2 从 ASP.NET 1.0 到

ASP.NET 2.0 3

1.2 ASP.NET 2.0 的内部变化 3

1.2.1 代码模型 3

1.2.2 编译 4

1.2.3 页面生命周期 6

1.2.4 可扩展性 8

1.2.5 高级缓存技术 10

1.2.6 性能 10

1.3 小结 11

第2章 ASP.NET 2.0 基础 12

2.1 C# 2.0 语言基础 12

2.1.1 数据类型 12

2.1.2 变量 20

2.1.3 类型转换 21

2.1.4 表达式 24

2.1.5 语句 25

2.2 C# 2.0 的新扩展 35

2.2.1 泛型 35

2.2.2 匿名方法 41

2.2.3 迭代器 44

2.2.4 分部类型 48

2.2.5 可空类型 49

2.3 小结 51

第3章 ASP.NET 2.0 的标准

控件组 53

3.1 ASP.NET 1.x 的标准控件组 53

3.1.1 Label 标签控件 53

3.1.2 Button、LinkButton、ImageButton

按钮控件 53

3.1.3 DropDownList 下拉菜单控件 54

3.1.4 ListBox 控件 54

3.1.5 CheckBox 和 RadioBuuton
控件 54

3.1.6 CheckBoxList 和 RadioButtonList
控件 54

3.1.7 BulletedList 控件 55

3.1.8 Panel 控件 55

3.1.9 View 和 MultiView 控件 55

3.1.10 DataList 和 Repeater 控件 55

3.2 新增控件简介 55

3.2.1 ImageMap 控件 55

3.2.2 BulletedList 控件 58

3.2.3 HiddenField 控件 60

3.2.4 FileUpload 控件 61

3.2.5 Wizard 控件 66

3.2.6 View 控件和 MultiView 控件 68

3.2.7 Substitution 控件 70

3.2.8 Localize 控件 71

3.3 小结 72

第4章 ASP.NET 2.0 的站点导航

控件组 73

4.1 站点导航控件综述 73

4.2 导航控件组 73

4.2.1 SiteMapPath 控件 73

4.2.2 Menu 控件 75

4.2.3 TreeView 控件 80

4.3 小结 85

第5章 ASP.NET 2.0 的安全设置、

个性化和登录控件组、验证

控件组 86

5.1 安全设置 86

5.2 个性化 86

5.3 登录控件组 87

5.3.1 Login 控件 87

5.3.2 LoginView 控件	92	7.3 数据展示控件	132
5.3.3 PasswordRecovery 控件	94	7.3.1 GridView 控件	132
5.3.4 LoginStatus 控件	97	7.3.2 DetailsView 控件	135
5.3.5 LoginName 控件	97	7.3.3 FormView 控件	141
5.3.6 CreateUserWizard 控件	97	7.4 小结	148
5.3.7 ChangePasword 控件	102	第 8 章 ASP.NET 2.0 的 MasterPage	
5.4 验证控件组	103	和主题	149
5.4.1 RequiredFieldValidator 控件	104	8.1 MasterPage 揭秘	149
5.4.2 RangeValidator 控件	107	8.1.1 动态使用 MasterPage	150
5.4.3 RegularExpressionValidator		8.1.2 在运行时访问 MasterPage	153
控件	108	8.1.3 MasterPage 和用户控件	153
5.4.4 CompareValidator 控件	110	8.2 主题	154
5.4.5 CustomValidator 控件	113	8.2.1 外观	154
5.4.6 ValidationSummary 控件	117	8.2.2 主题的应用范围	154
5.5 小结	119	8.2.3 定义主题	155
第 6 章 ASP.NET 2.0 的 WebParts		8.2.4 使用主题	155
控件组	120	8.3 小结	156
6.1 WebParts 控件组综述	120	第 9 章 综合实例——新闻发布	
6.2 WebParts 控件组	120	系统	157
6.2.1 WebPartsManager 控件	120	9.1 系统概述	157
6.2.2 ProxyWebPartsManager 控件	122	9.2 系统功能设计	157
6.2.3 WebPartsZone 控件	122	9.3 系统模块设计	158
6.2.4 CatalogZone 控件	122	9.4 数据库设计	159
6.2.5 EditorZone 控件	123	9.4.1 实体分析	159
6.2.6 AppearanceEditorPart 控件	123	9.4.2 数据库设计	160
6.2.7 BehaviorEditorPart 控件	124	9.5 系统实现	163
6.2.8 LayoutEditorPart 控件	124	9.5.1 系统总体设计	163
6.2.9 PropertyGridEditorPart 控件	124	9.5.2 系统的实现	168
6.2.10 ConnectionZone 控件	124	9.5.3 系统运行效果	221
6.3 WebParts 应用实例	124	9.6 小结	222
6.3.1 创建 Web 部件页	124	第 10 章 综合实例——BBS 系统	223
6.3.2 使用户能够编辑页面和更改		10.1 系统概述	223
布局	127	10.2 系统功能设计	224
6.3.3 在运行时添加 WebParts	130	10.3 系统模块划分	224
6.4 小结	130	10.4 数据库设计	225
第 7 章 ASP.NET 2.0 的数据		10.4.1 实体设计	225
控件组	131	10.4.2 数据库设计	228
7.1 数据控件组综述	131	10.5 系统实现	230
7.2 数据源控件 DataSource	131	10.5.1 系统总体设计	230

10.5.2 系统的实现	234	11.4.1 实体设计	285
10.5.3 系统运行效果	281	11.4.2 数据库设计	288
10.6 小结	282	11.5 系统实现	292
第 11 章 综合实例——网上书店	283	11.5.1 系统总体设计	292
11.1 系统概述	283	11.5.2 系统的实现	298
11.2 系统功能设计	284	11.5.3 系统运行效果	342
11.3 系统模块划分	285	11.6 小结	344
11.4 数据库设计	285		

第 1 章 初识 ASP.NET 2.0

1.1 ASP.NET 2.0 概述

1.1.1 .Net 简介

按照 Microsoft 公司的定义：.NET 就是 Microsoft 公司的 XML Web 服务平台。不论操作系统或编程语言有何差别，XML Web 服务都能使应用程序在 Internet 上传输和共享数据。

Microsoft® .NET 平台包含广泛的产品系列，它们都是基于 XML 和 Internet 行业标准构建的，并提供从开发、管理、使用到体验 XML Web 服务的每一方面。Microsoft 公司已在以下五个方面创建 .NET 平台。

1. .NET 体验

.NET 体验是 XML Web 服务，通过它，可以从 Internet 和独立的应用程序中以集成的方式访问信息，做到省时省钱。Microsoft 公司将发布面向个人和面向企业的不同的 .NET 体验。Microsoft 公司的一些正在过渡到 .NET 体验的产品包括 MSN®、bCentral™、Passport 和 Microsoft Visual Studio® .NET。

2. 客户端

客户端是指 PC、工作站、电话、笔记本电脑、Tablet PC 和游戏控制台以及其他智能设备。这些设备之所以“智能”，是由于其能够访问 XML Web 服务。智能客户端使用支持 XML Web 服务的软件，不论您使用的客户端的位置、类型还是数量如何，您都可以访问您的数据。Microsoft 公司提供的一些 .NET 客户端软件包括 Windows® CE、Windows Embedded、Windows 2000 和 Windows XP。

3. 服务

除开发人员创建 XML Web 服务之外，Microsoft 公司创建了一个包含构造块服务的核心集。这些构造块服务用来执行例行任务，并可作为主体，供开发人员在其上创建自己的服务。第一个创建的代号为“HailStorm”的 XML Web 服务集是以用户为中心的服务，服务于人而不是特定的设备、网络或者应用程序。HailStorm 基于 Microsoft Passport 用户身份验证系统。通过 HailStorm，用户在需要的时候可以接收相关信息，并可根据预先建立好的首选项将其传递给正在使用的设备。

XML Web 服务与 Web 站点的不同之处在于：Web 站点是把信息呈现给用户，它们是服务器向用户传递信息的媒介，而 XML Web 服务提供应用程序与应用程序之间交互的直接手段。应用程序以本地为宿主，也可以以远程系统为宿主，利用 XML 和 SOAP 消息通过 Internet 进行交互。

4. 服务器

Microsoft .NET 企业级服务器为部署、管理和编排 XML Web 服务构筑了 Microsoft .NET

服务器结构。由于在设计中考虑了工程的关键性能, 这些服务器不仅可以提供给企业足够的敏捷性, 以满足他们通过 XML Web 服务将系统、应用程序与合作伙伴集成的需要, 而且其灵活性也可适应不断变化的业务要求。Microsoft.NET 企业级服务器包括 Microsoft Windows® 2000 服务器系列:

- Microsoft Application Center 2000, 用以部署和管理高可用性和可扩展的 Web 应用程序;
- Microsoft BizTalk™ Server 2000, 用以跨应用程序和组织的构建基于 XML 的业务过程;
- Microsoft Commerce Server 2000, 用以快速构建可扩展的电子商务解决方案;
- Microsoft Content Management Server 2001, 用以进行动态电子商务 Web 站点的内容管理;
- Microsoft Exchange Server 2000, 可以在任何时间任何地点实现消息传递和协作;
- Microsoft Host Integration Server 2000, 用以桥接旧式系统上的数据和应用程序;
- Microsoft Internet Security and Acceleration Server 2000, 用以进行安全、快速的 Internet 连接;
- Microsoft Mobile Information 2001 Server, 用以使应用程序可以被诸如蜂窝电话这样的移动设备支持;
- Microsoft SharePoint™ Portal Server 2001, 用以查找、共享和发布商业信息;
- Microsoft SQL Server™ 2000, 用以存储、检索和分析结构化 XML 数据。

5. 工具

Microsoft Visual Studio .NET 和 Microsoft .NET 框架为开发人员创建、部署和运行 XML Web 服务提供了一套完整的解决方案。它们可使 XML Web 的性能、可靠性和安全性达到最佳化。

Visual Studio .NET 是 Microsoft 公司专为 .NET 构建的下一代主流多语言开发工具。Visual Studio .NET 可以帮助开发人员使用他们自己选用的语言快速地创建可缩放的 XML Web 服务和应用程序。Visual Studio .NET 提升了下列高产的编程语言: Microsoft Visual Basic, 它包含了新的面向对象的编程功能; Microsoft Visual C++, 它提升了 Windows 开发并可用于创建 .NET 应用程序; C#, 它给 C 和 C++ 开发人员带来了 RAD (Rapid Application Develop, 快速应用开发)。

.NET 框架是一个高产的、基于标准的、多语言应用程序的执行环境, 它处理基本管线杂务, 使部署更容易。它以提供应用程序的执行环境来管理内存和进行版本控制, 改善了应用程序的可靠性、可扩展性和安全性。.NET 框架由多个部分组成, 包括公共语言运行库 (用来生成 XML Web 服务的类库集), 还包括 ASP.NET。

以上是在 Microsoft 公司的网站上找到的对 .Net 的定义, 虽然经过这么多年的发展, 上面的定义有些陈旧, 不过仍然可以给读者一个感性的认识。总的来说, .Net 是很多技术的合集, 是 Microsoft 公司的一个战略。在 .Net 战略中, .Net Framework 占有很重要的地位, 它是 Microsoft 公司为其 .Net 战略而提出的用于开发下一代应用程序和 Web 服务的一个框架。它包括 Web 服务、Web 应用 (ASP.NET)、数据存储技术 (ADO.Net)、Windows Forms 等, 其中 ASP.NET 是本书介绍的重点。

1.1.2 从 ASP.NET 1.0 到 ASP.NET 2.0

从上一小节可知, ASP.NET 是 Microsoft 公司推出的 .NET 框架中用于实现 Web 应用的一种技术。

从 .Net 的提出到现在已经过去了好多年, .Net Framework 的版本从当初的 1.0 经过 1.1, 直至 2005 年提出的 2.0, 相应的 ASP.NET 也从 1.0 发展到了 2.0。

ASP.NET 2.0 的推出不像 ASP.NET 1.0 到 ASP.NET 1.1 那样只是对原有的东西进行修补, 而是对 ASP.NET 技术的一次重大升级。它不仅在原来的基础上增加了许多新的控件, 而且还有许多内部变化。

1.2 ASP.NET 2.0 的内部变化

ASP.NET 2.0 较 ASP.NET 1.1 有许多内部变化。本节中, 其内部变化介绍的部分内容来自 MSDN 中文官方网站, 希望对读者有所帮助。

1.2.1 代码模型

ASP.NET 2.0 内部工作方式最明显的变化可能与如何创建 ASP.NET Web 页有关。在本节中, 我们将研究对代码隐藏模型的更改以及这些更改如何影响 ASP.NET 开发。

1. ASP.NET 1.x 中的编码模型

在 ASP.NET 1.x 中, 开发人员开发 Web 窗体主要有两种选择。

1) 开发人员可以采用传统的 ASP 模型并直接在 ASPX 页中编写代码。这个过程称为代码内联, 对于简单的命令非常奏效。但是, 对于较复杂的代码, 编写代码内联将导致难以阅读呈现 (HTML) 与功能 (代码) 混合的 Web 页。

2) 可以将业务逻辑和事件处理代码写入一个只有代码的文件中, 该文件称为代码隐藏文件。代码隐藏模型将一个只有代码的文件与包含呈现标记的 ASPX 文件连接起来。通过将代码和呈现分离, 允许设计人员处理呈现文件而让开发人员处理代码文件, 使开发小组的工作速度得以提高。ASP.NET 1.x 中的编码模型如图 1-1 所示。

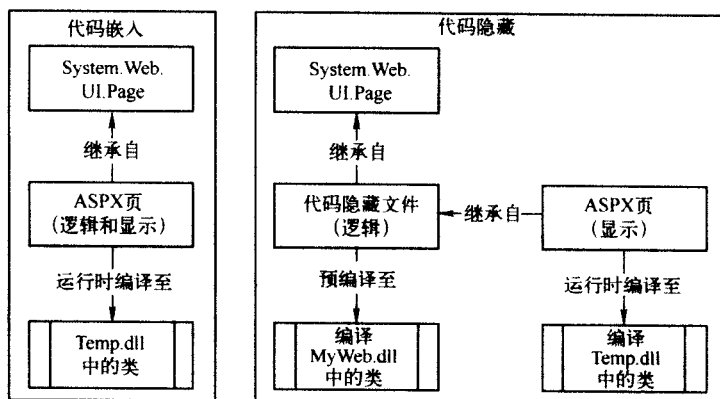


图 1-1 ASP.NET 1.x 的编码模型

使用代码隐藏模型的主要困难和代码隐藏文件必须与 ASPX 页面同步的方式有关。虽然从编程的意义上来说, ASPX 页继承了代码隐藏文件,但实际上两个文件通过一种更为复杂的关系连接在一起。

这种模型带来很多不便,如继承的复杂性、编译的复杂性等。

2. ASP.NET 2.0 中的编码模型

ASP.NET 2.0 继续提供代码内联和代码隐藏编码模型。就代码内联模型而言,除了 Microsoft Visual Studio 支持单文件开发之外,几乎没有变化。

ASP.NET 2.0 通过修改代码隐藏文件的特性,解决了代码隐藏模型的继承和编译问题。在 ASP.NET 2.0 中,代码隐藏文件不再是 System.Web.UI.Page 类的完整实现。取而代之的是,代码隐藏文件是一种称为局部类的新结构。这种局部类包含所有用户定义的代码,但是省略由 Visual Studio .NET 在 ASP.NET 1.x 中自动生成的所有基础结构和连接代码。当请求一个具有新代码隐藏文件的 ASPX 页面时,ASP.NET 2.0 运行时会真正地将 ASPX 页和局部类合并为一个类,而不是两个单独的类。局部类利用一个新关键字(在 Visual Basic 中为 Expands,在 C#中为 Partial)来表明该类中的代码应当在运行时与另一个类合并。同样,ASPX 页利用一个称为 compilewith 的新指令来表明它与代码隐藏文件的结合。ASP.NET 2.0 中的代码隐藏模型如图 1-2 所示。

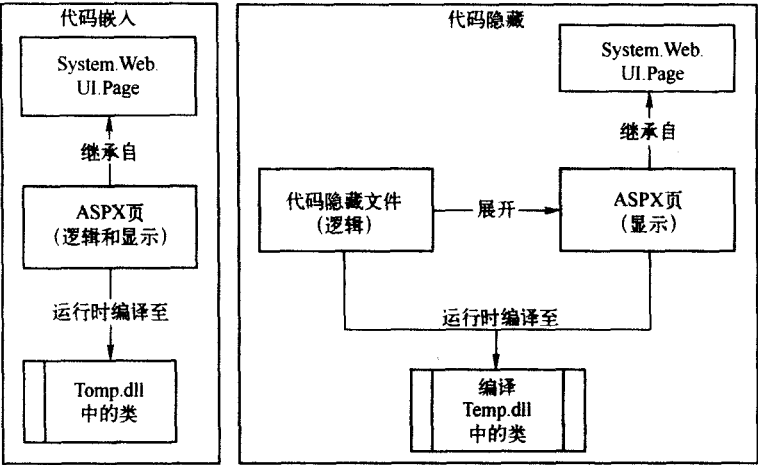


图 1-2 ASP.NET 2.0 中的代码隐藏模型

这种编码模型大大降低了继承和编译的复杂性。

1.2.2 编译

由于 ASP.NET 1.x 中引入了页模型, ASP.NET Web 页的编译过程总是分成两个阶段。首先,代码隐藏文件和其他支持类编译到一个程序集中,然后在运行时编译单独的 ASPX 文件。虽然这种模型具有许多优点,但它也有一些缺点。ASP.NET 2.0 提供基本模型的几种替代模型,并根据特定需要提供更为广泛的编译选项。

1. ASP.NET 1.x 中的编译

ASP.NET 1.x 中的主要编译模型会导致一个应用程序程序集(包含所有的已编译代码的

隐藏文件和其他的源代码) 和一个为每个被请求的 ASPX 页面而创建的临时程序集。在有些情况下, 编译器优化 (如批处理) 会引起将所有临时 ASPX 页面编译到同一个程序集中。在任一种情况下, 所有 ASPX 页面都被编译到一个临时程序集中, 可以被加载到 ASP.NET 运行库。ASP.NET 1.x 中的编译模型如图 1-3 所示。

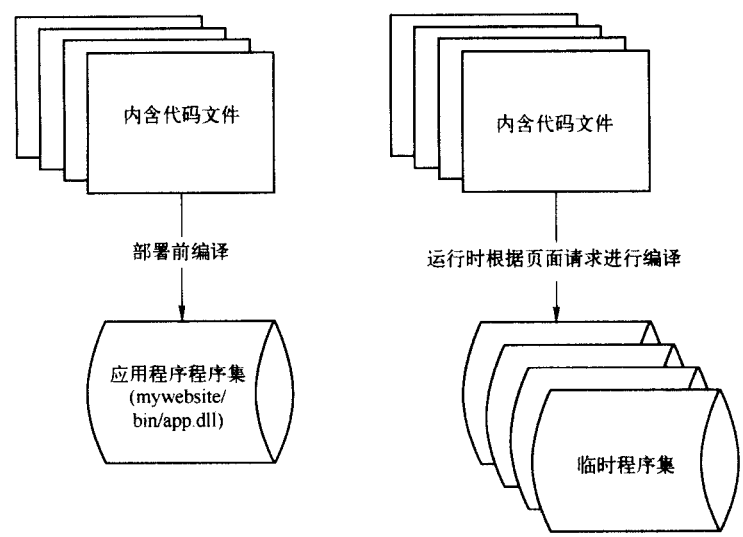


图 1-3 ASP.NET 1.x 中的编译模型

虽然这种模型有优点, 但它也有两个主要缺点。首先, ASPX 页面必须要以人们可以阅读的形式部署到 Web 站点。如果开发人员使用代码内联模型, 这意味着, 一些 (或所有) 的业务逻辑也可以部署在生产服务器上。虽然没有将 IIS 和 ASP.NET 配置为公开原始的 ASPX 页面, 但聪明的攻击者仍可以通过任何攻击打开到 Web 服务器的通道来访问这些文件。其次, 第一次有人请求某 Web 页时, 响应速度将比正常速度慢一些, 原因在于 ASP.NET 运行库必须编译 ASPX 页面。

2. ASP.NET 2.0 中的编译

ASP.NET 2.0 为 Web 应用程序提供了以下三种不同的编译模型。

(1) 普通编译 (ASP.NET 1.x)

在一个普通的 ASP.NET Web 应用程序中, 代码隐藏文件被编译到一个程序集并存储在 \bin 目录中。根据要求编译 Web 页 (ASPX)。该模型对大多数 Web 站点都运行得不错。但是, 编译过程使得第一次请求 ASP.NET 页时的速度比随后的请求速度缓慢。ASP.NET 2.0 继续支持这种编译模型。

(2) 部署预编译

ASP.NET 2.0 的一种新功能, 允许在部署前对项目进行完整编译。在完整编译中, 所有的代码隐藏文件、ASPX 页面、HTML、图形资源以及其他的后端代码都被编译到一个或多个可执行程序集中, 这取决于应用程序的大小和编译设置。这些程序集包含所有的已编译 Web 站点代码, 而资源文件和配置文件被复制, 没有做修改。这种编译方法以牺牲修改部署后 Web 站点的能力为代价, 提供了最好的性能和安全性。如果您使用高可见或高安全的 Web 站点,

这种选项是最终部署的最好选择。但是，如果您正在构建一个运行局部 Intranet 的小站点，并且更改站点非常频繁，那么完整预编译可能有点过分。

(3) 完整的运行时编译

在部署预编译的另一个极端，ASP.NET 2.0 提供一种在运行时编译整个应用程序的新机制。也就是说，可以将未编译的代码隐藏文件和其他相关的代码放在\app_code 目录中，并让 ASP.NET 2.0 创建和维护对程序集的引用，这些引用将在运行时根据这些文件生成。这种选项以在服务器上存储未编译代码为代价，在更改 Web 站点内容方面提供了最大的灵活性。

1.2.3 页面生命周期

ASP.NET 2.0 在 ASP.NET 页的生命周期中提供了两个主要变化：第一，ASP.NET 2.0 公开新的事件来支持新功能，包括母版页、个性化以及集成的移动设备支持；第二，ASP.NET 2.0 为 Web 窗体引入跨页投递。

1. 新事件

与 ASP.NET 1.x 相比，ASP.NET 2.0 提供了一种粒度更细的页面生命周期方法栈。这些添加的方法为 Web 开发人员提供了更高级的控制。可以通过 ASP.NET 页上的 Page 对象来访问这些事件。

表 1-1 为显示综合方法列表。方法列显示实际事件方法名称，活动列表示该事件是否总处于活动状态或者仅在 PostBack 操作期间是活动的。例如，一方面，新方法 TestDeviceFilter 可用来确定哪种设备筛选器就绪，并用这些信息来决定如何显示页面；另一方面，新方法 LoadControlState 只有在 PostBack 期间才激发。可以重写该方法（结合 SaveControlState）来在 PostBack 期间创建保存和还原控件状态的备选序列化方案。

表 1-1 显示综合方法列表

方 法	活 动
Constructor	Always
Construct	Always
TestDeviceFilter	Always
AddParsedSubObject	Always
DeterminePostBackMode	Always
OnInit	Always
LoadPersonalizationData	Always
InitializeThemes	Always
OnInit	Always
ApplyControlSkin	Always
ApplyPersonalization	Always
OnInitComplete	Always
LoadPageStateFromPersistenceMedium	Post Back
LoadControlState	Post Back

(续)

方 法	活 动
LoadViewState	Post Back
ProcessPostData1	Post Back
OnPreLoad	Always
OnLoad	Always
ProcessPostData2	Post Back
RaiseChangedEvents	Post Back
RaisePostBackEvent	Post Back
OnLoadComplete	Always
OnPreRender	Always
OnPreRenderComplete	Always
SavePersonalizationData	Always
SaveControlState	Always
SaveViewState	Always
SavePageStateToPersistenceMedium	Always
Render	Always
OnUnload	Always

查看页面生命周期的底层细节，可以看到 ASP.NET 2.0 中提供的许多功能（如主题和个性化）将在什么地方容易实现。例如，主题在 `InitializeThemes` 事件中处理，而个性化数据将在 `LoadPersonalizationData` 中加载并稍后用于 `ApplyPersonalization` 方法。请注意，就哪一个 UI 元素将决定 Web 应用程序的最终外观和感觉而言，方法的顺序非常重要。

2. 跨页投递

页面生命周期中的另一个主要变化涉及回发事件和 Web 窗体。在 ASP.NET 1.x 中，Web 窗体自动回发给它们的宿主页，即当用户提交一个窗体时，窗体数据总是提交回包含该原始窗体的页面。这种设计决策考虑了存储控件状态的容易性，但限制了开发人员执行更复杂操作的能力。

在 ASP.NET 2.0 中，Web 窗体控件有一个新特性，让开发人员决定在进行提交操作时将窗体数据发送到何处。在大多数情况下，要求 `PostBack` 机制，因此它仍是默认的机制。但是，如果开发人员想将数据投递到不同的窗体，现在是可能的，如图 1-4 所示。

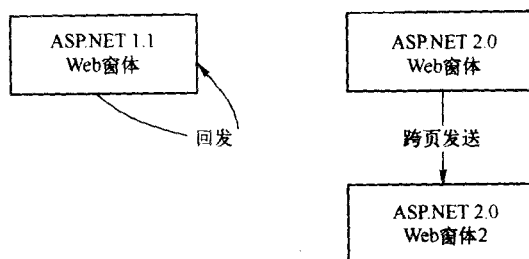


图 1-4 `PostBack` 与跨页投递

1.2.4 可扩展性

ASP.NET 最初被设计为一种开放式框架，即可以扩展、修改或替换构成 ASP.NET 的许多模块和组件，以适应特定的要求。在 ASP.NET 2.0 中，这种框架的可扩展特性由新的 HTTPHandlers 和 HTTPModules 清楚地阐明，二者现在是该框架的一个标准部分。

1. 请求管道

请求管道如图 1-5 所示。

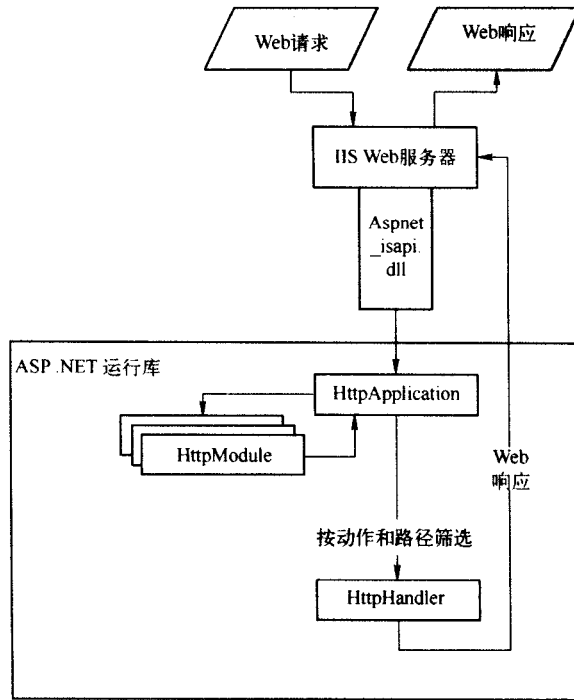


图 1-5 请求管道

请求管道是指：在 ASP.NET 中，请求从 Web 服务器通过 Internet 服务器应用程序编程接口（ISAPI）筛选器传递，并继续传递给实际的 ASP.NET 运行库。当 IIS 接收一个请求时，根据 IIS 的设置将扩展名映射到一个 ISAPI 筛选器，即将 .ASPX、.asmx、.asd 和其他扩展名映射到 ASPnet_isapi.dll。ASPnet_isapi.dll 只是一种启动 ASP.NET 运行库的 ISAPI 筛选器。一旦请求到达 ASP.NET 运行库，它在 HTTPApplication 对象处启动，该对象担当 ASP.NET Web 应用程序的宿主。

HTTPApplication 对象具有以下功能：

- 1) 读取机器级和应用程序级的配置文件。
- 2) 通过一个或多个 HTTPModule 实例传递请求。每个 HTTPModule 提供一种服务，如会话维护、身份验证、配置文件维护。这些模块将请求传递回 HTTPApplication。
- 3) 根据谓词和路径将请求传递给 HTTPHandler。谓词指请求中使用的 HTTP 谓词（GET、POST、FTP 等），而路径指应用程序中的 URL。根据处理程序的配置方式，该请求可能作为

一个 ASP.NET 页 (System.Web.UI.Page 为 IHttpHandler 的一种实现) 加以处理, 或者该请求可能触发另一个操作。例如, 批编译所有的 Web 页 (precompilation.asd 触发 PrecompHandler)。

在 ASP.NET 2.0 中, 该模型没有变化, 但是, 添加了几种新模块和处理程序以提供其他的 service。与 ASP.NET 1.x 一样, 您可以扩展、替换或重新配置任何模块或处理程序类, 以提供自己的自定义功能。

2. 新模块

显然, 已经添加了新 HTTPModules 以支持 ASP.NET 2.0 中提供的新 service。具体地说, 具有默认模块设置的 ASP.NET 应用程序将包括为以下目的而添加的新模块。

1) SessionID。会话识别机制已经从 ASP.NET 1.x 会话模块分离, 以提供对 Cookie、URL 重写以及会话 ID 生成的其他形式的更多控制。

2) 角色管理。这种新模块用于提供基于角色的 service, 以支持新用户识别机制。该模块有助于将 ASP.NET 应用程序和内置在 .NET 框架中基于角色的安全性结合起来。

3) 匿名识别。新的个性化功能支持匿名用户。该模块有助于跟踪匿名用户可以访问的功能, 以及跟踪在请求之间维护这些功能的方式。

4) 配置文件。该配置文件模块连接新的配置文件 service, 为用户提供特定的持久数据存储。

3. 新处理程序

除了这些新模块, ASP.NET 2.0 还引进了新的处理程序以支持应用程序配置工具以及其他的新功能, 如批编译请求。这些新处理程序中最重要的一点是包括处理 Web 站点管理请求的 “.axd” 系列。这些处理程序启动内部的管理工具, 这些管理工具允许开发人员配置 ASP.NET 用户和其他设置。管理处理程序包括:

1) Web 管理。WebAdminHandler 是管理 Web 站点的主页。该处理程序为管理 ASP.NET 2.0 Web 应用程序提供了起点。

2) 跟踪。ASP.NET 1.x 中的 TraceHandler 已进行了改进, 它是 ASP.NET 1.x 中唯一的 “.axd” 处理程序。

3) Web 资源。由于有了新的管理工具和 WebResourcesHandler, 现在可以在部署后配置 Web 资源。

4) 缓存图像。CachedImageServiceHandler 支持缓存图形组件。

5) 计数器。SiteCountersHandler 使用页面计数器模块为 ASP.NET 2.0 应用程序提供访问统计信息。

6) 预编译。正如先前提到的一样, 可以使用 PrecompHandler 在一个 ASP.NET 应用程序中批编译所有的 ASPX 页。

7) Web 部件导出。WebPartExportHandler 支持存储和传输 Web 部件布局。Web 部件是一种新机制, 用于个性化门户样式的 Web 应用程序的外观和内容。

与以前一样, HTTPForbiddenHandler 连接到不应该被返回的任何文件类型。在 ASP.NET 2.0 中, 禁止的文件类型列表已扩展为包括母版页、外观文件和其他的开发新控件。