

21 世纪高等学校计算机应用技术规划教材



ASP.NET Web 程序设计与应用 (第2版)

汪维清 汪维华 主编



清华大学出版社

21 世纪高等学校计算机应用技术规划教材

ASP.NET Web 程序设计与应用 (第 2 版)

汪维清 汪维华 主编

清华大学出版社
北京

内 容 简 介

本书以 Visual Studio 2013 和 Windows 7 为开发平台,以 C# 为编程语言,通过一系列实例详细地介绍 ASP.NET 网站开发技术,包括 C# 的基本语法知识、常用控件的基本使用方法、动态数据库连接技术、ASP.NET 程序开发基础、XML 和 Web 服务等内容。全书共 9 章,既包括基本的原理及语法,也包含大量编程技术细节和技巧。为了便于学生理解,各章均配有习题。

本书可作为各专业的学生或工程技术人员学习 ASP.NET 程序的教材,也可作为学习 ASP.NET 编程的初级和高级读者的参考用书。

为方便教师教学和自学,本书配有电子教案,可到清华大学出版社网站本书页面下载。

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

ASP.NET Web 程序设计与应用/汪维清,汪维华主编.--2 版.--北京:清华大学出版社,2015
21 世纪高等学校计算机应用技术规划教材
ISBN 978-7-302-40075-2

I. ①A… II. ①汪… ②汪… III. ①网页制作工具—程序设计—高等学校—教材
IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2015)第 089621 号

责任编辑:刘 星
封面设计:杨 兮
责任校对:徐俊伟
责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

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

课 件 下 载: <http://www.tup.com.cn>,010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:17.75 字 数:450 千字

版 次:2011 年 8 月第 1 版 2015 年 8 月第 2 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:063915-01

ASP.NET 是一种建立在通用语言上的程序构架,能被用于一台 Web 服务器来建立强大的 Web 应用程序。ASP.NET 具有许多比现在的 Web 开发模式强大的优势。本书是作者结合多年的教学和项目开发经验,并根据近几年教学改革的实践以及对人才培养的要求而编写的。

本书的目的是以 Visual Studio 2013 和 Windows 7 为开发平台,以 C# 为开发语言,通过一系列实例详细地介绍 ASP.NET 网站开发技术的基础知识和工作原理,综合运用文字、图形和表格加强读者对教学内容的理解。

本书面向毫无程序设计基础的读者,采用案例方式,全面详细地介绍 C# 的基本语法知识、常用控件的基本使用方法、动态数据库连接技术、ASP.NET 程序开发基础、XML 和 Web 服务等。全书共 9 章,各章的主要内容如下。

第 1 章介绍动态网页(ASP)、.NET Framework 与 ASP.NET 各自的特点及其之间的关系,详细介绍 IIS 服务器的安装与配置,并通过一个 ASP.NET 实例来初步认识开发网站的过程。

第 2 章介绍 Visual C#.NET 程序设计基础,包括 Visual C#.NET 基本数据类型、常量和变量、表达式和程序基本结构等内容。

第 3 章介绍网页设计基础知识,包括静态网页、动态网页的基本概念,特别是详细讨论了 ASP.NET 的各种控件的使用方法,并通过开发实例介绍 ASP.NET 各类常用控件。

第 4 章介绍网页布局基本知识,包括站点建立、母版页和内容页的设计以及网站导航基本控件的使用等内容。

第 5 章介绍基本内置对象基础,包括 Page 对象、Cookie 对象、Request 对象、Response 对象、Application 对象和 Session 对象等。通过本章的学习,读者能轻松地在两个网页之间传递变量、输出数据以及记录变量值等。

第 6 章介绍数据库相关操作,包括 ADO.NET 数据库开发原理、Connection 对象、Command 对象、DataReader 对象、DataAdapter 对象、DataSet 对象、数据绑定和数据控件访问数据库等数据库开发技术基础。

第 7 章介绍有关文件与文件夹的基本操作、读写文件、文件的上传与下载等内容。

第 8 章介绍 XML 和 Web 服务的基本概念,XML 文件的读写,Web 服务的创建、发布与使用等内容。

第 9 章介绍与网站发布相关的内容,包括 IIS 的配置、FTP 的配置、网站的发布、Web 安装程序的创建以及 Web 网站安装等。

本书是基于 C# 介绍 ASP.NET 程序开发技术,书中的所有例程都在 Visual Studio 2013 编程环境中测试通过,能够非常方便地设计出 Windows 程序的图形用户界面,去掉了

以前版本中那些晦涩难懂的托管语法形式，简化了编程过程，程序代码更加清晰易懂。

本书由汪维清、汪维华主编，其中，第1~5章由汪维清编写，第6~9章由汪维华编写。

动态网页程序设计是一项不断发展的程序技术，ASP.NET更是博大精深，鉴于作者水平有限，经验不足，书中难免存在错误和不当之处，敬请专家、同行和读者批评指正。

编 者

2015年3月

第 1 章 ASP.NET 基础	1
1.1 ASP.NET 概述	1
1.1.1 ASP 简介	1
1.1.2 ASP 工作原理	2
1.1.3 ASP.NET 简介	5
1.2 .NET Framework	6
1.2.1 .NET Framework 的作用	6
1.2.2 .NET Framework 的组成	7
1.3 ASP.NET 服务安装与配置	9
1.3.1 安装 IIS 服务器	9
1.3.2 安装开发工具	19
1.4 第一个 ASP.NET 程序	24
习题 1	28
第 2 章 C# 编程语言基础	29
2.1 C# 程序构成	29
2.2 C# 数据类型	33
2.2.1 值类型	34
2.2.2 引用类型	38
2.3 常量、变量与表达式	40
2.3.1 常量	40
2.3.2 变量	42
2.3.3 运算符与表达式	45
2.3.4 运算符的优先级与结合性	50
2.3.5 类型转换	51
2.3.6 数组 Array	53
2.4 基本结构	56
2.4.1 顺序结构	56
2.4.2 选择结构	58
2.4.3 循环结构	65
2.4.4 break、continue 与 return 语句	68
习题 2	71

第3章 网页设计	72
3.1 网页设计概述	72
3.2 静态网页设计	73
3.3 动态网页设计	75
3.3.1 ASP	75
3.3.2 ASP.NET	76
3.4 HTML 控件	77
3.4.1 HtmlForm 控件	79
3.4.2 HtmlInputText 控件	79
3.4.3 HtmlInputButton 控件	80
3.4.4 HtmlTable 控件	82
3.4.5 HtmlSelect 控件	85
3.4.6 HtmlImage 控件	85
3.5 Web 服务器端控件	87
3.5.1 ASP.NET Label 控件	87
3.5.2 ASP.NET TextBox 控件	88
3.5.3 ASP.NET Button 控件	88
3.5.4 ASP.NET HyperLink 控件	91
3.5.5 ASP.NET Image 控件	93
3.5.6 ASP.NET DropDownList 控件	94
3.5.7 ASP.NET CheckBox 控件	96
3.5.8 ASP.NET RadioButton 控件	99
3.6 Web 表单验证控件	102
3.6.1 RequiredFieldValidator 控件	103
3.6.2 RangeValidator 控件	105
3.6.3 ValidationSummary 控件	105
3.6.4 CompareValidator 控件	108
3.6.5 RegularExpressionValidator 控件	110
3.6.6 CustomValidator 控件	113
3.7 登录控件	115
3.7.1 Login 控件	115
3.7.2 LoginName 控件	117
3.7.3 LoginStatus 控件	117
3.7.4 LoginView 控件	118
3.7.5 CreateUserWizard 控件	119
3.7.6 ChangePassword 控件	119
3.7.7 PasswordRecovery 控件	120
3.8 用户控件	120

3.8.1	用户控件简介	121
3.8.2	创建用户控件	121
3.8.3	调用用户控件	121
3.8.4	用户控件属性增加	122
习题 3	126
第 4 章	网页布局	127
4.1	站点建立	127
4.1.1	IIS 的添加和运行	127
4.1.2	Web 服务器	127
4.2	母版页和内容页	132
4.2.1	母版页和内容页概述	132
4.2.2	创建母版页和内容页	133
4.3	网站导航	137
4.3.1	SiteMapPath 控件	137
4.3.2	TreeView 控件	141
4.3.3	Menu 控件	145
习题 4	147
第 5 章	基本内置对象	148
5.1	ASP.NET 对象概述	148
5.2	Page 和 Cookie 对象	149
5.2.1	Page 对象	149
5.2.2	Cookie 对象	150
5.3	Request 和 Response 对象	153
5.3.1	Request 对象	153
5.3.2	Response 对象	162
5.4	Application 和 Session 对象	166
5.4.1	Application 对象	166
5.4.2	Session 对象	169
习题 5	171
第 6 章	数据库操作	172
6.1	数据库应用开发概述	172
6.2	ADO.NET 数据库开发方式	173
6.2.1	ADO.NET 数据库开发概述	173
6.2.2	.NET Framework 数据提供程序	173
6.2.3	.NET Framework DataSet	174
6.2.4	数据库操作过程	175

6.3	Connection 对象	176
6.3.1	连接字符串的差异	177
6.3.2	数据库连接方式	177
6.3.3	应用程序中连接字符串的存储	179
6.3.4	连接池	179
6.4	Command 对象	180
6.4.1	设置连接和 SQL 命令	181
6.4.2	执行命令	181
6.5	DataReader 对象	184
6.5.1	DataReader 对象概述	184
6.5.2	从 DataReader 读取数据	185
6.5.3	DataReader 对象的使用	186
6.6	DataAdapter 对象与 DataSet 对象	187
6.6.1	ADO.NET 数据集工作原理	187
6.6.2	DataAdapter 对象	188
6.6.3	DataSet 对象	189
6.6.4	DataTable 对象	194
6.6.5	DataColumn 对象	195
6.6.6	DataRow 对象	195
6.6.7	多表操作	196
6.7	数据绑定	197
6.7.1	单值数据绑定	197
6.7.2	重复值数据绑定	199
6.8	数据控件访问数据库	201
6.8.1	数据源控件	201
6.8.2	数据展示	204
	习题 6	219
第 7 章	文件操作	220
7.1	概述	220
7.1.1	文件和流	220
7.1.2	流相关类介绍	221
7.2	文件与文件夹基本操作	221
7.2.1	Directory	222
7.2.2	DirectoryInfo	225
7.2.3	File	225
7.2.4	FileInfo	230
7.3	读写文件	230
7.3.1	StreamReader	230

7.3.2	StreamWriter	231
7.3.3	FileStream	232
7.4	文件的上传与下载	235
7.4.1	上传文件	235
7.4.2	下载文件	239
	习题 7	240
第 8 章	XML 和 Web 服务	241
8.1	XML	241
8.1.1	XML 概述	241
8.1.2	写 XML 文档	242
8.1.3	读 XML 文档	244
8.1.4	XmlDocument 类	247
8.2	Web 服务	248
8.2.1	Web Service 简介	248
8.2.2	创建 Web Service	249
8.2.3	发布 Web Service	255
8.2.4	使用 Web Service	256
	习题 8	260
第 9 章	设置与发布	261
9.1	配置 IIS 服务器	261
9.1.1	Web 服务器与 FTP 服务器	261
9.1.2	设置 IP 地址	263
9.1.3	安装 IIS	266
9.1.4	配置 IIS	266
9.2	创建 FTP 远程管理账号	266
9.2.1	安装 FTP 组件	267
9.2.2	添加 FTP 站点	267
9.3	发布网站	269
9.3.1	使用 FTP 方式发布网站	269
9.3.2	使用 HTTP 方式发布网站	271
9.3.3	使用文件系统方式发布网站	272
9.4	使用 Web 安装项目部署应用程序	273
	习题 9	273
	参考文献	274

第 1 章

ASP.NET 基础

介绍动态网页 ASP、.NET Framework 与 ASP.NET 各自的特点及其之间的关系,详细介绍 IIS 服务器的安装与配置,并通过一个 ASP.NET 实例来初步认识开发网站的过程。主要实现以下目标:

- 了解 ASP、.NET Framework 与 ASP.NET 的基本概念及其关系;
- 掌握 IIS 的安装与配置;
- 掌握 ASP.NET 简单网页的制作。

1.1 ASP.NET 概述

1.1.1 ASP 简介

ASP 是一种服务器端脚本编写环境,可以用来创建和运行动态网页或 Web 应用程序。ASP 网页可以包含 HTML 标记、普通文本、脚本命令以及 COM 组件等。利用 ASP 可以向网页中添加交互式内容(如在线表单),也可以创建使用 HTML 网页作为用户界面的 Web 应用程序。与 HTML 相比,ASP 网页具有以下特点:

- 利用 ASP 可以实现突破静态网页的一些功能限制,实现动态网页技术。
- ASP 文件是包含在 HTML 代码所组成的文件中的,易于修改和测试。
- 服务器上的 ASP 解释程序会在服务器端执行 ASP 程序,并将结果以 HTML 格式传送到客户端浏览器上,因此使用各种浏览器都可以正常浏览 ASP 所产生的网页。
- ASP 提供了一些内置对象,使用这些对象可以使服务器端脚本功能更强。例如,可以从 Web 浏览器中获取用户通过 HTML 表单提交的信息,并在脚本中对这些信息进行处理,然后向 Web 浏览器发送信息。
- ASP 可以使用服务器端 ActiveX 组件来执行各种各样的任务,例如,存取数据库、发送 E-mail 或访问文件系统等。
- 由于服务器是将 ASP 程序执行的结果以 HTML 格式传回客户端浏览器,因此使用者不会看到 ASP 所编写的原始程序代码,可防止 ASP 程序代码被窃取。
- 方便连接 Access 与 SQL 数据库。
- 开发需要有丰富的经验,否则会留出漏洞,让黑客进行注入攻击。

ASP 不仅仅局限于与 HTML 结合制作 Web 网站,还可以与 XHTML 和 WML 结合制

作 WAP 手机网站。其原理是一样的。

1.1.2 ASP 工作原理

当在 Web 站点中融入 ASP 功能后,将发生以下事情:

- (1) 用户在浏览器地址栏中输入网址,默认页面的扩展名是 asp。
- (2) 浏览器向服务器发出请求。
- (3) 服务器引擎开始运行 ASP 程序。
- (4) ASP 文件按照从上到下的顺序开始处理,执行脚本命令,执行 HTML 页面内容。
- (5) 页面信息发送到浏览器。

早期的 ASP(ASP 1.0、ASP 2.0、ASP 3.0)是 IIS 的一种开放式的无须进行编译的应用程序环境。也就是说,ASP 程序是解释执行的。IIS 是服务器上安装的 Internet 信息服务器(Internet Information Server),它是 Microsoft 公司开发的一个网络文件和应用程序服务器(即 Web 服务器),这个服务器包含在操作系统中。在 Windows 7 中,它的版本是 6.1。IIS 支持 HTTP、FTP 和 Gopher 协议。由于 ASP 是服务器端的脚本编程环境,而所有的程序都是解释执行,这意味着在这个环境中的所有程序在每次被访问的时候都需要 IIS 进行解释,从而客户端会得到一个执行结果。

ASP 请求的处理过程如图 1.1 所示。注意,由于 ASP.NET 对 ASP 请求的处理过程与早期版本的 ASP 的处理有所不同,所以在服务器端并没有详细表示出请求的处理过程。

具体的处理过程是这样的:在客户机中,有一个用于浏览网页的浏览器,用户在这个浏览器中输入 HTTP 请求。HTTP 请求通过 Internet 找到相应的 Web 服务器,并把这个请求传给这个服务器相应的处理模块。由这个处理模块负责找到相应的 ASP 程序或 ASP.NET 程序,进行相应的执行(对于 ASP 程序来说,是通过一个名叫 asp.dll 的 ISAPI DLL 进行程序的解释;而对于 ASP.NET 程序来说,则是通过 aspnet_isapi.dll 进行处理)。执行结果通过 Internet 返回给客户端,形成 HTTP 响应。

ASP 是一个服务器端脚本编写环境,用于创建动态的交互式 Web 服务器应用程序。在 ASP 动态网页中,不仅可以包含服务器端脚本代码,也可以使用 ASP 内建对象和服务器组件,此外还可以包含服务器端文件。

1. 服务器端脚本

服务器端脚本是一系列指令,用于向 Web 服务器发出命令。若要在 ASP 页面中插入服务器端脚本,首先需要设置所使用的脚本语言,有以下 3 种设置方式。

1) 使用 @ Language 指令

@ Language 指令设置用于解释脚本命令的语言。语法格式如下:

```
<% @ Language = 脚本语言 %>
```

注意: @ Language 指令必须放在文档的第一行。在 @ 符号与关键字 Language 之间要

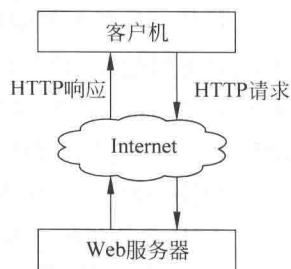


图 1.1 一个 ASP 请求的处理过程

有一个空格。

2) 使用<script> 标记的相关属性

若要在文档中包含服务器端脚本,也可以使用<script>标记的 Language 属性来设置所有的脚本语言,并使用 Runat 属性指明脚本是在服务器端运行的。例如:

```
<script language = "VBScript" runat = "server">  
:  
</script>
```

【例 1-1】 利用 ASP 动态网页显示字体的 5 种大小。

(1) 选择“开始”→“程序”→“附件”→“记事本”命令。在记事本程序中输入如下代码:

```
<% @ language = VBScript %>  
<html>  
<head>  
<title>如何设置所使用的脚本语言</title>  
</head>  
<body>  
<% for i=3 to 7 %>  
<Font color = "red" size = <% = i %>>  
欢迎来闪客启航学习!<br>  
</font>  
<% next %>  
</body>  
</html>
```

(2) 选择“文件”→“另存为”命令,保存文件名为 first. asp。

(3) 把文件 first. asp 放在 D:\test 文件夹下。

(4) 打开 IE 浏览器,在地址栏中输入 http://localhost/first. asp 后按 Enter 键,结果如图 1.2 所示。



图 1.2 简单 ASP 文件

2. ASP 的包含文件

如果要在一个页面中重复调用一段子程序或在多个页面中调用一段子程序,可以将这

一段子程序单独放在一个文件中,称其为被包含文件,然后在每个要调用这段子程序的 .asp 页中将“被包含文件”包含进来。其格式如下:

```
<!-- # INCLUDE Virtual | file = "文件名" -->
```

说明:“文件名”是被包含的文件名。Virtual 表示被包含文件的路径开始于虚拟目录。例如,如果一个被命名为 aa.inc 的文件属于一个名为 /bb 的虚拟目录,则下述命令将把 aa.inc 的内容插入到包含该命令的文件中:

```
<!-- # include virtual = "/bb/aa.inc" -->
```

表示使用相对路径,相对路径开始于含有当前文件的目录。例如,当前文件位于 bb 中,而被包含文件 aa.inc 在 bb/cc 目录中,下述命令将把 aa.inc 插入到包含该命令行的文件中:

```
<!-- # include file = "cc/aa.inc" -->
```

如果被包含文件存在于父文件所在的文件夹或其子文件夹中,可以使用 file 参数指定相对路径,也可以使用 virtual 参数指定虚拟路径;如果被包含文件存在于父文件夹所在文件夹的上层文件夹中,则只能使用 virtual 参数指定虚拟路径。

【例 1-2】 通过使用 myFont.inc 包含文件显示“HelloWord!”。此被包含文件放在调用此被包含文件的同一目录下。

(1) 选择“开始”→“程序”→“附件”→“记事本”命令。在记事本程序中输入如下代码:

```
<%  
Sub SayHello  
    response.write("HelloWord!")  
End Sub  
%>
```

(2) 选择“文件”→“另存为”命令,保存文件名为 myFont.inc。

(3) 把文件 myFont.inc 放在 D:\test 文件夹下。

(4) 选择“开始”→“程序”→“附件”→“记事本”命令。在记事本程序中输入如下代码:

```
<% @ language = "VBScript" %>  
<!-- # include file = "myFont.inc" -->  
<html>  
<title># include 命令的应用示例</title>  
<body>  
<% SayHello %>  
</body>  
</html>
```

(5) 选择“文件”→“另存为”命令,保存文件名为 second.asp。

(6) 把文件 second.asp 放在 D:\test 文件夹下。

(7) 双击桌面图标 Internet Explore (IE 浏览器),在地址栏中输入 http://localhost/second.asp 后按 Enter 键,结果如图 1.3 所示。

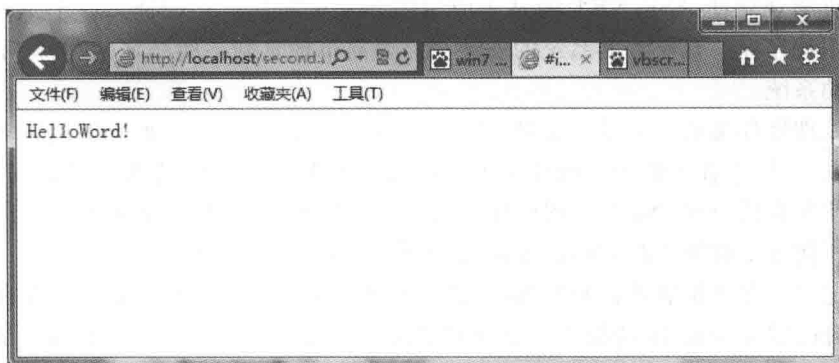


图 1.3 ASP 的包含文件

1.1.3 ASP.NET 简介

ASP.NET 常用的主要有两种开发语言,VB.NET 和 C#,C# 相对更为常用,因为 .NET 独有的语言,VB.NET 则为以前 VB 程序设计,适合于以前 VB 程序员。如果新接触 .NET,没有其他开发语言经验,建议直接学习 C#,本书就以 C# 为开发语言介绍 ASP.NET。

ASP.NET 能被用于一台 Web 服务器来建立强大的 Web 应用程序。ASP.NET 具有许多比现在的 Web 开发模式强大的优势。

- 执行效率大幅提高: ASP.NET 是把基于通用语言的程序在服务器上运行。不像以前的 ASP 即时解释程序,而是将程序在服务器端首次运行时进行编译,这样的执行效果,当然比一条一条地解释强很多。
- 世界级的工具支持: ASP.NET 构架可以用 Microsoft 公司最新的产品 Visual Studio.NET 开发环境进行开发,WYSIWYG(What You See Is What You Get,所见即为所得)编辑。这些仅是 ASP.NET 强大化软件支持的一小部分。
- 强大性和适应性: 因为 ASP.NET 是基于通用语言的编译运行的程序,所以它的强大性和适应性,可以使它运行在 Web 应用软件开发者的几乎全部的平台。通用语言的基本库、消息机制、数据接口的处理都能无缝地整合到 ASP.NET 的 Web 应用中。ASP.NET 同时也是语言独立化的,所以,可以选择一种最适合的语言来编写程序,或者把程序用很多种语言来写,现在已经支持的有 C#(C++ 和 Java 的结合体)、VB、JScript。将来,这样的多种程序语言协同工作的能力保护现在的基于 COM+ 开发的程序,能够完整地移植到 ASP.NET。
- 简单性和易学性: ASP.NET 使运行一些很平常的任务(如表单的提交、客户端的身份验证、分布系统和网站配置)变得非常简单。例如,ASP.NET 页面构架允许建立自己的用户分界面,使其不同于常见的 VB-Like 界面。
- 高效可管理性: ASP.NET 使用一种字符基础的、分级的配置系统,使服务器环境和应用程序的设置更加简单。因为配置信息都保存在简单文本中,新的设置有可能不需要启动本地的管理员工具就可以实现。这种被称为 Zero Local Administration

的哲学观念使 ASP.NET 的基于应用的开发更加具体和快捷。一个 ASP.NET 的应用程序在一台服务器系统的安装只需要简单地复制一些必需的文件,不需要重新启动系统。

- 多处理器环境的可靠性: ASP.NET 已经被刻意设计成为一种可以用于多处理器的开发工具,它在多处理器的环境下用特殊的无缝连接技术,将极大地提高运行速度。即使现在的 ASP.NET 应用软件是为一个处理器开发的,将来多处理器运行时不需要任何改变都能提高它们的效能,但现在的 ASP 却做不到这一点。
- 自定义性和可扩展性: ASP.NET 设计时考虑了让网站开发人员可以在自己的代码中自己定义 plug-in 的模块。这与原来的包含关系不同,ASP.NET 可以加入自己定义的任何组件。网站程序的开发从来没有这么简单过。
- 安全性: 基于 Windows 认证技术和应用程序配置,可以确信源程序是相当安全的。

不论是 ASP 文件还是 ASP.NET 文件都是一个可以用任何文本编辑器编辑的纯文本文件,只要将这个文件的扩展名保存为 .asp(ASP 程序)或者 .aspx(ASP.NET 程序)就可以了。现在也有很多的开发工具可以进行 ASP.NET 程序的开发,比如 Microsoft 的 Visual InterDev 等开发工具。不过也有很多程序员选择使用记事本来开发 ASP 程序或 ASP.NET 程序。当然,对于 ASP.NET 来说,还可以使用最新的 Visual Studio.NET 进行各种控件的开发。为此,必须介绍 .NET Framework。

1.2 .NET Framework

1.2.1 .NET Framework 的作用

Microsoft .NET Framework 是生成、部署和运行 Web 服务及应用程序的平台。它提供了一个生产率高且基于标准的多语言环境,用于将现有投资与下一代应用程序和服务集成,同时提供了解决 Internet 规模应用程序的部署和操作难题的灵活性。

Microsoft .NET Framework 通过一个全面托管的、受保护的和特性丰富的应用程序执行环境,把应用程序开发转换为 XML Web 服务开发,简化了开发和部署,实现了与各种编程语言的无缝集成。

以前,无论是以 VB 或 C++ 语言编写的程序,还是以其他任何可编译语言编写的程序,其各自的语言编译器都会将源代码编译成目标代码或可执行代码(也就是机器指令)。

使用 .NET 时,不同之处在于以 VB、C++ 或 .NET 所支持的任何语言编写的程序都将由其各自的编译器编译为中间语言,也称为 IL(中间语言)或 MSIL(Microsoft IL)。然后,CLR(Common Language Runtime,公共语言运行库)将执行 IL。这一点,.NET 所支持的语言和 Java 非常类似。不同的是,使用 .NET,所有支持的语言都使用同一个公共的运行库——CLR。

特定语言的编译器将 .NET 源代码编译为 MSIL 和元数据,再由 CLR 转换成机器码,然后操作系统执行此机器码。由此可以看出,.NET 程序实际上被编译了两次。由于 IL 与机器码非常接近,因此,第二次编译(从 IL 到机器码)的速度比较快。

MSIL 可用于实现不同高级语言的互操作,因为无论使用哪种 .NET 语言,都会生成

MSIL,然后由 CLR 执行此 IL。MSIL 是独立于 CPU 的指令集,这样就可以通过不同硬件上的 CLR 版本轻松实现软件移植。

IL 不是字节代码或机器码,但与其非常接近。因此,在执行应用程序时,从 IL 到机器码的转换速度非常快。MSIL 是独立于 CPU 的指令集。CLR 将这些指令转换为 CPU 特定的指令集。

.NET Framework 的作用在于:

- 提供一致的面向对象的编程环境。
- 提供将软件部署和版本控制冲突最小化的代码执行环境。
- 提供保证代码安全执行的代码执行环境。
- 提供可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序时保持一致。
- 按照工业标准生成所有通信,以确保基于 .NET Framework 的代码可与任何其他代码集成。

1.2.2 .NET Framework 的组成

.NET Framework 具有两个主要组件: CLR 和 .NET Framework 类库,如图 1.4 所示。

公共语言运行时和 Java 虚拟机一样也是一个运行时环境,它负责资源管理(内存分配和垃圾收集)。公共语言运行库,是一种多语言执行环境,支持众多的数据类型和语言特性。它管理着代码的执行,更加简单。这是一种可操控的执行环境,其功能通过编译器与其他工具共同展现。为了提高平台的可靠性,以及为了达到面向事务的电子商务应用所要求的稳定性级别,CLR 还要负责其他一些任务,比如监视程序的运行。

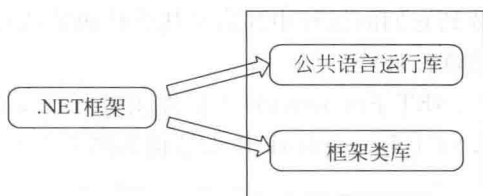


图 1.4 .NET Framework 组件

可以将公共语言运行库看作一个在执行时管理代码的代理,它提供内存管理、线程管理和远程处理等核心服务。这类似于 Java 的虚拟机。事实上,代码管理的概念是公共语言运行时的基本原则。以公共语言运行库为目标的代码称为托管代码,不以公共语言运行库为目标的代码称为非托管代码。

在 CLR 的控制下执行,受一定约束条件的支配,如编译器必须产生针对特定 CPU 的可执行代码,这样的程序叫作托管代码。使用 .NET Framework 托管代码的好处为:内存管理、混合语言的能力、更好的安全性、对版本控制的支持等。程序员可以在托管代码调用所有 .NET Framework 类库中提供的功能丰富强大的类,使得开发工作变得更轻松、软件功能更强大。不在 CLR 的控制下执行的程序,叫作非托管代码。托管代码与非托管代码的关系见图 1.5。

在 CLR 的控制下执行,受一定约束条件的支配,如编译器必须产生针对特定 CPU 的可执行代码,这样的程序叫作托管代码。使用 .NET Framework 托管代码的好处为:内存管理、混合语言的能力、更好的安全性、对版本控制的支持等。程序员可以在托管代码调用所有 .NET Framework 类库中提供的功能丰富强大的类,使得开发工作变得更轻松、软件功能更强大。不在 CLR 的控制下执行的程序,叫作非托管代码。托管代码与非托管代码的关系见图 1.5。

.NET Framework 之前的所有 Windows 程序都是非托管代码。非托管代码无法调用 .NET Framework 类库,也无法享受 .NET Framework 带来的其他所有好处。但是 .NET Framework 可以与非托管代码联合操作。.NET Framework 提供了托管代码与非托管代码互操作的服务,比如与 COM 互操作的 Interop 服务。