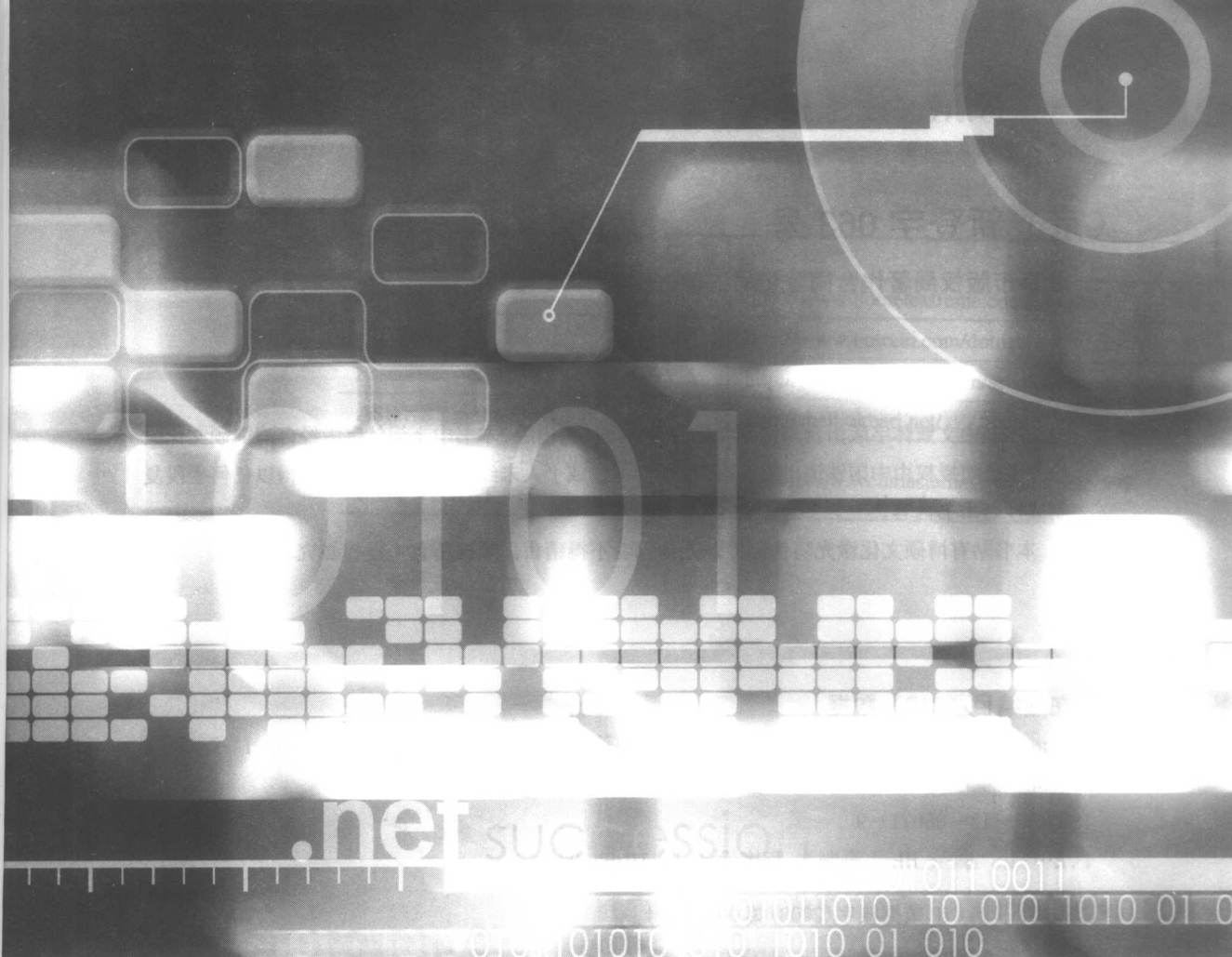




ASP.net 交互式 Web 数据库程序设计

廖信彦 © 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



ASP.net 交互式 Web 数据库 程序设计

廖信彦 编著

中国铁道出版社

2003·北京

(京)新登字 063 号

北京市版权局著作合同登记号: 01-2002-6708 号

版 权 声 明

本书中文繁体字版由台湾博硕文化股份有限公司出版。本书中文简体字版经台湾博硕文化股份有限公司授权由中国铁道出版社出版。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

本书贴有博硕文化激光防伪标签,无标签者不得销售。版权所有,侵权必究。

图书在版编目(CIP)数据

ASP.net 交互式 Web 数据库程序设计 / 廖信彦编著. —北京: 中国铁道出版社, 2003. 1

(.NET 战略丛书)

ISBN 7-113-05071-9

I. A… II. 廖… III. 主页制作-程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2003)第 003557 号

书 名: ASP.net 交互式 Web 数据库程序设计

作 者: 廖信彦

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 郭毅鹏

责任编辑: 苏 茜 刘文龙

封面设计: 孙天昭

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张: 30.5 字数: 726 千

版 本: 2003 年 2 月第 1 版 2003 年 2 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05071-9/TP·865

定 价: 48.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

导 读

本书各章节的范例与案例都使用 Access 2000/2002(XP)当成后台数据源,但是所有的数据表可百分之百移植到后台数据库(如:SQL Server、Oracle 等)而无问题,惟一的差别仅在于数据库的连接字符串,此于各章节的 ASP.NET 文件内都有简短的代码解说。

为了能够更正确地执行本书各章节的案例/范例,建议参考下列四项重点:开发环境、光盘范例的安装、创建虚拟目录以及默认的主页位置等,一一列如下。

开发环境

本书所有范例的开发平台有两种,分别是

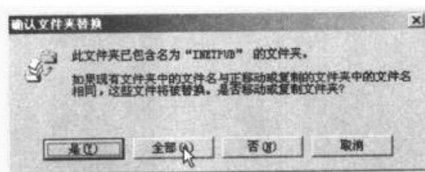
- Windows 2000 Advanced Server + IIS 5.0 + .NET Framework (中文版) + SQL Server 7.0/2000
- Windows 2000 Professional + IIS 5.0 + .NET Framework (中文版) + Access 2000/2002

任一章节的范例在上述平台下都能正确地执行。

复制光盘范例到硬盘

安装 IIS 5 时,默认的文件夹是 C:\inetpub,而本书的范例位于 C:\inetpub\asp_net,因而需将光盘中的对应目录复制到硬盘,步骤如下:

- 打开“资源管理器”,点选光驱图标后,找到 inetpub 文件夹。
- 拖动 inetpub 文件夹到 C(或其他)磁盘驱动器图标,然后放开鼠标键,此刻系统开始复制,倘若发现有同名的文件夹,将会弹出图导-1 的对话框。



图导-1 “确认文件夹替换”对话框

- 单击“全部”按钮直接覆盖。

最后可参考光盘根目录下 Disk.txt 的目录结构,以便和 C:\inetpub\asp_net 文件夹相互对比。

创建虚拟目录

请操作 IIS 的管理工具新建四个虚拟目录,分别是:

- Asp_Net: 对应于 C:\inetpub\asp_net 文件夹。

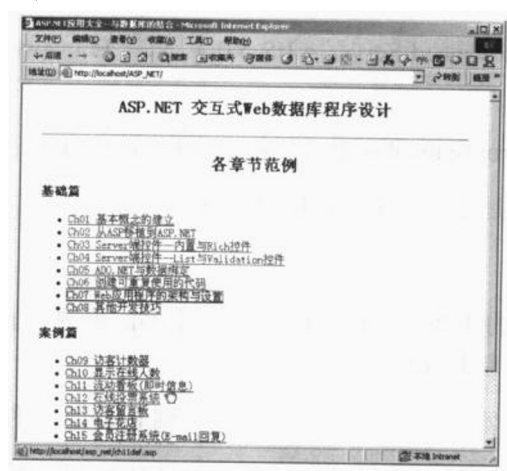
- NewGlobal: 对应于 C:\inetpub\asp_net\ch07 文件夹。
- Login: 对应于 C:\inetpub\asp_net\ch07\Login 文件夹。
- AspChat: 对应于 C:\inetpub\asp_net\ch17 文件夹。

如果不知道或不熟悉如何创建虚拟目录的话, 可参考 7-1-3 节的说明。

链接主页位置

安装 Web Server(IIS 5)、复制光盘范例文件、创建虚拟目录之后, 便可准备链接到本书默认的主页。请打开浏览器, 在地址栏输入 http://localhost/asp_net, 倘若一切正常的话, 应该会看到图导-2 所示的主页画面。

接下来请按下每一章的超级链接, 以便进一步执行其下的各个范例。



图导-2 链接本书主页

目 录

第 1 章 基本概念的建立.....	1
1-1 ASP.NET 介绍	2
1-1-1 动态网页技术的演进.....	2
1-1-2 下一代的 ASP——ASP.NET	5
1-2 全新的运行环境	8
1-2-1 .NET Framework	8
1-2-2 Web 应用程序的架构	10
1-3 ASP.NET 新功能列表	10
1-4 安装 ASP.NET	16
1-4-1 系统需求	16
1-4-2 安装信息	17
1-5 本章总结	19
第 2 章 从 ASP 移植到 ASP.NET	21
2-1 网页架构的改变	22
2-1-1 Page 前导指令	22
2-1-2 使用编译语言	24
2-1-3 ASP.NET 网页的编辑习惯	24
2-1-4 Namespaces 的使用	29
2-2 Server Controls 的使用	29
2-3 程序语言 (VB) 的改变	33
2-3-1 完整的数据类型	34
2-3-2 变量的声明方式	34
2-3-3 不支持 Set/Let 语句	35
2-3-4 类属性的定义	35
2-3-5 结构化的错误处理	36
2-3-6 其他	36
2-4 其他注意事项	39
2-4-1 Late Binding 和 Early Binding	39
2-4-2 Managed/Unmanaged 组件	39
2-5 本章总结	40

ASP.net交互式Web数据库 程序设计

第3章 Server端控件——内置与 Rich 控件	41
3-1 Web 控件介绍	42
3-1-1 HTML 控件	42
3-1-2 ASP.NET 控件	44
3-1-3 两类控件的对照	47
3-1-4 控件家族	49
3-2 内置控件	49
3-2-1 文本输入	50
3-2-2 单选与复选钮	51
3-2-3 显示列表框与下拉列表框	54
3-2-4 其他	57
3-3 Rich 控件	58
3-3-1 AdRotator 控件	58
3-3-2 Calendar 控件	61
3-4 本章总结	68
第4章 Server端控件——List 与 Validation 控件	69
4-1 List 控件	70
4-1-1 DataGrid	70
4-1-2 Repeater	75
4-1-3 DataList	77
4-2 Validation 控件	79
4-2-1 RequiredFieldValidator	80
4-2-2 RangeValidator	83
4-2-3 CompareValidator	85
4-2-4 RegularExpression Validator	87
4-2-5 CustomValidator	90
4-2-6 ValidationSummary 控件	93
4-3 本章总结	99
第5章 ADO.NET 与数据绑定	101
5-1 ADO.NET 介绍	102
5-1-1 使用旧的 ADO	102
5-1-2 和 ADO 的差异	104
5-1-3 使用 Namespaces	105
5-2 ADO.NET 对象模型	106
5-2-1 DataReaders	107

5-2-2	DataSets	109
5-3	数据绑定 (Data Binding)	113
5-3-1	绑定数组	113
5-3-2	绑定 XML 数据	115
5-3-3	连接数据库	118
5-3-4	和 DataGrid 控件的集成	120
5-4	数据处理	124
5-4-1	查询数据	124
5-4-2	插入数据	127
5-4-3	修改数据	131
5-4-4	删除数据	134
5-5	本章总结	138
第 6 章	创建可重复使用的代码	139
6-1	Pagelet	140
6-1-1	Pagelet 介绍	140
6-1-2	创建第一个 Pagelet	142
6-1-3	使用参数化 Pagelet	145
6-2	Code Behind 窗体	148
6-2-1	Code Behind 窗体介绍	149
6-2-2	创建并使用 Code Behind 窗体	149
6-3	自定义 ASP.NET 控件	154
6-3-1	自定义控件介绍	155
6-3-2	创建并使用 ASP.NET 控件	155
6-3-3	分发 ASP.NET 组件	158
6-4	本章总结	158
第 7 章	Web 应用程序的架构与设置	159
7-1	Web 应用程序的设置	160
7-1-1	Application/Session 对象的角色	160
7-1-2	Global.asax 的架构	163
7-1-3	Event Handler 的执行顺序	168
7-2	Web 应用程序的配置	172
7-2-1	Web.config	172
7-2-2	配置节的设置	173
7-3	安全性 (Security)	176
7-3-1	认证 (Authentication)	176
7-3-2	授权 (Authorization)	182

ASP.net交互式Web数据库 程序设计

7-4 本章总结	183
第 8 章 其他开发技巧	185
8-1 Web Services	186
8-1-1 Web Services 简介	186
8-1-2 创建 Web Services	187
8-1-3 描述 Web Services	192
8-2 Cache	193
8-2-1 Output Cache	194
8-2-2 Data Cache	195
8-3 错误处理、跟踪及调试	199
8-3-1 错误处理	199
8-3-2 跟踪	203
8-3-3 调试	206
8-4 Cookies 的处理	207
8-4-1 ASP 3.0 的处理机制	207
8-4-2 ASP.NET 的处理机制	208
8-5 访问文件系统	210
8-5-1 写入文件	211
8-5-2 读取文件	214
8-6 发送 E-mail	215
8-7 文件上传	216
8-8 本章总结	218
第 9 章 访客计数器	219
9-1 文件架构	220
9-2 设计前的思考	221
9-3 加强版图形模式访客计数器	223
9-4 终极版图形模式访客计数器	227
9-5 本章总结	230
第 10 章 显示在线人数	231
10-1 文件架构	232
10-2 设计前的思考	233
10-3 到底有几人在线	234
10-4 与计数器结合	239
10-5 本章总结	242



第 11 章 流动看板（即时信息）	243
11-1 文件架构	244
11-2 设计前的思考	245
11-3 设计流动看板	246
11-4 与在线人数结合	250
11-5 即时股价信息	252
11-6 本章总结	259
第 12 章 在线投票系统	261
12-1 文件架构	262
12-2 设计前的思考	263
12-3 文本模式的投票结果	264
12-4 模块化的文本模式投票结果	271
12-5 图形模式的投票结果	273
12-6 本章总结	278
第 13 章 访客留言板	279
13-1 文件架构	280
13-2 设计前的思考	281
13-3 基本型留言板	282
13-4 高级型留言板	290
13-5 本章总结	300
第 14 章 电子花店	301
14-1 文件架构	302
14-2 设计前的思考	303
14-3 设计订购窗体	304
14-4 第二份确认窗体	309
14-5 写入订单数据表	318
14-6 本章总结	324
第 15 章 会员注册系统	325
15-1 文件架构	326
15-2 设计前的思考	327
15-3 第一次登录	328
15-4 核验 ID	331
15-5 电子邮件回复	335

ASP.net交互式Web数据库 程序设计

15-6 取消登录	343
15-7 本章总结	347
第 16 章 网上书店（购物袋）	349
16-1 文件架构	350
16-2 设计前的思考	351
16-3 设计注册窗体	352
16-4 设计查询窗体	355
16-5 购物袋的设计	365
16-6 更新订购量	368
16-7 取消订单与结帐	370
16-8 本章总结	379
第 17 章 网络聊天室	381
17-1 文件架构	382
17-2 设计前的思考	383
17-3 前置操作	385
17-4 聊天室的设计	394
17-5 查询窗口	405
17-6 本章总结	410
第 18 章 统计图表	411
18-1 文件架构	412
18-2 设计前的思考	413
18-3 公司运营状况	414
18-4 报表产出时间	420
18-5 股票 K 线图	432
18-6 本章总结	439
附录 A ASP.NET 的对象模型	441
A-1 Application 对象	442
Application 对象的属性	442
Application 对象的方法	443
A-2 ASPError 对象	444
ASPError 对象的属性	445
A-3 HttpContext 对象	446
HttpContext 对象的属性	446
HttpContext 对象的方法	448



A-4	Request 对象	449
	Request 对象的属性	449
	Request 对象的方法	454
A-5	Response 对象	454
	Response 对象的属性	455
	Response 对象的方法	458
A-6	Server 对象	460
	Server 对象的属性	460
	Server 对象的方法	460
A-7	Session 对象	462
	Session 对象的属性	462
	Session 对象的方法	464
附录 B	Internet 上的 ASP.NET 资源	467



1

基本概念的建立

本章摘要

- ASP.NET 介绍
- 全新的运行环境
- ASP.NET 新功能列表
- 安装 ASP.NET
- 本章总结

ASP.net交互式Web数据库 程序设计

ASP (Active Server Pages) 是近年来相当流行的动态网页开发技术, 从 1996 年首度面市以来, 至今已过了六个年头, 版本编号也从 1.0 迅速扩展到 3.0。

虽然历经了三代, 不过 ASP 2.0 到 ASP 3.0 之间着实改变不大 (嗯, 至少 Programming 的方式变异不多), 顶多加了一个新的对象 (ASPErrors), 若干种方法; 若真要说有什么变化的话, 反倒是 IIS 5 本身针对 Scripting 语言与 ADO (ActiveX Data Object) 进行了最佳化, 所以能在不更改原有代码的情况 (绝大部分 Web 开发者的期望) 下, 提升整体的性能。

好了, 快乐的日子过没多久, Windows 2000/IIS 5.0/ASP 3.0 推出不过一年多, 马上又有个叫 .NET 架构的家伙出现。

简单地讲, 由于 .NET 架构改变了操作系统与应用程序之间沟通的机制, 所以以往用 VB、ASP 或其他程序语言开发的应用系统, 为了能在新的 .NET 架构上发挥更好的性能, 或多或少都要稍微改写一番 (嗯, 至少也要重新编译、链接)。而 .NET 架构受 ASP 的影响, 可说是进行了一次大革命, 只有内置的对象继承 ASP 3.0, 其他几乎是概念整个大翻新, 相当于重新学习一套新的开发方式。

听起来相当惊人, 不过事实便是如此, 至于真相到底如何呢? 请读过本章后再行评论, 下文即进入全新的 ASP.NET 之旅。

1-1 ASP.NET 介绍

ASP.NET 先前称为 ASP+, 早在 1997 年年底由开发 IIS (Internet Information Server) 的一组人员提出, 对于 ASP 优点、缺点的不断争辩造就了 ASP+ 的诞生, 其前提是让 ASP 的开发工作变得更不“复杂”, 而那时候 ASP 才刚过一岁生日。

其后当然是一连串的开发奋斗史, 例如帮 ASP 加上一直缺乏的编译语言 (Compiled Language) 环境、组件基础及面向对象等模型, 并且搭配适当的开发工具 (如: Visual Studio.NET, 先前则是使用 Visual InterDev) 等。至此为止, 开发 ASP.NET 网页可说是再简单不过。

在深入了解 ASP.NET 的内涵前, 文中再复习一次 Web 开发技术的沿革, 相信在读完本节之后, 大部分的开发者定然会认为 ASP.NET 的出现真是再自然不过了。

1-1-1 动态网页技术的演进

就经验得知, 一般的网页 (HTML 文件) 是静态的, 纯粹展示某些特定的内容, 通常也很少更改。相对于 HTML 文件, 如果可和上网者进行一些交互, 例如: 访客留言、网页搜索等, 我们便称这种网页为“动态”网页。

常用来开发动态网页的技术, 从以往熟知的 CGI、WinCGI, 至现今流行的 ASP、ASP.NET、PHP、JSP 等, 均可作为当时的代表, 下面小节便一一揭示。

CGI

CGI 是 Common Gateway Interface 的缩写, 中文名为“通用网关接口”, 它是位于 Web 服务器与 Web 应用程序之间的接口, 后者通过此接口与前台浏览者产生交互。由于 CGI 是

第 1 章 基本概念的建立

标准化的 Web 服务器接口规格，它和服务端的应用程序（script 或可执行文件）有着密不可分的关系，所以大部分人就将 CGI 与 Server 端的 script（或可执行文件）合而为一，直接称之为 CGI 程序，而非 CGI 接口。

基于 CGI 和 Web Server 密不可分的关系，能拿来编写 CGI 的语言也就和操作系统与 Web 服务器极为相关，Unix 平台下有 Shell Script、C、Perl 等，而 Windows 平台则有 Batch File、VB、VC/VC++、Delphi、PerlScript 等，种类可说是相当广泛。

无论采用何种平台、Web 服务器或程序语言，CGI 最大的用途便是和前台浏览者产生交互，再根据其需求进行下一道操作，如打开文件系统或访问后台数据库等。

最明显的例子便是搜索引擎，用户于浏览器输入查询条件或关键字（如：CGI）之后，单击“Submit”按钮发送请求到 Web 服务器，接着 Web 服务器便调用其上的 CGI 程序分析请求，再到数据库或文件捕获符合条件（含 CGI 字符串）的记录，最后将结果以标准 HTML 格式的文件返回给前台、显示于浏览器上方，流程如图 1-1 所示。

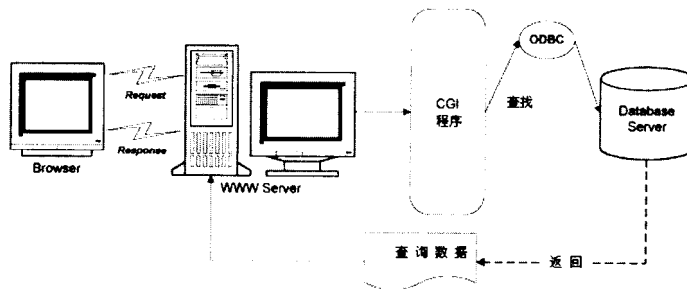


图 1-1 CGI 的运行模式

ISAPI

除了标准的 CGI 之外，微软也在他们的 Web 服务器（IIS）上引入了另一种服务器端接口，称为 Internet Server Application Programming Interface（ISAPI），它其实就是一组位于网站服务器上的 API。这些 API 集成在网站服务器中，除了 ISAPI 之外，Netscape 也在他们的 Enterprise Server 创建了一组 API，可于其上执行服务器端的 JavaScript，称为 NSAPI。如图 1-2 所示。

CGI 和 ISAPI/NSAPI 的差异，如图 1-2 所示。

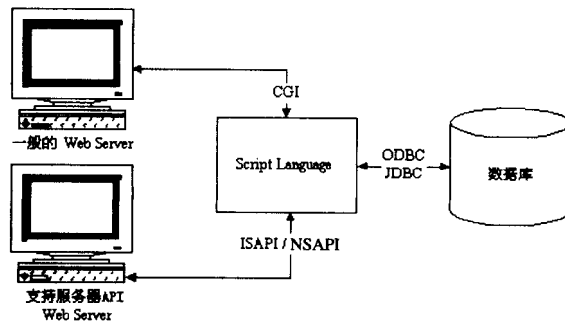


图 1-2 CGI 和 ISAPI/NSAPI 的差别

ASP.net交互式Web数据库 程序设计

ISAPI 文件几乎都是一个一个的动态链接库 (DLL)，它和 CGI 程序 (script 或执行文件) 最大的不同在于前者 (DLL 文件) 内含编译过的代码，可由 Web 服务器直接调用、运行。相对于 CGI 接口，我们可将这些 DLL 文件称为 ISAPI 接口。

从 ISAPI 接口到 ASP 1.0 诞生前还有个过渡产品，名为 IDC (Internet Database Connector)，它引入模板 (Template) 的概念，让 Web 设计者能快速地套用现成的 HTML 网页创建 Web 应用程序，说来现在的 ASP.NET 网页倒有一部分的概念派生于此，届时再加以介绍。

Index Server 2.0 的查询接口有些是以 ASP + IDC + Template (.htx) 来设计，有兴趣的朋友不妨在 NT 4.0/IIS 4/Index Server 2.0 的环境下链接到以下网址。

<http://localhost/iissamples/iissamples/default.htm>

ASP 1.0/2.0/3.0

CGI 依存着 Web 服务器而生，在挑选 Web 服务器之后，连带也决定了用来设计 CGI 的工具。Unix/Linux 平台下的 Web Server 大多采用 Apache，所以 Perl、PHP 是一时之选；Windows 平台下常见的 Web Server 有 IIS/PWS、WebSite、Apache 等，因而 VB、VC、Delphi、PerlScript、IDC、ISAPI 和 ASP 也是不错的选择。

由此推之，ASP 活跃的舞台其实是 IIS (Internet Information Server/Service) /PWS (还有其他 Web 服务器支持 ASP，例如：WebSite 2.x Pro)，所以在执行 ASP 程序时，一定要先安装 IIS 或 PWS (Personal Web Server)。

ASP 1.0 刚推出时是 IIS 3.0 的一组 add-on，需在安装 IIS 3.0 之后额外再加装 aspc.exe。以 ASP 开发的 Web 应用程序在分析前台用户的请求，或者响应结果给前台浏览器时，都是借助 asp.dll，实际上该文件便是一 ISAPI 接口程序，角色即为解译引擎。

ASP 2.0 内附于 IIS 4/PWS 4 (无须额外安装)，ASP 3.0 内附于 IIS 5，版本虽然一直升级，但背后的解译操作还是依赖 asp.dll。而 ASP 1.0 至 2.0 最大的不同，在于后者引入 Web 应用程序的概念与构建，并且允许在某个 Web 应用程序 (各自独立的内存空间) 中启动外部组件，同时加入了事务 (Transaction) 的机制。

ASP 2.0 到 3.0 最大的差别，在于后者将 MTS (Microsoft Transaction Server) 与 COM 的核心集成在一起，称为 COM+；其次便是允许在独立的进程 (Process) 内执行外部组件，借以和 Web 应用程序分隔开，避免因启动或执行组件失败时，导致原有的 Web 应用程序当掉。

总括而言，ASP 和 CGI 的情况类似，它不专指某种程序语言，而是和程序语言无关的脚本语言 (Scripting Language) 运行环境，只要 Web Server (ActiveX Scripting Host) 安装了适当的解译引擎 (ActiveX Scripting Engine)，Host 便能调用 Engine 执行 script。

安装 IIS/PWS 后，便拥有了两种默认的 Scripting Engine，其为 VBScript 与 JScript。

一般人老是以为 ASP 即是 VBScript 或 JavaScript，其实概念上有某种程度的混淆，例如 ActiveState 公司的 PerlScript 就可让 Web Server (IIS) 执行以 Perl 编写的 ASP 程序。

ASP 既不是某种特定的程序语言，也非单纯的 HTML 文件，严格说来，它倒像是两者的结合，所以在称谓上众说纷云、莫衷一是，较常见或约定成俗的名称是 ASP 程序或 ASP 文件，后续的文章中将以这两项名词为代表。



CGI 呈现的交互效果有目共睹,但慢慢地大家也发现 CGI 本身隐藏了若干缺点,例如:多人调用时效率过低、系统资源损耗过快、CGI 程序不易开发、维护成本过高、较不容易学习、不易调试等,当然,破坏总比建设容易,其中几点虽有些言过其实,但也不乏真正切中要害者。

CGI 最为人诟病者就是效率不高,如果每一份窗体都会执行一支 CGI 程序,多人同时使用时便会打开同样的可执行文件/script 若干次,如此一来将会耗费 Web Server 上过多的资源,导致效率严重降低。其次是 CGI 程序不易调试的情况,目前为止尚无一套完整、好用的调试工具,所以开发 CGI 程序时没问题最好,有问题还真让人搔破脑袋也想不出……

因此,ASP 便是微软欲于 IIS 上取代 CGI 的利器,冀望能在 Windows 2000/NT + IIS 4.0/5.0 之平台消除 CGI 的缺陷,同时又能让 Web 开发者于服务器端开发动态、交互性、高效率的应用程序,最重要的是:学习简单、容易调试及维护。

图 1-3 是 ASP 的运行模式,可和图 1-1 做个比较。

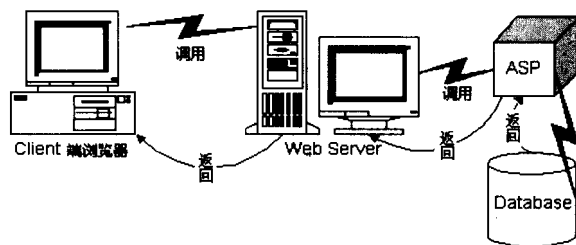


图 1-3 ASP 的运行模式

ASP.NET

ASP.NET 的原始构想源自于 1997 年年底,目的是为了了解决开发 ASP 程序时过于“复杂”、“繁琐”与“重复”等现象。就像 ASP 2.0 至 3.0 集成 MTS 与 COM runtime (运行环境) 为 COM+ 一般,开发 ASP.NET 网页,甚至是传统的 Windows 应用程序、Web Services 与 Windows Services 时,往后都是建构于新增的一组架构 (Framework) 上,名为 .NET Framework,使用时也和前一版 (3.0) 不太一样。

虽然名称都有“ASP”三个字,但我们可以把 ASP.NET 作为一全新的开发环境,从前有开发 ASP 网页的经验与投资,部分尚有用处;但若完全发挥 ASP.NET 网页极致的话,那么说要从头学起也不为过,下面我们就以一个小节的内容来介绍这种动态网页开发技术。

1-1-2 下一代的 ASP——ASP.NET

从外观看来,ASP.NET 网页和 ASP 网页长得几乎一样,同样有 <% ... %> 节段、<script>... </script> 标记,相同并强化的对象模型 (Request、Response、Server、Application、Session、HttpContext 及 ASPError 对象) 等;但由于前者根基于前文所提的 .NET 架构上,所以采用了和前一版本 (3.0) 不一样的机制,例如:舍弃 scripting 语言 (如: VBScript、JavaScript) 采用编译式语言 (如: VB、C#),加入 Server 端控件 (Control) 取代 HTML 元素等,下面是个简单的比较。

