

动态网站设计捷径

——ASP

林风 李国群 范利 等著



Internet



清华大学出版社

http://www.tsp.com

动态网站设计捷径

——ASP

林 风 李维章 赵 莉 编著

西安电子科技大学出版社

2000

内 容 简 介

ASP 是基于 Windows NT 4.0 和 Microsoft IIS 上的开放式脚本开发环境, 它很好地把 HTML 和脚本开发融合在一起, 集简单性、高效性和易扩展性于一身。由于脚本在服务器端执行, 因此它几乎可以支持任何一种浏览器。较 CGI 编程, 使用 ASP 和 ADO 组件可以更容易地在 Internet 或 Intranet 上建立整合型的 Web 数据库应用平台。

本书从 Internet 的一般概念谈起, 阐述了 ASP 应用程序使用的开发工具和平台, 介绍了 ASP 开发使用的 HTML、VBScript 语言, 以及 ASP 内建组件的使用方法, 同时详细讲解了 ASP 的数据库组件 ADO 的使用方法, 最后通过实例说明 ASP 服务器组件的编写方法。使用本书的一般读者可以从易到难, 逐步掌握 ASP 程序设计方法。对于水平较高的读者则可以将此书作为 ASP 编程时的参考手册。

图书在版编目(CIP) 数据

动态网站设计捷径: ASP/林风等编著. —西安:西安电子科技大学出版社, 1999.11

ISBN 7 - 5606 - 0795 - 0

. 动... . 林... . 因特网 - 主页 - 设计 - 应用程序, ASP . TP393.4

中国版本图书馆 CIP 数据核字(1999) 第 63433 号

责任编辑 臧延新 钟宏萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 E-mail:xdupfb@pub.xaonline.com

经 销 新华书店

印 刷 陕西画报社印刷厂

版 次 1999 年 11 月第 1 版 2000 年 3 月第 2 次印刷

开 本 787 毫米 × 1092 毫米 1/16 印张 19.75

字 数 465 千字

印 数 4001 ~ 8000 册

定 价 26.00 元

ISBN 7 - 5606 - 0795 - 0/TP · 0405

如有印制问题可调换

////////////////////////////////////

前 言

欢迎进入动态 Web 网站编程的世界。ASP(Active Server Pages) 是目前非常流行的开放式的 Web 服务器应用程序开发技术, 它将脚本、超文本和强大的数据库访问功能结合在一起, 并提供了众多的服务器组件用以创建分布式的和基于 Web 的商业化应用程序。由于 ASP 提供如此简单的方式来建立数据驱动的 Web 驱动程序, 它将取代 CGI(公共网关接口) 和繁琐的 Perl 编程语言, 成为今后 Web 开发的流行语言。随着数百个第三方厂商在 ASP 开发方面的介入, ASP 目前已可运行于绝大部分流行的 Web 服务器上。

随着 Internet 的迅速发展, 不管是专业的 ISP(Internet 服务提供者) 和 ICP(Internet 内容提供者), 还是一般的政府机关、银行、交通部门、学校、医院、服务业, 甚至是每个人, 都积极寻求在 Internet 上发布信息, 提供新型的网上管理和服务。可以这样说, 网页设计和网站的编辑正成为一种新兴的热门行业, 而使用 ASP 设计这些应用程序则是您最佳的选择。

一般的网站往往是静态的网站。这就是说, 新增、修改和删除网页都需要手动去完成, 能不能有更好的办法呢? 答案就是使用数据库和与其相关的管理应用程序。本书的目的就是为了让读者掌握 ASP 和 ADO 的网站编程技术, 熟练地使用 ASP 编写 Web 服务器程序, 动态地处理来自浏览器的请求, 从而帮助读者创建完全个人化的动态 Web 网站。

本书从 Internet 的一般概念谈起, 首先介绍了基于 ASP 的应用程序所需使用的开发工具和平台; 其次对 HTML 网页编写技术做了介绍; 为了使读者对 ASP 编程打下良好的基础, 接下来对 ASP 编程所使用的 VBScript 语言进行了讲解; 进入 ASP 编程技术, 本书详细地说明了 ASP 服务器提供的 Response、Request、Session、Application 和 Server 组件的属性和方法, 同时以大量的篇幅描述了 ASP 的数据库访问组件 ADO 的使用方法; 最后, 为想进一步提高 Web 编程技术的读者提供了一个 ASP 组件编程的实例。本书除了对 ASP 技术进行介绍以外, 也对 CGI、ISAPI 和 IDC 编程做了简单介绍。

通过对本书的学习, 可以使初学者循序渐进, 快速掌握 ASP 编程技术。对于水平较高的读者, 则可以将此书作为 ASP 编程时的实用参考手册。

由于作者水平有限, 不足之处在所难免, 欢迎广大读者批评指正。

编著者

1999 年 7 月

目 录

第 1 章 Web 程序开发简介	1
1.1 Web 的起源.....	1
1.2 超文本标记语言(HTML).....	2
1.3 超文本传输协议(HTTP).....	3
1.4 公共网关接口(CGI).....	3
1.4.1 什么是 CGI 程序.....	3
1.4.2 CGI 程序的输入.....	5
1.5 Internet 服务器应用程序接口(ISAPI).....	6
1.5.1 从 CGI 到 ISAPI.....	6
1.5.2 ISAPI 扩展动态链接库的入口函数.....	7
1.5.3 ISAPI 程序的输入：Extension Control Block(ECB).....	7
1.5.4 ISAPI 过滤器.....	9
1.6 Active Sever Pages(ASP).....	10
第 2 章 开发工具及开发平台	12
2.1 ASP 浏览器：IE(Internet Explorer).....	12
2.2 ASP Web 服务器 IIS.....	13
2.2.1 IIS 的概念.....	13
2.2.2 IIS 服务器的特性.....	14
2.2.3 IIS 体系架构.....	15
2.2.4 IIS 提供的服务.....	15
2.2.5 IIS 的软件要求.....	16
2.2.6 IIS 的使用.....	16
2.3 ASP 网页设计工具 FrontPage 98.....	17
2.3.1 FrontPage 服务器扩展(FrontPage Server Extensions).....	19
2.3.2 目录结构简介.....	20
2.3.3 控制对网站的访问.....	22
2.4 ASP 网站开发工具 Visual InerDev 6.0.....	23
2.4.1 Visual InterDev 的概念.....	23
2.4.2 建立 Web 站点.....	25
2.4.3 用 Visual InterDev 规划 Web 站点.....	27
2.4.4 新增一个 Web 页面.....	28
2.4.5 连接到数据及创建可重用的 Data Command.....	31
2.4.6 使用数据绑定控件显示数据.....	31
2.4.7 在 Visual InterDev 中对服务器端及客户端脚本除错.....	32
2.4.8 使用 Visual InterDev 进行小组开发.....	33
2.4.9 创建及修改数据库对象.....	34

2.4.10 系统要求.....	35
2.5 ASP 数据库服务器 SQL Server.....	36
第 3 章 HTML 简介	37
3.1 HTML 的概念.....	37
3.2 HTML 标记使用说明.....	39
3.2.1 Head 中的标记.....	39
3.2.2 文本标记.....	40
3.2.3 超链接(Hyperlink and Anchor)	44
3.2.4 定义列表(Definition List)	45
3.2.5 列表标记.....	45
3.2.6 图片标记.....	47
3.3 Table 表格.....	48
3.4 HTML 表单(Forms)	48
3.4.1 表单的概念.....	48
3.4.2 Form 标记.....	49
3.4.3 输入域(Input Field) : INPUT	49
3.4.4 文本框 : TEXTAREA.....	51
3.4.5 下拉框 : SELECT.....	52
3.4.6 选项 : OPTION.....	52
3.4.7 表单提交(Form Submission)	52
第 4 章 VBScript 编程语言	54
4.1 VBScript 的概念.....	54
4.2 VBScript 的数据类型.....	55
4.3 VBScript 的变量.....	56
4.3.1 变量的概念.....	56
4.3.2 变量声明.....	56
4.3.3 命名规则.....	56
4.3.4 变量的范围及生命周期.....	57
4.3.5 给变量赋值.....	57
4.3.6 数组.....	57
4.4 VBScript 的常量.....	58
4.5 VBScript 的运算符.....	59
4.6 VBScript 过程和函数.....	60
4.6.1 过程(Sub)	60
4.6.2 函数(Function)	60
4.7 程序流程.....	61
4.7.1 If...Then...Else 结构.....	61
4.7.2 Select...Case 结构.....	62
4.7.3 Do...Loop 和 For...Next 结构.....	63

4.8	VBScript 对象的使用	66
4.9	VBScript 程序模式	67
第 5 章 ASP 概述		69
5.1	ASP 程序的运行方式	69
5.2	设定 ASP 执行环境	70
5.3	ASP 文件	70
5.4	ASP 的执行和发布	72
5.5	ASP 提供的内建对象	73
5.6	ASP 服务器组件	74
5.7	包含一个文件至 ASP 文件中	75
5.8	使用 ASA 文件	76
5.9	ASP 新增功能	76
第 6 章 向客户端输出数据：Response		78
6.1	Response 功能概述	78
6.1.1	输出信息	78
6.1.2	重定向网址	80
6.1.3	由数据库向客户端输出图形	82
6.2	Response 对象的属性	84
6.2.1	Buffer 属性	84
6.2.2	设置 MIME 类型：ContentType	85
6.2.3	设置过期时间：Expires	85
6.2.4	设置时间间隔时间：ExpiresAbsolute	85
6.2.5	返回状态：Status	86
6.2.6	选择字符集：Charset	86
6.3	Response 对象的方法	87
6.3.1	写文本数据至客户端：Write	87
6.3.2	写二进制数据至客户端：WriteBinary	87
6.3.3	重定向一个网站：Redirect	87
6.3.4	结束一个网页：End	88
6.3.5	附加登录：AppendToLog	88
6.3.6	清除缓冲区：Clear	88
6.3.7	发送缓冲区：Flush	89
6.4	向客户端写入 Cookie	89
第 7 章 得到客户端的数据：Request		92
7.1	Request 对象概述	92
7.2	使用 QueryString 得到 GET 方法的数据	92
7.3	使用 Form 集合得到 POST 方法的数据	94
7.4	得到服务器变量：ServerVariables	100
7.5	得到客户端的 Cookie	103
7.6	限制用户访问 Web 页面	104

第 8 章 保存网页信息：Session	107
8.1 Session 对象	107
8.1.1 使用 Session 传递参数	107
8.1.2 Session 和 Cookie 的区别	108
8.1.3 浏览器不支持 Cookie 使用 Session	109
8.1.4 Session 的过期	109
8.1.5 Session 变量的优缺点	109
8.1.6 使用@ENABLESESSIONSTATE	110
8.2 Contents 集合	110
8.3 唯一识别符(SessionID)	110
8.4 过期时间(TimeOut)	111
8.5 释放 Session(Abandon)	111
8.6 Session 对象的事件	112
8.6.1 Session_OnStart 事件	112
8.6.2 Session_OnEnd 事件	113
8.7 使用 Session 的实例	113
8.7.1 提交用户的名称及密码	113
8.7.2 展示 Session	114
8.7.3 设置 Session	116
第 9 章 保存应用程序的信息： Application	118
9.1 Application 对象	118
9.2 设置和取得 Application 对象	118
9.3 Application 对象的方法	119
9.3.1 Lock 方法	119
9.3.2 Unlock 方法	120
9.4 Application 对象的事件	120
9.4.1 Application_OnStart 事件	121
9.4.2 Application_OnEnd 事件	121
9.5 正确使用 Application 对象	121
9.6 Application 对象使用实例	123
9.6.1 输入用户信息	123
9.6.2 设置 Application 变量	124
9.6.3 使用 Application 变量	125
9.7 计算在线用户	126
第 10 章 服务器对象： Server	128
10.1 Server 对象的属性	128
10.2 Server 对象的方法	128
10.2.1 MapPath 方法	129
10.2.2 CreateObject 方法	130
10.2.3 HTMLEncode 方法	132

10.2.4	URLEncode 方法	132
第 11 章	ASP 服务器组件	134
11.1	广告翻转器组件(AD Rotator)	134
11.2	浏览器能力(Browser Capabilities)	136
11.3	数据库访问组件(ADO)	138
11.4	文件存取组件(File Access)	139
11.4.1	FileSystemObject 对象的方法说明.....	142
11.4.2	TextStream 对象的方法说明.....	145
11.5	使用 Content Linking 组件.....	146
11.6	用 ASP 发送 Email	149
第 12 章	Web 数据库程序设计初步	152
12.1	数据库化的网站.....	152
12.2	Web Assistant	153
12.3	使用 IDC 访问数据库.....	155
12.3.1	IDC 的优点.....	156
12.3.2	建立数据来源(DSN)	156
12.3.3	输入数据的 Form 表单	159
12.3.4	IDC 文件中的内容.....	160
12.3.5	HTX 模板文件.....	160
12.4	使用 ADO 访问数据库.....	163
12.4.1	ADO 的概念.....	163
12.4.2	ADO 对象结构.....	164
12.4.3	ADO、DAO 和 RDO	165
12.4.4	ADC/RDS	166
第 13 章	使用 ADO Recordset 对象	168
13.1	创建 ADO Recordset 对象	168
13.2	存取数据库的资料.....	170
13.3	浏览记录集内的资料	171
13.4	分页浏览程序.....	172
13.5	Recordset 对象的属性.....	174
13.6	Recordset 对象的方法.....	185
13.7	Fields 集合和 Field 对象.....	197
13.8	Fields 集合的属性和方法	200
13.9	Field 对象的属性.....	201
13.10	Field 对象的方法	205
第 14 章	使用 ADO Connection 对象	207
14.1	建立和关闭 Connection 对象.....	207
14.2	使用 Connection 对象和数据源建立连接	208
14.3	执行数据库查询.....	209
14.4	使用 Errors 集合进行错误处理	210

14.5	使用事务处理数据	210
14.6	ADO Connection 对象的属性	212
14.7	ADO Connection 对象的方法	216
第 15 章	使用 ADO Command 对象	220
15.1	建立 Command 对象	220
15.2	执行 Command 对象	222
15.3	使用 Command 对象调用存储过程	223
15.4	Command 对象的属性	226
15.5	ADO Command 对象的方法	228
15.6	ADO Command 对象的参数	229
15.7	Parameters 集合的属性和方法	230
15.8	Parameter 对象的属性	231
15.9	Parameter 对象的方法	234
15.10	命令与参数	235
15.11	使用命令和参数的实例	236
第 16 章	ADO 的错误处理	244
16.1	Visual Basic 的错误	244
16.2	ADO 与多重错误	245
16.3	Errors 集合的属性和方法	246
16.4	Error 对象的属性	247
16.5	处理错误状况	249
第 17 章	使用 ADO Property 对象	254
17.1	ADO Property 概论	254
17.2	Properties 集合的属性和方法	255
17.3	Property 对象的属性	256
第 18 章	ActiveX Server Components 设计入门	259
18.1	建立 ASP 组件开发调试环境	259
18.1.1	设置调试帐号	260
18.1.2	改变系统登录	261
18.2	使用 ATL 创建组件	262
18.3	设置调试环境	268
18.4	执行应用程序	271
18.5	恢复测试环境	272
18.6	不用停止 Web 服务的调试方法	273
18.7	文件上载组件	275
附录 I	VBScript 函数摘要	290
附录 II	ASP 网上资源	306

第 1 章

Web 程序开发简介

1.1 Web 的起源

在当今全球信息化大潮中，互联网（即因特网——Internet）异军突起，给 20 世纪末的人类社会增添了无限的活力，成为现代社会生活中一道亮丽的风景线。互联网带给人们的不仅仅是技术，而是一种以信息为标识的崭新的生活方式。它正在改变着人们的工作和生活方式。有人曾以“围城”比喻互联网，但所不同的是：“外面的人想进去，里面的人却不想出来。”那么，互联网何以有如此魅力呢？它又是如何诞生和成长的呢？

其实，互联网衍生于美国政府的一项大型计划。1969 年，美国国防部高级研究计划署（ARPA，Advanced Research Projects Agency）出于对美国国家安全的考虑，希望能建立一个很稳固的网络，让电脑连接到网络上，而且希望电脑的新增或删除不会影响到已有的电脑，这个网络就叫做 ARPA 网。而这个网络最成功的技术就是 TCP/IP（Transmission Control Protocol over Internet Protocol）协议。TCP/IP 是一项分组交换的网络通讯协议，它可以在所建立的电脑网络上使用，确保当网络上任意一个节点出了问题时，并不会影响其它正常的网络节点。现在，Internet 上几乎所有的服务和应用都是以 TCP/IP 为基础开发出来的。

90 年代初，冷战结束，世界格局由两极走向多极化。和平、发展和环保已成为当今国际社会的主要议题。ARPA 网原本是单为军方服务的，但在实际的运行中，人们越来越深入地认识到 ARPA 网可以为更多的人服务。于是，美国国防部在 1990 年正式宣布取消阿帕网。取而代之的是美国国家科学基金会（NSF）管理的互联网。不再局限于一小部分人用的互联网正式登场了，到 1995 年 4 月，NSF 正式完成了互联网的私有化工作。

互联网为什么能风靡全球，这和 WWW（World Wide Web）的出现有莫大的关系。早期的网络中，科学家们使用 FTP 传送文件，使用 Gopher 以选单为基础浏览网络上的资料。80 年代末，在欧洲核物理实验室（CERN）工作的 Tim Berners-Lee 开始寻找取得网络中不同的硬件平台上的资料，并以相同的格式显示出来的方法。Berners-Lee 研读了人们自然阅读的研究报告，最后研究显示，我们视觉处理经验是以页为基础的。当我们阅读书籍、文件或杂志时，我们的目光是上下双向地浏览，同时文件内容和格式也会影响到我们目光的移动，结果是大部分人并非一行行地读，而是往前或往后跳跃地阅读资料。据此，Berners-Lee 得到一个结论：电子资料应以页的方式呈现。Berners-Lee 观察了另一个事实：书籍、杂志及其它阅读资料都有连接，使得读者可以通过这些连接来浏览资料。这些连接可以在主题、

目录、摘要和附录中找到，读者可以通过这些连接在文件中到处翻阅移动。

根据这些观点，Berners-Lee 使用以超文本为中心的管理方式来组织网络上的资料。一组资料通过页面展示出来，称为一个网页，并提出了建立、存取与浏览网页的方法。首先，Berners-Lee 从 Standardized Generalized Markup Language(SGML) 中摘录所需的标记，建立了一个子集合称之为超文本标记语言(Hypertext Markup Language, 简称 HTML), 在其中增加了超链接，并设计了超文本传输协议(Hypertext Transport Protocol, 简称 HTTP) 来取得超链接文件。最后，HTTP 和 HTML 使用 Uniform Resource Locator(URL) 来定位网络上的文件，以提供一个分散的客户端/服务器系统的基本结构，这个系统 Berners-Lee 称之为 World Wide Web, 即 WWW。这里，用来保存资料信息并处理客户请求的电脑是 Web 服务器，而用来查询并浏览资料的工具就是浏览器。

WWW 的出现使互联网上的信息一下子就活跃起来。WWW 的最大贡献在于使互联网真正成为交互式的。人们可以浏览自己感兴趣的网站，可以给 Web 网站增加内容和发布消息，也可以编辑 Web 网站上的内容，甚至还可以通过网络跟天南海北的朋友聊天。

互联网的诞生是人类交流方式的一次伟大革命。以电子邮件的普遍使用为标志的阿帕网带来了通信方式的革命；而 WWW 和网络浏览器的普遍使用则标志着网络使用方式的革命。

世界上第一个 WWW 浏览器 Mosaic 出自 Mark Andreeseen 之手。1992 年，Mosaic 在互联网上一出现就马上风行起来，竟然使 WWW 当年网络数据传输量增长了 341 634%。网络浏览器这些年不断推陈出新，发展至今，功能也越来越强大。现在比较著名的网络浏览器有微软的 IE(Internet Explorer) 及网景公司(Netscape) 的 Navigator。

网络浏览器的发明，使 WWW 立即成为人们所关注的焦点，同时网络用户也迅速增加起来。由于这些用户并不仅仅是到网络上“冲浪”，其中有不少人也希望在网络上发布自己制作的网页。据统计，1994 年前后的 18 个月中，互联网上的用户就创建了 300 万个网页。人们在庆幸共享资源逐渐丰富之余，又不得不面对信息爆炸的问题。互联网已经成了信息的海洋，在网上查询所需要的信息则犹如大海捞针，要解决这个问题就必须借助有效的网络搜索引擎。

1994 年，杨致远和戴维·菲罗设计了现在 WWW 上著名的搜索引擎“雅虎(Yahoo)”。雅虎的搜索引擎使大海捞针成为可能。现在，只要你用键盘输入自己感兴趣的人名、关键字等，搜索引擎就会立即把相关的信息列出来。雅虎的出现对网络浏览起到了推波助澜的作用。

互联网发展到今天，正逐步渗透到社会生活的各个层面。网上的世界日新月异，越来越精彩，Web 网站的开发与管理技术也随着不断翻新进步。本书着重介绍的 ASP 技术就是其中的佼佼者之一。以下我们先简单地介绍网站开发技术发展的历程。

1.2 超文本标记语言(HTML)

HTML 全称 Hypertext Markup Language, 即超文本标记语言，主要被设计用来格式化文字。早期的 HTML 基本上是一种排字语言，它告诉电脑文字要使用的颜色、文字大小，

以及文字的形状。这和大部分的排字语言没有多大差别。人们不断对 HTML 进行完善和更新，例如，可以加入多窗口，可以使用表格细致地安排网页格式，可以使用 Imagemap 对图形的部分区域建立连接。

HTML 与其它标记语言的不同点在于：它可以和相关内容建立超链接，并且可以通过表单标记产生互动的页面。在最新版本的浏览器中，诸如 JavaScript、VBScript、Java Applet、ActiveX 等新技术被加入到浏览器中。

1.3 超文本传输协议(HTTP)

HTTP 全称 Hypertext Transport Protocol，即超文本传输协议。HTTP 是建立在 TCP/IP 的应用层之上的，它是 Web 服务器上的信息来源与浏览器之间交流的核心通讯协议。

HTTP 协议是没有状态的，它的一般实现过程如下：

- (1) 连接 —— 客户端与指定的服务器建立连接。
- (2) 请求 —— 连接建立好了以后，客户端提出请求送往服务器。这个请求一般包含以下信息：客户端使用的通讯协议，要请求的对象名称，对象在服务器上的位置，服务器应当如何回应，以及客户端使用什么方式来取得这个对象(GET 或者 POST)。
- (3) 响应 —— 服务器收到客户端的请求后，取得相关对象并送回客户端。
- (4) 关闭 —— 在接受完对象后，关闭连接。

每一次客户端请求都包含以上四个步骤。HTTP 协议还描述了更多的实现内容，这里就不再描述了。

1.4 公共网关接口(CGI)

1.4.1 什么是 CGI 程序

早期的 Web 网站设计很简单，没有多少技术含量，网站设计更像是资料编辑的过程，CGI 的出现改变了这一现象。

CGI 全称为 Common Gateway Interface，即公共网关接口。那么 CGI 是做什么用的呢？

如果你在 Yahoo 搜索引擎上搜寻过资料，就应该知道，首先要将希望查找的关键字输入到 Yahoo 首页的关键字框中，然后执行搜寻，很快 Yahoo 就会将符合条件的资料返回给你。这里，当执行搜寻时，浏览器就将输入的资料作为一个请求传送给 Web 服务器，而 Web 服务器由谁来处理这个请求呢？Web 服务器又如何得到你传递过来的资料呢？答案就是 CGI。Web 服务器上已经放置着一个设计好的 CGI 程序，它通过 CGI 规定的环境变量得到你传送的资料，在 Web 服务器的数据库里查找到符合条件的资料，再用超文本的方式返回。这就是 CGI 所做的工作。

CGI 可以存取数据库与其它应用程序，回应用户的输入并动态地产生超文本网页的界

面。下面我们来了解 CGI(Common Gateway Interface) 的具体含义。这里的 Common 是指为了产生一个高级且跨平台的界面, 而用一个标准的通讯协议来处理服务端程序, 以及客户端程序与应用程序之间的数据流; Gateway 是指 CGI 程序就像是一个 HTTP 服务端与其它可以在运行时接收命令的应用程序的媒介; Interface 则是这个规格提供给程序开发者一个与 HTTP 服务端独立的环境。编写 Gateway 时唯一需要你做的便是去了解如何通过 HTTP 通讯协议来处理数据流。

CGI 程序不过是一串指令的集合而已。当然, 这些指令必须遵循 CGI 的标准, 而且可以及时执行, 另外, 它还可以执行使用者定义的工作以及提供动态的输入。CGI 程序的过程可以分为三个主要的部分: 读、执行和转换。所谓读是指读取服务器提供的资料, 有必要的話, 还需要对资料的格式进行适当的转换以方便后续处理; 执行就是执行资料提取或执行特定指令; 转换则是把程序的结果转换成为 HTML 格式并将该格式化信息送到标准输出。

任何一种可以在服务端执行的语言均可用来编写 CGI 程序。例如, 在 Windows NT Server 平台上, 便可以使用 C/C++、Perl 和 Visual Basic 等语言。所有的程序代码都是储存在服务器上, 并且是由客户端/服务器连接的客户端来启动的。启动 CGI 后, 服务器会暂时储存局部环境变量, 如服务端的 IP 地址、使用者的输入、文件、文件的 URL 等信息。由于 CGI 程序或界面通常是内嵌在某个超文本网页里的, 因而被当作是一个参考的链接。当这些参考链接根据使用者的输入而自动地或动态地启动时, 其相对应的程序便会随之执行起来。在第一种情况中, 当客户端请求某个具有 Script 参考的网页时, 相应的程序就会同时启动。CGI 有两种方式来完成由使用者启动的方式, 其一便是直接使用在画面上的超文本链接来驱动程序的执行; 另外一种方式则是由使用者填写电子资料并传递给 Web 服务器。一般来说, 在网络上搜寻数据库是用后一种方式。

CGI 的出现使用户通过浏览器就可以访问远端的数据库, CGI 在接到客户端的输入后, 发出对数据库搜寻的请求, 接着回传结果给客户端。这中间共有七个步骤:

- (1) 使用者在客户端的 PC 上使用在线的格式来传递欲搜寻的字符串。
- (2) 使用者的输入是以 HTTP 请求的方式传送给 Web 服务器的。
- (3) Web 服务器启动 CGI。
- (4) CGI 将搜寻的字符串转换成 SQL 搜寻语句, 并将它传给数据库。
- (5) 数据库传回搜寻的结果给 CGI 程序。
- (6) CGI 将结果转成 HTML 的格式并将这个信息传回 Web 服务器。
- (7) Web 服务器传回这个格式化的结果给客户端。

正是因为网络上有了 CGI, 才弥补了浏览器存取数据库能力的不足。这样商业上的需要也才真正得到了满足。这就为互联网技术在企业环境的普及与应用奠定了基础。同时, CGI 成为了大多数网站的主要元件也就成了顺理成章的事情。

至今, 仍然有大量的网站在使用 CGI 程序, 尤其是在 UNIX 平台上。

1.4.2 CGI 程序的输入

当调用一个 CGI 程序时，一些有用的信息同时被产生，这些信息可以分为以下三部分：

- 关于 Client、Server 和使用者的信息；
- 使用者所提供的表单资料；
- 额外的文件路径名称信息。

大部分关于 Client、Server 和使用者的信息都存放在 CGI 的环境变量中，使用者提供的表单资料可能放在环境变量内，也可能直接放在 Request 的主体内；额外的文件路径名称信息一样是放在环境变量中。

CGI 的环境变量对于 CGI 程序的输入是非常重要的，让我们来看一看服务器变量对应 CGI 的环境变量，它们的参数值说明如表 1-1 所示。

表 1-1 CGI 环境变量

参 数 值	说 明
AUTH_TYPE	服务器的正确辨识方式，如使用者名称
CONTENT_LENGTH	客户端发出内容的长度
CONTENT_TYPE	指出正在传输的资料的 MIME 类型，如“ text/html ”
GATEWAY_INTERFACE	服务器使用的 CGI 版本
HTTP_HeaderName	新建的专用页首
LOGON_USER	视窗 NT 说明所用。用户登录 Windows NT 的帐号
PATH_INFO	服务器上所需文件的虚拟路径
PATH_TRANSLATED	服务器上所需档案的绝对路径
QUERY_STRING	查询 HTTP 请求中间号(?)后的信息
REMOTE_ADDR	客户端机器的地址。发出请求的远程主机(Client) 的 IP 地址
REMOTE_HOST	客户端机器的名称。发出请求的主机(Client) 名称。如果服务器无此信息，它将设置为空的
REQUEST_METHOD	提出请求的方式，如 HTTP 的 GET、PUT、HEAD、POST 等
SCRIPT_MAP	URL 的网域名称
SCRIPT_NAME	程序执行时的虚拟路径
SERVER_NAME	连接客户端服务器的使用端口。出现在 URL 中的服务器主机名、DNS 化名或 IP 地址
SERVER_PORT	发送请求的端口号。Web 服务器回答要求
SERVER_PORT_SECURE	1 为一安全的使用端口，0 则为一非安全的使用端口
SERVER_PROTOCOL	Web 服务器回传文件的传输协议，服务器版本号 如 HTTP/1.0)
SERVER_SOFTWARE	服务器所使用的网络软件
LOCAL_ADDR	主机 IP 地址，返回接受请求的服务器地址。如果在绑定多个 IP 地址的多宿主机器上查找请求所使用的地址时，此变量非常重要

续表

参 数 值	说 明
ALL_HTTP	客户端发送的所有 HTTP 标题头

以下是一个用 Perl 语言编写的小程序，用来显示 Server 的一些不同信息，如 CGI 和 HTTP 的版本等。

```
#!/usr/local/bin/perl
print "content-type: text/html" , "\n\n";
print "<html>" , "\n";
print "<head><title>About this server</title></head>" , "\n"
print "<body><h1>About this server</h1>" , "\n"
print "<hr><pre>";
print "server Name: " , $ENV{'SERVER_NAME'} , "<BR>" , "\n";
print "Running on Port: " , $ENV{'SERVER_PORT'} , "<BR>" , "\n";
print "Server Software: " , $ENV{'SERVER_SOFTWARE'} , "<BR>" , "\n";
print "Server Protocol:" , $ENV{'SERVER_PROTOCOL'} , "<BR>" , "\n";
print "CGI Version: " , $ENV{'GATEWAY_INTERFACE'} , "<BR>" , "\n";
print "<hr></pre>" , "\n";
print "</body></html>:" , "\n";
exit( 0 );
```

CGI 环境变量是非常重要的，实际上我们后面所提的 ISAPI 和 ASP 编程，它们都是通过使用 CGI 环境变量来取得客户端信息的，只是实现细节上有所不同而已。

1.5 Internet 服务器应用程序接口(ISAPI)

1.5.1 从 CGI 到 ISAPI

对大部分的 Internet 程序来说，CGI 是开发 Web 程序较佳的一种选择，但 CGI 也有其不足之处。它主要有以下两个缺点：

- (1) 无法取得不同 HTTP 对话的 Context 信息；
- (2) 执行时需要独立的进程(Process)。在访问人数增多的时候甚至会使服务器崩溃，因为每个访问服务器的人都需要一个独立的进程来执行 CGI 程序。

为了弥补 CGI 的缺失，微软创建了 ISAPI(Internet Server Application Programming Interface)。ISAPI 是一个可以在 Web 服务器上执行的程序，并能够和 Web 浏览器动态地进行交互的程序界面。微软把 ISAPI 当作一套通用的扩展例程，用于调用外部应用程序，以及控制浏览器和服务器之间的数据流量。

ISAPI 是一个开放式的技术规范，它得到了 Windows NT 和其它操作系统上的第三方

Web 服务器的支持。ISAPI、IIS 和 Windows NT Server 三者完美结合而产生的 HTTP 平台，其特色是高性能、高性能价格比及可变规模。在 ISAPI 的体系结构中，可以使用动态链接库(DLL)的格式编写多线程、多实例的应用程序。这使得 ISAPI 可以重用资源，而且处理器开销较少。

ISAPI 的发展主要表现在以下两个方面：

(1) 在 ISAPI 应用程序上克服 CGI 的性能问题。运用 ISAPI 所编写的程序之所以要比 CGI 快，是因为操作系统无需像执行 CGI 程序一样，每次都进行环境的复制并启动一个新的进程。另外，ISAPI 程序本身就是一个 DLL，故而在执行时可以动态地被载入到内存中。

(2) ISAPI 过滤器可实现对发送到 IIS 和从 IIS 返回的信息进行的预处理和后处理。例如，你可以编写一个过滤器，只允许指定的浏览器访问你的网站。

相比较而言，ISAPI 要比 CGI 功能更强大，运行更快，表现更灵活。但在 ISAPI 编程中却需要高级编程技巧，要求编程人员必须对 Microsoft C++ 开发系统相当熟悉。

在 IIS 4.0 发行之前，ISAPI 程序和 IIS 位于同一个进程地址空间(Address Space) 内，因此，一旦任何的 ISAPI 程序出错死机，整个 IIS 就会停止工作。但从 IIS 4.0 开始，ISAPI 程序的执行就已经独立于 IIS 进程了。

1.5.2 ISAPI 扩展动态链接库的入口函数

ISAPI 程序是一个拥有许多自定规则的 DLL，每个 ISAPI 程序都有以下两个程序的进入点(Entry Point)：

- GetExtensionVersion
- HTTPExtensionProc

当 IIS 第一次载入一个 ISAPI 程序时，它会去调用 GetExtensionVersion 函数，这个函数主要用来辨认 ISAPI 程序的版本，其函数原型如下：

```
BOOL WINAPI GetExtensionVersion( HSE_VERSION_INFO *pVer )
```

而另一个进入点 HTTPExtensionProc 函数原型如下：

```
DWORD HTTPExtensionProc( LPEXTENSION_CONTROL_BLOCK *lpEcb )
```

它是你的处理过程的真正入口点。

1.5.3 ISAPI 程序的输入：Extension Control Block(ECB)

在 CGI 程序中，需要由用户端传入一系列的环境变量。ISAPI 也有类似的要求。这就是我们上面谈到的 HTTPExtensionProc 所需要的 EXTENSION_CONTROL_BLOCK 结构，如表 1-2 所示。

表 1-2 EXTENSION_CONTROL_BLOCK 结构

名 称	说 明
cbSize	目前 EXTENSION_CONTROL_BLOCK 结构的大小