



网络实用技术丛书

ASP与SQL Server 网站架设

393.092

贾佳 郝洪明 主编



机械工业出版社
China Machine Press

网络实用技术丛书

TP393.092
J32

ASP与SQL Server 网站架设

贾 佳 郝洪明 主编



A0875688



机械工业出版社
China Machine Press

本书介绍用微软公司的ASP与SQL Server 建设动态网站的方法。内容包括ASP基础知识、应用程序、内置对象及组件, SQL Server的操作及结构化查询语言, 数据库访问组件等。本书用大量实例讲解了如何使用这两个先进工具进行站点设计, 内容丰富、通俗易懂, 适合初学者学习网站设计方法, 也适合想了解网站设计方法的人员参考。

版权所有, 侵权必究。

图书在版编目(CIP)数据

ASP与SQL Server网站架设 / 贾佳等主编. - 北京: 机械工业出版社, 2001.1

(网络实用技术丛书)

ISBN 7-111-08612-0

I. A... II. 贾... III. ① 网站-设计-应用软件, ASP ② 关系数据库-数据库管理系统, SQL Server IV. TP393.092

中国版本图书馆CIP数据核字(2001)第00681号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 金 锋

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2001年1月第1版第1次印刷

787mm×1092mm 1/16·13.75印张

印数: 0 001-5 000册

定价: 22.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前 言

随着Internet的兴起和以网页为载体的网络信息的广泛传播,网页制作中对于内容的动态显示与更新需求量越来越大。编写一个令人满意的动态网页已成为许多网页设计人员的目标。在众多的编写环境中,Microsoft Active Server Pages(ASP)以其强大的技术力量背景及容易掌握的语言环境迅速占领了大片市场、获得了众多网页设计人员的青睐。为了使更多的人掌握ASP应用程序的编写,本书以丰富的事例、简单的语言较全面的介绍了ASP应用程序的设计方法与技巧,并详细地介绍了Microsoft SQL Server在ASP应用程序中的应用。

全书共分十三章。前三章讲述什么是ASP,怎样开发ASP应用程序,ASP的内置对象及内置组件的详细使用方法。第4章讲述Microsoft SQL Server及结构化查询语言。第5章全面讲解数据访问对象ADO及其使用方法。第6章用一个留言簿的应用例子演示了上面介绍的内容。第7章介绍如何调试ASP的应用程序及如何注意ASP应用程序的效率。第8章用一个简单的例子介绍如何在ASP中自建组件。第9至13章用丰富实用的例子演示了ASP的各种功能。

本书由贾佳和郝洪明主编,参加编写的还有:张强、刘小汕、赵欣、胡泽阳、石友华、谢娜、姜明、陈煜;黎攻、刘源平校审了全书。限于作者水平,书中内容难免有疏漏和不妥之处,敬请广大读者不吝指正。

2000年11月

第1章 ASP初次接触

本章要点:

- 什么是ASP? 它能帮助我们做什么?
- 怎样开发ASP程序? 它在什么环境下运行?

1.1 ASP简介

1.1.1 什么是ASP

Microsoft Active Server Pages (ASP)实际上是将标准的HTML文件拓展了一些附加特征, ASP像标准的HTML文件一样包含HTML语句并且在一个浏览器上解释并显示。但它为HTML编写人员提供了在服务器端运行脚本的环境, 使HTML编写人员可以利用VBScript和JScript或其他第三方脚本语言来创建ASP, 实现过去需要编写复杂的CGI程序才能实现的有动态内容的网页, 如计数器等。一个ASP文件的后缀为.asp, 其中包含实现动态功能的VBScript或JScript语句, 如果去掉那些VBScript或JScript语句, 它和标准的HTML文件没有任何区别。ASP提供了一些内建对象(参见后面关于内建对象的章节)。利用这些内建对象, 你可以使你的脚本更加强大; 这些对象会允许您从浏览器中接收和发送信息。例如, 利用Request对象, 你可以接收用户的HTML表单中的信息并加以处理。ASP包含标准的ActiveX组件(参见后面关于组件的章节), 这些组件可以实现一些复杂的功能, 如使用数据库等。不过, 你并不会被此局限, 你可以轻松创建属于你自己的附加ActiveX组件。这意味着你可以不加限制地拓展你的ActiveX组件(参见后面关于自制组件的章节)。ASP可以和诸如SQL Server这样的数据库进行挂接, 在本书后面的章节中, 我们将详细地学习这方面的内容。重要的是, 这是ASP一个非常强大的功能所在, 在线商务以及在线论坛等各种非常高级的、动态更新的站点都需要数据库的支持, 而且需要随数据库内容的更新而自动更新, 这样, 你就可以利用ASP自己建立很多类似的高级站点。

通过上面的描述, 我们可以了解到, ASP就是由服务器端脚本、对象以及组件拓展过的标准网页并为其提供了运行的环境。而它的网页并不是在建立初期存在的, 而是当某个浏览器向它提出请求时, 它才根据需要产生所需要的标准网页, 这克服了过去HTML编写的网页的静态缺点, 从而使网页上可以有許多动态的信息存在。如: 当前的计算机时间、计数器的计数值等。

1.1.2 ASP的功能

ASP是一套服务器端的脚本运行环境, 当用户从浏览器向Web服务器请求 .asp 文件时, ASP 脚本开始运行, 然后 Web 服务器调用 ASP, ASP全面读取请求的文件, 执行所有的脚本命令, 并将标准的Web 页传送给浏览器, 而并不包含你所写的脚本命令。

对于Web 服务器来说, ASP与HTML有着本质的区别, HTML不经任何处理送回给浏览器, 而ASP的每一个命令都首先被用来生成HTML文件, 因此ASP允许生成动态内容。

另一方面, 对于浏览器来说, ASP和HTML几乎是没有区别的, 仅仅是后缀为.asp和.htm的区别, 当我们在客户端提出对ASP的申请后, 我们的浏览器接受的是HTML格式的文件。因此它适用于任何浏览器。

根据以上特性, 我们用ASP可以方便地实现诸如表格信息收集、计数器、留言簿、公告板、聊天室甚至电子商务等过去必须由CGI才能实现的功能。

ASP还可以轻松地实现对页面内容的动态控制, 为不同的浏览者定制不同的页面内容, 实现个性化的网站。

1.1.3 使用范围

你在利用ASP实现各种功能时几乎没有什么限制, 只要网站存在并正常运行, 以下是简单的应用示例列表:

- 在你的网站主页顶部添加一个滚动显示的广告栏。
- 从HTML的表单中接受信息并且存到数据库中。
- 根据不同访问者显示不同内容, 创建个性化主页。
- 在你的主页中添加点击计数器。
- 根据用户浏览器的版本、类型和能力显示不同档次的内容。
- 连接多个主页使之容易网际导航。
- 跟踪用户网站上的活动信息并且存入日志文件。
- 使用基于Windows NT的Intranet建立你的MIS应用。

用ASP编写出的系统, 完全可以达到原来使用Client/Server的网络数据库系统的水平。使用ASP最大的好处是, 在你对程序进行升级时, 你只需要修改服务器上的ASP文件, 而不需要修改客户端程序。

以上只是简单介绍, 看完下面章节, 你会对利用ASP做的事情有更加全面深入的了解。你会真正学会如何使用ASP建立各种应用。

1.2 ASP的开发

1.2.1 环境的建立

ASP是微软开发的服务器端脚本环境, 内含于 IIS (Internet Information Server) 或PWS (Personal Web Server) 中, 开发ASP应用程序系统必须符合以下三个要求之一:

- 如果你使用的是Windows NT Server系统, 那么你需要安装版本为3.0/4.0的Internet Information Server (IIS), 3.0版本的IIS包含在Microsoft Visual Studio 97中, 也可到<http://www.microsoft.com>去下载4.0版本的IIS。安装好后, IIS会在Windows NT所在的硬盘上建立其主页发布路径\inetpub\wwwroot, 而IIS会在机器启动后自动执行。把你编写好的ASP文件复制到这个路径下, 当然, 你应该具备在这个路径的读写权限。打开浏览器, 在

地址输入框内输入`http://localhost/test1.asp` (假设你编写的ASP文件的名字是`test1.asp`), 接着, 你便能看见运行结果了。

- 如果你使用的是Windows 95/98系统, 那么你需要安装 Personal Web Server (PWS), 它可以在Windows 98 的安装盘的Tools\PWS路径下找到。安装好后, PWS会在Windows所在的硬盘上建立其主页发布路径`Inetpub\wwwroot`, 而PWS会在机器启动后自动执行。把你编写好的ASP文件复制到这个路径下, 打开浏览器, 在地址输入框内输入`http://localhost/test1.asp` (假设你编写的ASP文件的名字是`test1.asp`), 接着, 你便能看见运行结果了。
- 如果你拥有自己租用的虚拟空间, 并且有CGI和ASP的权限, 则可直接把文件复制到管理员给你的路径上。

注意, 如果你在IIS的当前机器上使用浏览器, 请注意你怎样才能在浏览器上执行ASP。当你用网络浏览器时调用ASP, 不能直接在`open` 命令中调用, 在这种情况下, IIS以及ASP是不会运行的。此时, 你将看到ASP脚本的文本而不是它执行的结果。这肯定不是你所希望的。

ASP自身提供了VBScript和JScript的驱动程序。ASP可以将可执行的脚本直接嵌入HTML文件, HTML开发和脚本开发在同一开发过程中就可完成。

1.2.2 开发工具

实际上ASP的应用程序就是以`.asp`为扩展名的纯文本文件, 你可以用任何文本编辑器打开并编辑它 (如记事本)。ASP程序中可以包含注释、HTML标记以及脚本命令。你只需将`.asp`程序放在Web服务器的虚拟目录下 (该目录必须要有可执行权限), 就可以通过HTTP的方式访问ASP程序了。

使用那些带有 ASP 增强支持的编辑器将更能提高效率, 如 Microsoft Visual InterDev。Visual InterDev 是数据库工具, 它能轻松地建立“开放数据库连接”(ODBC)的数据库。如果您从未使用过 HTML, 请考虑先使用 Microsoft FrontPage。使用 FrontPage 创建文档和格式化文本就像使用文字处理工具一样简单。接下来还可以使用 Insert Script 命令在FrontPage 创建的HTML 页中插入简单的 ASP 命令。

1.2.3 程序编写

在ASP中, 所有脚本命令都由定界符`<%` 和`%>` 包含, 任何在这个符号中包含的内容都被认为是一个脚本, 你可以在其中插入任何命令, 只要这个命令对正在使用的脚本语言有效即可。下面是个例子:

```
<html>
<head><title>ASP Script 示例</title></head>
<body>
这是个
<% FOR I=1 to 10 %>
非常,
<% NEXT %>
非常长的句子。
```

```
</body>
</html>
```

在记事本中输入以上代码，保存为\Inetpub\wwwroot \test.asp，打开浏览器，在地址输入框内输入http://localhost/test.asp。

该段Active Server Pages执行后在浏览器上显示为：

这是个非常，非常，非常，非常，非常，非常，非常，非常，非常，非常，非常，非常长的句子。

这段脚本利用VBscript的For...Next循环生成了10个“非常”的拷贝。

ASP默认为您主要使用的是VBscript，也就是说，在使用<%和%>时不需要做任何事情来说明使用何种语言。ASP也可以使用任何一个可以和Active X Script标准兼容的脚本语言引擎，不过，需要用ASP的处理指令来指定。下面描述具体的处理指令。

ASP 提供了您使用的脚本语言以外的指令。这是一些输出指令和处理指令。

ASP 的输出指令 <%= expression %> 显示表达式的值，expression为任何有效的表达式。以下是个例子：

```
<html>
<head><title>ASP示例</title></head>
<body>
现在的时间：<%=time%>
</body>
</html>
```

在上面的例子中，VBScript的时间函数值会输出到显示器上。

你可以用下面的方法完成同样的事，见下面例子：

```
<html>
<head><title>ASP示例</title></head>
<body>
现在的时间:<%Response.Write(TIME)%>
</body>
</html>
```

在这个例子中，VBScript的Time函数值将被ASP的Response对象输出来，response对象的write()方法是将表达式中的值显示出来（下一章中，你会学到更多有关对象和方法的知识）。

那么补充说明的是对于ASP内部<%=和%>也就是代表了Response.write()方法，因此无论你用哪一种方法其实是一样的，没有任何区别。

ASP 处理指令 <%@ keyword %> 为 ASP 提供处理 .asp 文件所需的信息。

处理指令必须出现在 .asp 文件的第一行。必须在@标记和关键字之间加入一个空格。处理指令有下列关键字：

- LANGUAGE 关键字设置页的脚本语言：<%@ LANGUAGE=VBScript %> 将本页的主脚本语言设置为VBScript。ASP的默认主脚本语言为VBScript，你也可以将任何一种具有脚本引擎的脚本语言设置为本页的主脚本语言。主脚本语言是用来处理在分界符 <% 和 %> 之间的命令的语言。
- CODEPAGE 关键字设置页的代码页（字符编码）：指明 ASP 处理某一特定页时所用的字

符集, <%@ CODEPAGE= 950 %>将本页代码页设置为中文繁体字符集。

- **LCID** 关键字设置页的现场标识符: 现场决定日期和时间、项目的格式, <%@ LCID = 1041 %>把现场设置为日文现场。
- **TRANSACTION** 关键字指定将在事务处理环境下运行的页。
- **ENABLESESSIONSTATE** 关键字指定 ASP 页是否使用会话状态。

可以在单个指令中包含多个关键字; 关键字/值对之间必须由空格分开。不要在等号 (=) 左右加入空格。下面的例子中设置了脚本语言和代码页:

```
<%@ LANGUAGE= VBScript CODEPAGE=950 %>
```

如果您的主脚本语言是 VBScript 或者 JScript, ASP 从指令中删除空白字符。使用其他脚本语言时, ASP 保留空白字符。空白字符包括空格、制表符、回车和换行符。

使用 VBScript 和 JScript 时, 在打开定界符之后和关闭定界符之前使用空白字符更容易阅读命令。下面的命令都是合法的:

```
<% Color = "Green" %>
<%Color="Green"%>
<%
Color = "Green"
%>
```

1.2.4 Global.asa文件

为了使ASP更好地工作, ASP设置了一个专用的Global.asa文件, Global.asa文件是一个可选文件, 用户可以在该文件中指定事件脚本, 并声明具有会话和应用程序作用域的对象。该文件的内容不是给用户显示的, 而是用来存储事件信息和由应用程序全局使用的对象。该文件的名称必须是 Global.asa 且必须存放在应用程序的根目录中。每个应用程序只能有一个 Global.asa 文件。

Global.asa 文件只能包含如下内容:

- 应用程序事件。
- 会话事件。
- <OBJECT> 声明。
- TypeLibrary 声明。

应用程序有两个事件, 即 Application_OnStart 事件和 Application_OnEnd 事件。

会话有两个事件, 即 Session_OnStart 事件和 Session_OnEnd 事件。

可以用任何支持脚本的语言编写 Global.asa 文件中包含的脚本。如果多个事件使用同一种脚本语言, 就可以将它们组织在一组 <SCRIPT> 标记中。

当用户保存对 Global.asa 文件所做的更改时, 在重新编译 Global.asa 文件之前, 服务器会结束处理当前应用程序的所有请求。在此期间, 服务器拒绝其他请求并返回一个错误消息, 说明正在重新启动应用程序, 不能处理请求。

当用户当前的所有请求处理完之后, 服务器对每个会话调用 Session_OnEnd 事件, 删除所

有活动会话，并调用 Application_OnEnd 事件关闭应用程序，然后编译 Global.asa 文件。接下来，用户的请求将启动应用程序并创建新的会话，触发 Application_OnStart 和 Session_OnStart 事件。

但是，对 Global.asa 文件中所包含的文件的更改与保存并不能使服务器重新编译 Global.asa。为了让服务器识别包含文件的改动，必须再重新保存一下 Global.asa 文件，如在 Global.asa 文件中加一个空格，然后保存它。

1.3 ASP的安全性

不论是在 Windows 98 还是 Windows NT 下编写调试 ASP 应用程序，最终 ASP 应用程序还是会应用在基于 Microsoft Internet Information Server (IIS) 的 Internet 或 Intranet 上，那么，它是不是安全呢？IIS 是运行在 Windows NT Server 上的信息服务，它提供 FTP、WWW 等广泛使用的 Internet 和 Intranet 服务。而 ASP 的运行环境由 IIS 提供，所以，IIS 是否安全直接影响到 ASP 的安全。

1.3.1 Windows NT 和 IIS

IIS 是 Windows NT Server 上的信息服务，它与 Windows NT 的用户账户和用户组有着密切的联系。Windows NT 安全性的核心之一是用户账户和组。在安装 IIS 的过程中，自动建立两个用户账户，它们被赋予一定的用户权力，并被放入特殊的组。这两个账户是 IUSR_计算机名 和 IWAM_计算机名。IUSR_计算机名用于匿名访问。IWAM_计算机名由 Microsoft 事务处理服务 (MTS) 和其他 IIS 应用程序使用。

由于 IIS 是 Windows NT Server 本地的一个服务，而账户 IUSR 又用于 IIS，相当于用户以交互方式登录到服务器上，所以必须赋予 IUSR_计算机名 “在本机登录” 的权力。任何匿名访问者都具备赋予 IUSR_计算机名的权力。如果不使用这个缺省匿名账户，可以指定其他用户账户作为匿名账户，并为这个账户指定合适的用户权力。

在提供其他验证方式时，IIS 与 NT 用户账户也有着密切的联系，这些方式都需要有效的用户账户。虽然 IIS 建立了匿名用户 IUSR_计算机名，但并没有建立用于其他验证方式的用户账户，应由管理员建立并管理与这些验证方式相关的用户账户。

Windows NT 提供两种文件系统，FAT 和 NTFS。虽然 IIS 能在 FAT 分区上正常工作，但 NTFS 文件系统有更多的优越性，例如：

- 1) 对于 FAT 文件系统（如 DOS），NTFS 是不可见的，避免了使用 DOS 命令对资源的破坏。
- 2) NTFS 文件系统能更有效地管理硬盘资源。
- 3) NTFS 允许对用户和组设置资源的访问权限，即配置资源的访问控制列表 (ACL)。

ACL 是指与某一资源，如文件或目录有关的用户、组及他们对资源的访问权限。ACL 是 Windows NT 安全模型的又一个核心内容，它提供了对硬盘资源灵活、精确的访问控制。每个目录或文件都有自己的 ACL，每个 ACL 本身又有一个 ACL，用于指定谁有权查看和更改 ACL。NTFS 和 ACL 是保护服务器资源的基础。

IIS 提供 Web 权限，如读和写，NTFS 也有它的读写权限，两者容易引起混淆。IIS 权限与 NTFS 权限的关键区别在于：Web 权限控制允许对 HTTP 资源执行的 HTTP 命令，NTFS 权限控制

用户对硬盘资源的访问权限。当浏览器发出资源请求时, HTTP命令随头信息一起发送, 例如, 一个用户请求读取(GET)一个Web页, 发送的头信息指定资源、资源的位置, 另外还要发送动词GET, 这个动词告诉IIS, 本次请求要对资源进行“读”操作。选择IIS的Web主目录标签、目录标签或虚拟目录标签的“读取”、“写入”复选框, 可以设置上述Web权限。如果Web权限是读, 并不意味着也设置了NTFS的读权限。因为Web权限只控制在HTTP请求中能够使用的HTTP命令。反过来, 如果设置NTFS权限是读, 意味着Web权限也是读。如果NTFS权限是读, 即使Web权限是读写, HTTP的写请求也会失败。如果Web权限和NTFS权限不一致, HTTP请求将使用两者权限中最严格的一个。

1.3.2 IIS安全漏洞

当ASP以其灵活、简单、实用、强大的特性迅速风靡全球网站的时候, 其本身的一些缺陷、漏洞也正威胁着所有的网站开发者。所有的ASP应用程序开发者应密切关注, 提高警惕。

微软再次被指责对其出品的Web服务器软件的安全问题不加重视。在微软的流行产品IIS Sever4.0中发现存在一种被称为“非法HTR请求”的缺陷。据微软称, 此缺陷在特定情况下会导致任意代码都可以在服务器端运行。但用发现这一漏洞的Internet安全公司eEye的CEO Firas Bushnaq的话说: 这只是冰山一角而已。Bushnaq说, 微软隐瞒了些情况, 比如黑客可以利用这一漏洞对IIS服务器进行完全的控制, 而恰恰许多电子商务站点是基于这套系统的。

尽管IIS存在一些漏洞, 但它的强大功能正被许多程序开发者所认识, 而微软也在不断地发行更高版本的ASP, 以提高其安全性。我们不能因为一些缺陷而放弃功能如此强大而又易于学习使用的服务器应用程序开发环境。为了尽量使自己的网站安全, 使自己的ASP应用程序安全, 应合理进行IIS的系统配置。

以下是IIS系统配置的安全性建议, 希望能给大家一些帮助:

- 使用最新版本的IIS4.0, 并安装NT最新版本的Service Pack5, 服务器的文件系统不要使用FAT, 应该使用NTFS。
- 把IIS中的sample、scripts、iisadmin和iisadc等Web目录设置为禁止匿名访问并限制IP地址。在微软还没有提供补丁之前, 把ism.dll有关的应用程序映射删除。
- 有条件的话就采用防火墙机制, 最简单的方法有, Web服务器开在前台, 目录放在后台。如果能用一个服务器一台机当然最好。
- Web目录、CGI目录、scripts目录和winnt目录等重要目录要用NTFS的特性设置详细的安全权限, 包含注册表信息的Winnt目录只允许管理员完全控制, 一般的用户只读的权限也不要给。凡是与系统有关的重要文件, 除了Administrator, 其他账号都应该设置为只读权限, 而不是everyone。
- 只开你需要的服务, 关掉所有不应该打开的端口, 如NetBios端口139, 这是一个典型的危险端口; 怎样禁止这些端口? 除了使用防火墙外, NT的TCP/IP设置里面也提供了这种功能: 打开“控制面板|网络|协议|TCP/IP|属性|高级|启用安全机制|配置”, 这里面提供了TCP和UDP端口的限制和IP协议的限制功能。
- 管理员的账号要设置得复杂一些, 建议加入特殊字符。

- 把FTP、Telnet的TCP端口改为非标准端口，通常笔者都是设置到10 000~65 000的范围。
- 删除可以删除的所有共享，包括打印机共享和隐藏的共享如ICP\$和Admin\$等，微软说这些特殊共享资源很重要，大多数情况下不能删除，而实际上放在Internet上的机器大多数不需要共享。IPC\$: 适用于远程管理计算机和查看共享资源,在网上最好不要用。Admin\$: 实际上就是 c:\winnt，也没有必要共享。C\$: 登录为Admin和Backup-operator的用户可以用\\计算机名\C\$的方式访问C盘，虽然仅限于局域网,但是远程黑客也有办法伪装成局域网的登录用户，所以都应该关掉。Print\$: 这是放打印机驱动程序的目录，也是个很危险的入口。Netlogon: 这是处理域登录请求的共享。如果你的机器为主域控制器，域内有其他机器要登录进来，就不要删除它，否则照样可以删除。用“服务器管理器”|“共享目录”|“停止共享”。
- 将ASP的目录集中管理，为ASP的程序目录设置详尽的访问权限。
- 把winnt下的sam._文件改名,实践证明这个可能泄露密码的文件可以删除不要。
- 对于已知的NT安全漏洞，都应该在自己的机器上做测试检查。并及时安装补丁程序。
- 有必要的情况下采用IIS4.0提供的SSL安全通信机制来防止数据在网上被截获。

第2章 ASP的内嵌对象

本章要点：理解ASP的内嵌对象并会使用它们。

通过上面的介绍，我们对ASP已经有了一个基本的了解，如果你想编写ASP应用程序首先应该掌握一种脚本语言，如VBScript，并且熟练掌握ASP的各种内嵌对象和ActiveX组件。这些对象以及组件都可以用来拓展ASP应用程序的功能，实际上，只要掌握了内嵌对象和ActiveX组件，你就可以编写功能强大的ASP应用程序。

一个对象是典型的具有方法、属性或者集合的东西，其中对象的方法决定了我们可以用这个对象做什么事情。对象的属性可以读取，它描述对象状态或者设置对象状态。对象的集合包含了很多和对象有关系的键和值的配对。

举一个日常的例子，本书是一个对象，这个对象包含的方法决定了你可以怎样处理它（即使用它），比如说，去读它，把它送人作为礼物，用它当作敲门工具，只要你愿意，甚至可以把它撕得粉碎。对象的属性即这本书的页数，作者是谁。对象的集合包含了许多键和值的配对，对书而言，每一页的页码就是键，那么值就是对应于页码的这页的内容。

ASP的内嵌对象和ActiveX组件十分类似，不过，当你使用ASP时，两者之间还是存在着明显的差异。首先，一个组件可能包含不只一个对象，其次，在你使用组件之前，将要明确地创建一个实例。

下面是对每一种内嵌对象的快速浏览：

- **Application对象**：Application对象用来存储一个应用中所有用户共享的信息。例如，可以利用Application对象在站点的不同用户间传递信息。
- **Request对象**：Request对象可以用来访问所有从浏览器到服务器间的信息，因此，可以利用Request对象来接受用户在HTML页的窗体中的信息。
- **Response对象**：Response对象用来将信息发送回浏览器。可以利用Response对象将脚本语言结果输出到浏览器上。
- **Server对象**：Server对象提供你运用许多服务器端的应用函数。例如，你可以利用Server对象来控制脚本语言在超过时限前的运行时间。也可以利用Server对象来创建其他对象的实例。
- **Session对象**：Session对象用来存储一些普通用户在滞留期间的信息，可以用Session对象来存储一个用户在访问你的站点时滞留时间。
- **ObjectContext对象**：ObjectContext对象可以用来控制ASP的执行。这种执行过程由Microsoft Transaction Server(MTS)来进行管理。

内嵌对象不同于正常的对象。在利用内嵌对象的脚本时，不需要首先创建一个它的实例。在整个网站应用中内嵌对象的所有方法、集合以及属性都是自动可访问的。

2.1 Response和Request 对象

Response 对象可以输出信息到客户端。包括直接发送信息给浏览器、重定向浏览器到另一个 URL 或设置 cookie 的值。下面例子演示了该对象的使用方法：

```
<html>
<head><title>Buffer Example</title></head>
<body>
<%
FOR i=1 to 500
    Response.Write(i&"<BR>")
NEXT
%>
</body>
</html>
```

在ASP中引用对象的方法的语法是“对象名.方法名”，如上面脚本中用的Response.Write(i&"
")语句，其含义为引用Response对象的Write方法在屏幕上显示从1~500，每一句命令执行后结果都立即显示，你可以实时地看到执行结果。

Response对象的Write方法将指定的字符串或表达式的结果写到当前的 HTTP 输出。

方法就是嵌入到对象定义中的程序代码，它定义对象怎样去处理信息。使用嵌入的方法，对象便知道如何去执行任务而不用提供额外的指令。

在某些场合下，你也可以利用缓冲区来延缓执行过程，当你利用缓冲区时，直到整个ASP执行结束才会将结果输出到浏览器上。下面是利用Response对象的Buffer属性修改后的例子：

```
<% Response.Buffer=True %>
<HTML>
<HEAD><TITLE>Buffer Example</TITLE></HEAD>
<body>
<%
FOR i=1 to 500
    Response.Write(i&"<BR>")
NEXT
%>
</body>
</html>
```

在ASP中设置对象属性的值的方法是“对象名.属性名=”，如上面脚本中的第一行用<% Response.Buffer=True %>语句将Response对象的Buffer的属性设为True。这也是和前面那个脚本仅有的一个区别。那么这页执行的时候，这个主页的所有内容会同时显示在浏览器上，这个主页会存在缓存区中直到脚本执行结束。

任何对Buffer属性进行修改的语句都必须在<HTML>语句和ASP脚本输出语句的前面，如果在<HTML>语句或者脚本输出后面修改Buffer属性，那么你的程序就会出错。

利用缓存程序可以根据某种条件来显示不同的主页，下面是随机显示的一个例子：

```
<% Response.Buffer=True %>
```

```

<HTML>
<HEAD><Title>第一页</title></head>
<Body>
这是第一页。
</body>
</html>
<%
Randomize
注释：初始化随机数生成器。
IF Int(2*rnd)=1 THEN Response.End
Response.Clear
%>
<HTML>
<HEAD><TITLE>第二页</title></head>
<body>
这是第二页。
</body>
</html>

```

在这个例子中，使用了Response对象的两个新的方法：End方法和Clear方法。End方法会立即停止ASP的执行和相应结果。你在执行End方法时不需要考虑是否进行了缓存输出。在这个例子中，End方法被用来防止在显示第一页时第二页也显示出来。

Clear方法是用来在不将缓存中的内容输出的前提下清空当前页的缓存，仅当使用了缓存输出的时候你才可以利用Clear方法，在这里面，Clear方法是用来防止显示第二个页面时第一个页面也显示出来。它把第一个页面从B缓存中清空了。

Response对象中在缓存输出中还用到了另一个方法是Flush，该方法可以将缓存中的内容立即显示出来。该方法有一点和Clear方法一样，它在脚本前面没有将Buffer属性设置为True时会出错。和End方法不同的是，该方法调用后，该ASP可继续执行。

一般情况下，你并不需要缓存输出一个ASP，在大的HTML主页或者运行较长的脚本中利用缓存时，用户的浏览器可能长时间没有反应，这通常会使得用户感到迷惑。

如果希望根据条件显示不同主页内容，完全可以简单地利用VB脚本来做判断，例如，下面这个例子实现了上面例子的同样功能但没有使用缓存。

```

<%
Randomize
' 初始化随机数生成器。
IF INT(2*RND)=1 THEN
%>
<HTML>
<HEAD><TITLE>第一页</TITLE></HEAD>
<BODY>
这是第一页。
</BODY>
</HTML>
<% ELSE %>
<HTML>

```

```
<HEAD><TITLE>第二页</TITLE></HEAD>
<BODY>
这是第二页。
</body>
</html>
<% END IF %>
```

只有一种场合下使用缓存是必须的，那就是在主页内容已经确定后，你却需要改变主页的Header。这时除了将Buffer属性设为True外别无选择。

浏览器的请求和服务端的响应都包含头信息，头信息提供了有关请求和响应的附加信息，也包括了浏览器生成请求和服务端提供响应的过程信息。

ASP包含了相当多的集合和方法来维护Headers。一部分方法对应于特定的任务，例如提供了站点内容分级以及站点限期时间等功能；其他的方法和集合用于一般的Headers维护。以下部分详细介绍如何使用这些集合方法来影响Headers。

当浏览器在服务端接收到一个主页时，这个请求就包括了相当数量的Headers，你可以利用Request对象的ServerVariable集合来接收这些Headers，ServerVariable集合既包括了Headers也包括了服务端的一些其他信息。

集合存储了相互有关系的字符串、数字、对象和其他值。集合与数组非常相近，但它在存储或取出项目时会自动扩展与搜索。另外，集合被修改后，项目的位置将会移动。可以通过集合中项目的名称、索引访问项目，也可以遍历访问项目。

下面的例子将ServerVariable集合的所有内容名称显示出来。

```
<HTML>
<HEAD><Title>服务端变量</Title></HEAD>
<BODY>
<%
FOR Each name IN Request.ServerVariables
Response.Write("<p><b>"&name"&"</b>:")
Response.Write(Request.ServerVariables (name))
NEXT
%>
</BODY>
</HTML>
```

显示结果为：

```
ALL_HTTP:HTTP_ACCEPT:image/gif,image/x-
xbitmap,image/jpeg,image/pjpeg,/*HTTP_ACCEPT_LANGUAGE:en
HTTP_CONNECTION:Keep_Alive
HTTP_HOST:platoHTTP_USER_AGENT:Mozilla/4.01[en] (WinNT;I)HTTP_COOKIE:USERID
=HTTP_ACCEPT_CHARSET=iso-8859-1*,utf-8
ALL_RAW:Accept:image/gif,image/x-xbitmap,image/jpeg,image/pjpeg,/* Accept-
Language: en
Connection: Keep-alive Host:plato User-Agent:Mozilla/4.01[en] (WinNT;I)
Cookie:USERID=Accept-Charset=iso-8859-1,*.utf-8
APPL_MD_PATH:/LM/W3SVC/1/ROOT/test/
APPL_PHYSICAL_PATH:/D:\Inetpub\wwwroot
```

```

AUTH_PASSWORD:
AUTH_TYPE:
AUTH_USER:
CERT_COOKIE:
CERT_FLAGS:
CERT_ISSUER:
CERT_KEYSIZE:
CERT_SECRETKEYSIZE:
CERT_SERIALNUMBER:
CERT_ISSUER:
CERT_SUBJECT:
... (显示结果略)

```

可以看出, `ServerVariables` 集合包含了很多类型的信息, 下面解释常用的信息类型:

ALL_HTTP: 客户端发送的所有 HTTP 头文件。

ALL_RAW: 检索未处理表格中所有的头名称。ALL_RAW 和 ALL_HTTP 不同, ALL_HTTP 在头文件名前面放置 HTTP_prefix, 并且标题名称总是大写的。使用 ALL_RAW 时, 头名称和值只在客户端发送时才出现。

APPL_MD_PATH: 检索 ISAPI DLL 的 (WAM) Application 的元数据库路径。

APPL_PHYSICAL_PATH: 检索与元数据库路径相应的物理路径。IIS 通过将 APPL_MD_PATH 转换为物理 (目录) 路径以返回值。

AUTH_PASSWORD: 该值输入到客户端的鉴定对话中。只有使用基本鉴定时, 该变量才可用。

AUTH_TYPE: 这是用户访问受保护的脚本时, 服务器用于检验用户的验证方法。

AUTH_USER: 未被鉴定的用户名。

CERT_COOKIE: 客户端验证的唯一 ID, 以字符串方式返回。可作为整个客户端验证的签字。

CERT_FLAGS: 如有客户端验证, 则 bit0 为 1。如果客户端验证的验证人无效 (不在服务器承认的 CA 列表中), bit1 被设置为 1。

CERT_ISSUER: 用户验证中的颁布者字段 (O=MS, OU=IAS, CN=user name, C=USA)。

CERT_KEYSIZE: 安全套接字层连接关键字的位数, 如 128。

CERT_SECRETKEYSIZE: 服务器验证私人关键字的位数。如 1024。

CERT_SERIALNUMBER: 用户验证的序列号字段。

CERT_SERVER_ISSUER: 服务器验证的颁发者字段。

CERT_SERVER_SUBJECT: 服务器验证的主字段。

CERT_SUBJECT: 客户端验证的主字段。

CONTENT_LENGTH: 客户端发出内容的长度。

CONTENT_TYPE: 内容的数据类型。同附加信息的查询一起使用, 如 HTTP 查询 GET、POST 和 PUT。

GATEWAY_INTERFACE: 服务器使用的 CGI 规格的修订, 格式为 CGI/revision。