

## ASP 提速技巧

技巧之一：提高使用 Request 集合的效率

访问一个 ASP 集合来提取一个值是费时的、占用计算资源的过程。因为这个操作包含了一系列对相关集合的搜索，这比访问

一个局部变量要慢得多。因此，如果打算在页面中多次使用 Request 集合中的一个值，应该考虑将其存贮为一个局部变量。

例如将代码写成下面的形式以加快脚本引擎处理速度：

```
strTitle=Request.Form("Title")
strFirstName=Request.Form("FirstName")
strLastName=Request.Form("LastName")
If Len(strTitle) Then strTitle=strTitle & " "
If strFirstName="" Then strFullName=strTitle & " " & strLastName
Elseif Len(strFirstName)=1 Then
strFullName=strTitle & strFirstName & ". " & strLastName
Else
strFullName=strTitle & strFirstName & " " & strLastName
End If
```

技巧之二：直接访问适当的集合

如果不是别无选择，否则不要使用 `strPage=Request("page")` 的这样的形式来获取参数，因为这将按顺序搜索全部的集合—

`QueryString`、`Form`、`Cookies`、`ClientCertificate`、`ServerVariable` 直到发现第一个匹配值的名称。这样做比直接访问适

当的集合效率低，并且是不安全的，除非能绝对保证这个值不会出现在另外一个集合中。

例如，可能希望搜索满足客户请求的 WEB 服务器名称，这通过出现在每个查询中的 `Request.ServerVariables` 集合中寻找

“`SERVER_NAME`”来实现。然而，假如其他的集合也包含名为“`SERVER_NAME`”的值（键名不区分大小写），当使用 `Request` (“`server_Name`”)时，就会得到错误的结果。总而言之，应尽可能直接访问适当的集合。

技巧之三：在费时操作前使用 `Response.IsClientConnected` 属性

使用 `Response.IsClientConnected` 是观察用户是否仍连到服务器并正在载入 ASP 创建的网页的有用方式。如果用户断开连接

或停止下载，我们就不用再浪费服务器的资源创建网页，因为缓冲区内容将被 IIS 丢弃。所以，对那些需要大量时间计算或

资源使用较多的网页来说，值得在每一阶段都检查游览者是否已离线：

```
..... Code to create first part of the page
If Response.IsClientConnected Then
Response.Flush
Else
```

```
Response.End
```

```
End If
```

```
..... Code to create next part of page
```

#### 技巧之四：优化 ASP 中的 ADO 操作

通常而言，数据构成了 WEB 站点的实际内容。所以，优化 ADO 操作以加速 ASP 代码执行，十分有用：

a. 仅选择所需的列：当打开 ADO 记录集时，除非需要获得所有的列，否则不应自动地使用表名（即 SELECT \*）。使用单独

的列意味着将减少发送到服务器或从服务器取出的数据量。即使需要使用全部列，单独地命名每个列也会获得最佳的性

能，因为服务器不必再解释这些列的名字。

b. 尽可能的使用存储过程。存储过程是预先编译的程序，含有一个已经准备好的执行计划，所以比 SQL 语句执行更快。

c. 使用适当的光标和锁定模式。如果所做的全部工作只是从记录集中读取数据，并将其显示在屏幕上，那么就使用缺省的

只能前移、只读的记录集。ADO 用来维护记录和锁定的细节的工作越少，执行的性能就越高。

d. 使用对象变量。当遍历记录集时一个肯定能提高性能的方法是使用对象变量指向集合中的成员。例如：

```
While Not RsGc.EOF
```

```
Response.Write "工程名称： " & RsGc("GcMC") & "(工程代码： " & RsGc("GcCode") & ")  
"
```

```
RsGc.MoveNext
```

```
Wend
```

可以用改写为下面的代码以加快执行：

```
set GcMc=RsGc("GcMc")
```

```
set GcCode=RsGc("GcCode")
```

```
While Not rsGc.EOF Response.Write "工程名称： " & GcMc & "(工程代码： " & GcCode & ")  
" RsGc.MoveNext
```

```
Wend
```

新的代码建立了对象变量的引用，所以可以使用对象变量而不是实际的变量，这意味着脚本引擎的工作减少了，因为在集

合中进行索引的次数变少了。

#### 技巧五：不要混用脚本引擎

我们知道，ASP 页面中既可以使用 VBScript，也可以使用 JScript。但是在同一个页面上同时使用 JScript 和 VBScript 则是不

可取的。因为服务器必须实例化并尝试缓存两个（而不是一个）脚本引擎，这在一定程度上增加了系统负担。因此，从性

能上考虑，不应在同一页面中混用多种脚本引擎。

本文内容来自互联网，著作权归原作者所有。由电子零件城 (<http://www.epcity.com/>) 整理并制作成 PDF 文件，仅供个人学习之用，不得用于任何商业目的，否则后果自负。如果您认为本 PDF 文件侵犯了您的任何权利，请来信 [epcity@epcity.com](mailto:epcity@epcity.com) 通知，本站立即删除。

搜集整理：电子零件城-笨笨兔 (QQ: 154502842)      2004-03-28