

学校的理想装备

电子图书·学校专集

校园网上的最佳资源

在web网页上使用

VBScript 和JScript



在 Web 网页上使用 VBScript 和 JScript

综述：本文讨论了 VBScript 和 JScript 之间的区别，以及它们的使用场合。本文还讨论了：

- Visual InterDev 中的 VBScript 和 JScript
- 内联服务器脚本和<Script>单元
- 脚本执行顺序

如果你要编写一个 Web 网页就必须编写一些脚本。幸运的是，你可以从几种脚本编写语言中选择一个。不幸的是，有这么多种语言可以选择，到底选哪一种呢？你应该怎么做？

或者，如果你获得了一个网页，它的脚本是用一种你不怎么使用的脚本语言编写的，你怎样才能够用这个脚本来增加一些新的功能呢？

在本文中，我将讨论到这些问题。而首先，我会指导你如何用微软公司所提供的这两种脚本编写语言来进行工作。

Microsoft® Visual Basic®, Scripting Edition (VBScript).

JScript®,它是 ECMAScript 的 Microsoft 版本，是一组包括 JScript 和 JavaScript 脚本编写语言的标准定义。

当然，还有其他的脚本编写语言，如 Perl 和 Rexx 等。不过这里我着重介绍两种语言，微软公司在 Internet Explorer 和 Internet Information Server (IIS)中为它们提供了 ActiveX® Scripting Language (ActiveX 脚本编辑语言)引擎。

我还要讨论你何时需要在同一个网页中使用这两种语言(如你所见，这是完全可行的)。为了对你有帮助，我列出了一系列有关的小问题，这些问题在你工作时必须牢记于心。

为了清楚地论述这个问题，我将从以下几个方面着手：

- VBScript 和 JScript 之间的区别
- 如何在 Visual InterDev 中确定脚本语言？
- 在同一个网页中应用两种语言
- 更新脚本编辑语言
- VBScript 和 JScript 之间的区别

VBScript 和 JScript 之间最大的区别是它们的结构。VBScript 是 Visual Basic 的子集，如果你曾经用过 Visual Basic 或者是 Visual Basic for Applications (VBA)，你就会觉得非常熟悉。不过它们并不是完全一样的，因为 VBScript 是特意在浏览器中进行工作而设计的，它不包括一些在脚本这个范围以外的特性，如文件访问和打印等等。

而另一方面，JScript 从一组编程语言如 C, C++, 以及 Java 等之中脱离出来的。如果你以前曾经用过 C 或者是 Java，那么 JScript 的结构你会觉得非常熟悉。即便你不知道这些语言也没有关系，JScript 的结构一点也不难——它像 C 和 Java 一样方便而没有它们那么复杂。

注意：不管怎样，JScript 和 Java 是完全不同的两种语言。Java 是一种对于网页应用程序和非网页应用程序都可以使用的完全成熟的开发语言。而 JScript 是一种主要用于脚本编写的脚本语言。

JScript 和 VBScript 的任务是相同的——都是让你能够编写自己的 Web 网页。每种语言都支持一些别的语言不支持的特性，不过，这些差异还不足以自动划分两种语言的使用范围。

更有可能的是，你会基于一些其他因素来选择脚本语言，最大的原因也是最实际的原因：你将要使用的平台能够支持你所选择的语言吗？你编写的是客户端还是服务器？如果是服务器，你的用户会使用什么浏览器和服务器软件？

下面的文章中会详细介绍这两种脚本编写语言，并且会就在特殊的环境中选择什么样的脚本语言给出一点指导意见。

什么时候选择 VBScript ？

VBScript 主要是用在微软公司的各种平台上。因此如果你想用 VBScript 来进行编写，你必须使用下列几种脚本之一：

服务器脚本。如果你想要用 VBScript 来编写服务器脚本（如在 ASP 网页中），这个服务器必须是 Microsoft Internet Information Server (IIS) 或者是其他的服务器。

客户端脚本。如果你想在浏览器脚本中用到 VBScript，这个浏览器必须是 Internet Explorer 的一个版本。因为其他流行的浏览器可能不支持 VBScript。

如果你使用的是 Visual InterDev，那么你应该已经在使用 IIS 了（或是其他相同的服务器），因而用 VBScript 来编写服务器脚本不是问题。如果你使用的不是 IIS，你要搞清楚该服务器是不是能够支持 VBScript。

然而，如果你要建立的是一个公共 Web 站点，你无法控制其他人使用的浏览器，那么 Internet Explorer 只能用于客户端脚本这个限制就成问题了。不过，如果你是在浏览器已知的情况下用 VBScript 编写客户端脚本，这还是可行的。

如果你的目标平台能够支持 VBScript，考虑一下下列因素再决定是否用 VBScript 来编写脚本：

错误控制。在 VBScript 中有 On Error Resume Next 来对特殊情况给出说明。如果你编写的是服务器脚本，错误控制是非常重要的，因为脚本的运行是无人照顾的。

格式化。VBScript 有这样的功能，能够轻易地把数据，数字和货币数据格式化。

在 Internet Explorer 中事件控制很方便，就像在 Visual Basic 中一样。在 VBScript 中也能够建立一个隐含的事件处理程序，这只需要用对象__事件结构形式来定义一项功能就行了，比如说 Button1_onclick。不过在 Internet Explorer 中，JScript 没有这项功能（但在其他的 JScript 主机上支持这项功能）。

什么时候选择 JScript ？

几乎所有允许编写脚本的浏览器都支持 JScript。因此，如果你是为一个公共 Web 站点编写一个广泛散发的应用程序，用 JScript 来编写客户端脚本是很自然的选择。你还会发现，因为 JScript 应用得如此广泛（还有与它类似的 JavaScript），是一种标准的脚本语言，你可以在书店里，也可以在网上找到很多有关的资源。

如果你想用 JScript 来编写服务器脚本，那么你应该检查一下你的服务器是否支持它。微软公司的 IIS 允许你用 JScript 来编写 ASP 网页的脚本，但这并不是所有的服务器都支持的。

影响到你做决定的其他因素还有：

动态执行。JScript 一个非常有力的特性是：它允许你在你的脚本内建立和执行脚本或者动态地进行评价。简而言之，在你的脚本中能够编写脚本。当你用 DHTML 进行工作时，这个特性是非常有用的，因为这样你就能够动态的操作 DHTML 的文档模型了。

面向对象。JScript 使用的是一种基于原型的对象结构，它能够允许你在脚本中定义对象。你能够往对象的原型中加入方法和特性，扩展内置对象和定制对象。

使用 JScript 时的一个警告：JScript 是区分大小写的，如果你仅仅使用过 Visual Basic

或 VBScript 也许对此难以适应。

如果你是初学 JScript，你还要知道如何用 JScript 来建立事件处理程序。在 Internet Explorer 中，JScript 会要求你链接到一个对象上，而这个对象又是与事件处理程序相连的。通常，只要在鉴别事件并且有处理程序功能的对象的标签栏中指明其属性就能够完成。如下例所示：

在其他的主机上，你可以使用隐含的事件绑定，不过它要求的结构与 VBScript 中稍稍有所不同。在 JScript 中约定的隐含处理程序是：

```
<SCRIPT LANGUAGE="JavaScript">
function btnSave::onclick(){
// script here
}
</SCRIPT>
```

值得注意的是，在这两个例子中 LANGUAGE 这项属性是设置为“JavaScript”的。Internet Explorer 对于“JScript”和“JavaScript”这两个名字都认识，而许多其他的浏览器却只认识“JavaScript”，不认识“JScript”。因此，为了保险起见，如果你不能确定该网页是用 Internet Explorer 来浏览，你最好用“JavaScript”。

要了解更多在 JScript 中建立事件处理程序的情况，请查看位于 MSDN Library Online 中的 Microsoft Visual InterDev™文档。（要访问 MSDN Library Online 必须有 MSDN Online Membership 资格，这一资格是免费的，不过要简单的一次性的注册一下。）

如何在 Visual InterDev 中明确所用的脚本编写语言？

不管你用什么语言来编写脚本，你必须明确的指出来，这样主机才知道用什么脚本引擎来处理你的脚本。到底用什么语言来编写，取决于你是用内联服务器脚本来进行工作还是用<SCRIPT>块来工作（不管是客户端脚本还是服务器脚本），同时也取决于你所使用的 Visual InterDev 的版本。

为内联服务器脚本选择脚本语言

如果你使用的是 IIS 和 ASP 网页，你可以编写两类服务器脚本：内联脚本（在<% %>分隔符中的脚本）和在<SCRIPT>单元中的脚本。一般来说，用内联脚本你可以把结果插入到网页中的 HTML 中。相反，在<SCRIPT>单元中的脚本是用来建立从内联脚本中调用的功能和子程序的。

注意：要建立一个在<SCRIPT>单元中的脚本并让它运行起来像内联脚本也是可能的。也就是说，脚本所陈述的不是一项功能或者某个子程序的一部分。不过我们并不推荐这种做法。要知道更多的细节请看本文后面部分的“服务器脚本执行顺序”。

对于服务器和客户端脚本来说，为<SCRIPT>单元选择语言的方法是一样的。要知道详细情况请见本文后面部分的“为一个<SCRIPT>单元设置语言”。

在一个网页中所有的内联脚本要用同一种语言来编写。当你建立了一个新的 ASP 网页后，网页中会自动包含一个@指示，它指明了内联脚本的脚本语言。@指示可以像这样：

```
<%@ Language="VBScript" %>
```

@指示一定是网页中的第一行内联脚本命令。如果没有这个指示，IIS 会缺省地指定用 VBScript 来作为内联脚本语言。

改变网页的内联脚本语言

在 Visual InterDev 1.0 中，可以对网页进行编辑把指示改变一下，设置一种新的语言，如“JavaScript.”等。

在 Visual InterDev 6.0 中，右键点击网页后就会出现该网页的 Properties 对话框。然后在 Default 脚本语言下，选择一种新语言。经过应用之后，编辑器会在@指示中更新为相应的脚本语言。当然，如果你愿意，你也可以自己手动编辑@指示。

你还能够为所有的新 ASP 网页更换缺省值。这个操作不能够改变现在的网页，不过它对你从现在起建立的所有网页都会有用。

为内联脚本更换缺省语言

在 Visual InterDev 1.0 中，从工具菜单中选择 Options 选项，然后再选择 HTML 一栏。在 Active Server Pages 的缺省脚本语言处选择一种语言。

在 Visual InterDev 6.0 中，右键点击 Project Explorer 中的项目名字，然后选择 Properties。选择 Editor Defaults 一栏，然后在缺省脚本语言处选择一个新的缺省值。

为<SCRIPT>单元设置语言

与内联服务器脚本不一样的是，你在<SCRIPT>单元中所使用的脚本语言在同一个网页中可以有很多种。如果某一块是用 VBScript 来编写的，那么另一块也可以用 JScript。（记住，不是所有的平台都支持 VBScript。）

注意：在同一个网页中混合几种语言有很多注意事项，详细内容请见本文后面部分的“在同一个网页中用几种不同的语言来工作”。

为一个<SCRIPT>单元指定语言

在<SCRIPT>标记中包含了一个 LANGUAGE 单元，它指明了你所要使用的语言。例如，在下面这个<SCRIPT>标记中就指明了该<SCRIPT>单元中的所有脚本都要用 JScript 编写：

```
<SCRIPT LANGUAGE="JavaScript">
function test(){
    alert("testing");
}
</SCRIPT>
```

注意：在 IIS 的服务器脚本中要求有 LANGUAGE 属性。而在客户端脚本中它只是一个选项。如果 LANGUAGE 属性没有了，处理程序会使用缺省的脚本语言。在 Internet Explorer 的客户端脚本中，缺省值通常是第一个<SCRIPT>单元的脚本语言，或者如果还没有指明任何语言，那就是 JScript 了。我们建议你最好不要依赖于缺省值，最好是为<SCRIPT>单元明确地设置一种脚本语言。

在一个<SCRIPT>单元中，你不能把 VBScript 和 JScript 混合起来使用。如果你想同时使用它们，在你的网页中你必须至少有两个<SCRIPT>单元。

Visual InterDev 1.0 和 Visual InterDev 6.0 都提供了能够自动生成脚本的工具。在 Visual InterDev 1.0 中有 Script Wizard 可以帮助你。而在 Visual InterDev 6.0 中有下列产生脚本的工具：

- Script Block 命令。只要你在网页中点击右键，出现的菜单中就会有这个命令。它能够在你点中的那一点插入一个<SCRIPT>单元的框架。
- Script Outline 命令。当你双击事件的名字时它就会建立一个事件控制器的框架。在你的设置的基础上，不同的脚本生成工具会决定你所要用的语言。你可以在个别网页中为脚本工具而改变缺省的语言种类。

在 Visual InterDev 1.0 中为 Script Wizard 设置语言

从 Tools 菜单中选择 Options 选项，然后选择 HTML 栏。在 Script Wizard 的缺省脚本语言中选择一种语言。

在 Visual InterDev 6.0 中为脚本生成工具设置一个网页中的语言

首先，右键点击网页，网页的 Properties 对话框出现。然后在缺省脚本语言下选择一种新的语言。你能够为服务器脚本和客户端脚本设置同一种语言。

注意：为产生服务器脚本块而作出的设置一样可用于为内联脚本设置语言。如果你想在内联脚本中使用一种语言，而在用工具产生的服务器脚本块中使用另外一种语言，你只能手工地进行语言改换。

你还可以改变你的项目的缺省值，这样你建立起来的所有网页都会有与脚本工具一样的缺省语言设置。

在 Visual InterDev 6.0 项目中改变脚本生成工具的缺省语言设置

在 Project Explorer 中右键点击项目的名字，然后选择 Properties。选择 Editor Defaults，在缺省脚本语言处选择一个新的缺省值。

注意：已产生的服务器脚本块语言的设置方法同样可以用于为内联脚本设置语言。如果你想在内联脚本中使用一种语言，而在用工具产生的服务器脚本块中使用另外一种语言，你只能手工地进行语言改换。

想知道 Visual InterDev 6.0 中脚本生成工具的更多信息，请查看位于 MSDN Library Online 中的 "Choosing a Scripting Language" 和 Visual InterDev 6.0 文档中的 "Handling Events with HTML Elements"。

在同一个网页中同时使用几种语言

我想我已经说得很清楚了，在同一个网页中可以使用不同的脚本编写语言。让再回顾一下：为什么你想这么做？以下是几个原因：

第一个原因是：你只是想在一个程序中利用某种语言的优点，而脚本的其他部分还是用另外一种语言来编写。

第二个原因是：可能你所面对的现有编码是用你不熟悉的语言来编写的。例如，设计时间控制是用 JScript 编写的，而你熟悉的语言却是 VBScript。在大部分情况下，你可以在同一个网页中混合几种脚本语言并且它们相互之间能够配合得很好。用一种语言编写的脚本可以调用用另外一种语言编写的程序并且能够共享全局变量。不过有时候还是会出现一点小问题，现在我将要就此说明一下。

在一个 VBScript 调用中的括号

当你从 VBScript 中调用一个 JScript 函数时，注意在调用过程中要有括号，即使这项函数不要求设置参数也需如此。例如，一个调用应该像这样：

```
retVal = callJSFunction()
```

如果你忘记了加括号，那么这个函数所返回的值就不是你所希望的值，而是一个包含了函数本身的目标。

服务器脚本的执行顺序

内联服务器脚本按顺序运行，从头到尾。你可以在一个服务器脚本中定义一个可调用的程序（函数或者是子程序），这样的话，如果需要就可以随时调用它们。

所有的内联脚本必须用同一种语言来编写，也就是在网页前面的 @ 指示中所指明的脚本语言。因此，在一个内联脚本中你不能把几种脚本语言混合起来。

你也许会说“等一等”。在理论上，把内联脚本放到 <SCRIPT> 单元中是绝对可能的——也就是说只要在单元中的脚本不是一个函数或者一个子程序就可以了。就像下例一样：

```
<% Response.Write("Some inline script<BR>")%>
<SCRIPT LANGUAGE="VBScript" RUNAT="Server">
    Response.Write("Script in a SCRIPT element<BR>")
</SCRIPT>
```

是的，你可以这么做。但是，这样你就会被 IIS ASP 处理程序的执行顺序控制。例如，

如果你建立了服务器脚本并且在 IIS 4.0 中运行它，你会发现执行顺序是这样的：

- 采用非缺省语言的<SCRIPT>单元脚本
- 内联脚本
- 采用缺省语言的<SCRIPT>单元脚本

以上的顺序在很大程度上依赖于 ActiveX 语言引擎的载入顺序，同时 IIS 中的 ASP 处理程序可能在将来会有所改变，因此，我们很不希望你按照这个顺序进行工作，不管你是只把<SCRIPT>单元用于函数或者子程序，还是把它仅仅用于执行顺序无关紧要的编码。

脚本块的顺序

当你把几种语言混合使用时，<SCRIPT>块在网页中出现的顺序非常重要，有时候会影响到它们是否能正常工作。看看下面这个简单的例子，它是在一个内联的 VBScript 脚本中调用一个用 JScript 编写的函数：

```
<SCRIPT LANGUAGE="VBScript">
    ' Calls a JScript function
    aNumber = 2
    doubledNumber = doubleMe(aNumber)
    document.write("The answer is " & doubledNumber)
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
function doubleMe(aNumber){
    return aNumber * 2;
}
</SCRIPT>
```

它不会起任何作用。说得更明白一点就是，文档写入说明会往网页中写入一个空的字符串。为什么？很简单，在处理 VBScript 块时，下面的 JScript <SCRIPT>块没有被读出来，没有进行语法分析，因此网页不能使用它。而当浏览器处理网页中的脚本块时，它是从上往下进行的。

在这个例子中，只需要调换一下脚本块的顺序就能够解决这个问题了。事实上，这类问题并不那么常见，在大多数情况下，<SCRIPT>中包含的函数和子程序只有当整个网页被全部下载下来并且所有的单元都可以使用之后才能够被调用。不过，你还是要记住，网页是按照一定的顺序来处理的，不同语言的<SCRIPT>块是分别处理的。

区分大小写以及命名习惯

JScript 能够区别大写和小写：在每个 JScript 的关键字，还有 JScript 姓名空间的所有部分中，你可以把大写和小写字母恰当的结合起来使用。这其中包括了你使用的对象模型中所有的关键字。例如，当你使用 ASP Response 对象时，必须正确地使用大写或小写字母（如 Response.Write），否则 JScript 会坚决不予承认。同样的，如果你为 DHTML onclick 事件编写一个处理程序时，如果用在 Visual Basic 中经常使用的形式 JScript 可能就不会承认。

VBScript 当然是不区分大写和小写字母的。在某种程度上，这种不计较大小写的忽视甚至在你用 VBScript 对 JScript 编写的单元进行访问时都适用。下面几个小例子中给出了两个脚本。第一个是用 JScript 编写的，其中还包括了一个全局变量（ctr）和一个小小的函数（doubleMe）。第二个脚本是用 VBScript 编写的，它是一个按钮处理程序，它能够调用 JScript 函数，并且报告全局变量的值。

```
<SCRIPT LANGUAGE=javascript>var ctr;function doubleMe(aNumber){ // Initialize
value of global counter    if(parseInt(ctr)){
    ctr = ctr + 1;}
    else{
        ctr = 1;
    }
    return aNumber * 2;
}
</SCRIPT>
```

```
<SCRIPT LANGUAGE="vbscript">
Function btn1_onclick()
    numberToDouble = 12
    alert("The doubled number = " & DoubleMe(numberToDouble))
    alert("You have doubled the number " & Ctr & " times.")
End function
</SCRIPT>
```

在 VBScript 脚本中，很明显，我在提到 doubleMe 函数和 ctr 变量时，没有正确使用字母的大小写，但是程序依然运行良好。

这是一个好消息。而坏消息就是，在下面这些情况下你一定要注意字母的大小写：

如果网页中的 JScript 脚本所包含的单元只能够通过字母的大写和小写来区分，例如，一个叫做 Timer() 的函数和一个叫做 timer 的全局变量等等。记住，尽管这种编码方式不怎么好，但是有时候在你的网页中还是可能会遇到原来就存在的编码。

如果你在一个函数调用中把 JScript 单元的名字作为一个字符串参数来使用

如果 JScript 包括的单元仅仅能够用字母的大小写来区分，那么 VBScript 就会让你把它们看成一样的（也就是说，不区分大写和小写）。VBScript 只能看看不管大小写的区别，它们是不是一样，因此它识别出的只有一个 JScript 单元。换一句话说，如果 VBScript 首先找到的是叫做 Timer() 的函数，那么它就无法找到叫做 timer 的全局变量了，反之也是一样。即便你在 VBScript 编码中十分注意的把它们做了区别结果也是一样。因为 VBScript 在处理这些名字的时候认为它们只有一种写法。不幸的是，这里没有其它的解决方法，除非在 JScript 的代码中不利用这些大小写之间的微小差别来区分不同的函数名称。

除了在某些特殊的情况下，把单元名作为参数使用这种问题并不常见。特殊情况之一是在 Visual InterDev 6.0 中的脚本对象模型中。下面我会把它作为一个例子来使用，你可以把它看作是相同情况下的处理准则。

Visual InterDev 6.0 的脚本对象模型都是用 JScript 来编写的。你也可以用 VBScript 来编写自己的编码，它能够与用 VBScript 区分大小写方式编写的脚本对象模型一起使用。有时候，你不能够直接调用某一方法，除非你把这个方法的名称传递到一个 JScript 函数中，而由函数来为你调用这个方法。

一个典型的例子就是警告方法，你可以用它把一个方法绑定到一个特殊的事件上。在下面这个例子中，一个按钮的 onmouseover 事件被绑定到 changecaption() 事件处理程序上。


```
<SCRIPT LANGUAGE="VBScript">Function setAdviseMethods()  
    Btn1.advise("onmouseover", "changecaption()")  
End Function  
</SCRIPT>
```

因为你把这些名字作为字符串传送到一个 JScript 函数中，所以你能够得到正确的大小写表示。稍后，JScript 会自动用它的区分大小写的规则来重新评估这些名字，如果大小写不正确，它会找不到它们。

从 VBScript 往 JScript 中传送数组

有时候一个 VBScript 程序会传送一个数组作为它的一个参数，或者作为它的返回值。你可以从 JScript 中调用这个程序，不过必须把 VBScript 数组转换成为一个可用的 JScript 数组。你只要在 JScript 函数中建立一个 VBArray 对象并且用 toArray 把它转换成为一个 JScript 数组就可以了。

注意：JScript 不支持多维数组，如果原来的 VBScript 数组是多维的，经过 toArray 的转换后它会成为一个一维的 JScript 数组。

在下面的例子中有一个 VBScript 脚本，它建立了一个数组，还有一个 JScript 脚本，它将说明如何才能获得并使用这个数组：

```
<SCRIPT LANGUAGE="VBSCRIPT">  
Function makeArrayVB()  
    ' Creates a VBScript array  
    dim anArray(1,1)  
    anArray(0,0) = "0,0"  
    anArray(0,1) = "0,1"  
    anArray(1,0) = "1,0"  
    anArray(1,1) = "1,1"  
    makeArrayVB = anArray  
End Function  
  
<SCRIPT LANGUAGE="JavaScript">  
// Accesses a VBScript array within a JScript script  
function getVBArray(){  
    var arrayObj;  
    var jsArray;  
    arrayObj = makeArrayVB();  
    jsArray = VBArray(arrayObj).toArray();  
    alert("VBScript array length = " + jsArray.length);  
    // Displays the contents of the array  
    for(i=1;i<=jsArray.length;i++){  
        alert(jsArray[i-1]);  
    }  
}  
</SCRIPT>
```

不过，在目前的 VBScript (4.0) 中，反过来是无法做到的。也就是说你不能把一个

JScript 数组转换为 VBScript 数组。如果你遇到的是一个 JScript 程序，要转换数组，你可以选择以下几种做法：

- 用 JScript 编写调用程序
- 用 VBScript 重新编写 JScript 程序
- 如果可能，把 JScript 数组转换成一种不同的结构，如转换成一个限制的字符串等，这样 VBScript 就能够使用了。例如，你可以用 toString 函数把一个数组转换成一个用逗号分隔的，无限制的字符串。这样，在 VBScript 中你就能够用 Split 函数把它分成不同的单元。很明显这种方法在很多时候并不管用，不过有时又的确有效。

更新脚本语言

VBScript 和 JScript 都不是一成不变的语言。两种语言的引擎都定时进行更新，增加一些新的特性，解决已经发现的问题。因为两种语言实际上都是作为 ActiveX 组件来应用的，所以每当你更新任何控制器时，你就可以同时更新它们。当你安装了新的语言引擎之后，任何需要调用引擎的应用程序会在它们运行某些脚本时调用这个新引擎——这些引擎包括 Internet Explorer 或 Visual InterDev 等。

如果要了解语言引擎的情况以及如何更新的方法，还有脚本的一般知识，请访问 Microsoft Scripting Web 站点，网址是：<http://msdn.microsoft.com/scripting/>。同时，祝你编写脚本时一切顺利！

