第22章　使用Shockwave和因特网

Shockwave于1995年问世时，它只是Netscape Navigator的外挂程序，它使浏览器能够播放经压缩的Director影片。从那时起，“Shockwave”这个词的内涵已经扩展，几乎任何与Director和因特网相关的事物都与它相关。

同时，Director也借鉴了Shockwave的各个方面的内容，用于其创作环境和放映机。在早些时候，只有在Web浏览器里运行的影片才具有访问因特网信息或控制浏览器位置等能力。而现在，用Director制作的任何作品都具有这样的能力。

这意味着当我们在创作Director作品和放映机时，所有Shockwave命令都有效。当涉及到控制Web浏览器的命令时，放映机甚至可以自动地运行用户的缺省的Web浏览器。

Shockwave Lingo可以被分为四部分。第一部分是控制Web浏览器的；第二部分是在因特网上收发信息的；第三部分是与HTML页面进行通讯的；最后一部分是直接与服务器和其他客户进行通讯的。

22.1　用Lingo控制Web浏览器

gotoNetPage是命令Web浏览器进入另一个Web网页的重要命令。这个命令可以使用相对位置，或者如果某个位置是以“http://”开头，就可以用绝对位置。例如：如果用户已经位于http://clevermedia.com，下面两条命令是相同：

```
gotoNetPage("http://clevermedia.com/resources/")
gotoNetPage("resources/")
```

我们也可以通过goto NetPage使用目标帧或窗口。Web创作者很熟悉如何使用目标。最基本地，每个浏览器窗口以及窗口内的每一帧都有一个名称。一个目标标签辅助符号可以指定哪个目标应该接收gotoNetPage信号。

如果我们已使用HTML开发了自己的一套窗口或帧，就已经知道了这些目标的名称。然而，有一些名称——如_blank和_top——是为特殊目的而保留的。表22-1给出了它们的用途。

表22-1　保留的HTTP目标名称

名　称	作　用
_blank	创建一个新的、没有命名的空白窗口
_self	把新页调到当前帧。即使HTML页面使用BASE结构也有效
_parent	在帧组里向上移动一级，调用新的一页来代替当前帧及其附属内容
_top	调入到当前窗口里，代替这里的全部帧

这里有一个使用了目标的gotoNetPage命令的例子：

```
gotoNetPage("mypage.html","_blank")
gotoNetPage("http://clevermedia.com","_top")
gotoNetPage("http://clevermedia.com","mainframe")
```

如果我们计划在放映机中使用这些命令，我们可能想能更多地对浏览器进行控制。Director使用在网络参数设置中所指定的浏览器，或如果没有指定，就使用系统的缺省浏览器。

我们可以使用browserName()函数检测浏览器的路径：

```
put browserName()
-- "Macintosh HD: Netscape Communicator Folder:Netscape Communicator"
```

我们也可以使用browserName命令设置此路径：

```
browserName "Macintosh HD:Other Browser"
```

如果使用FileIO Xtra的displayOpen函数，我们也可以让用户选择浏览器应用程序，并用其结果使用browserName命令。

如果，由于某种原因，我们想取消放映机的运行某个浏览器的能力，可以把 browserName命令用在下面这种旧的句法里：

```
browserName(#enabled, FALSE)
```

用TRUE代替FALSE，就允许运行浏览器了。

除了告诉浏览器显示那一个页面外，我们还可以用一部新影片代替现在的Shockwave影片。gotoNetMovie命令等同于网络命令go。可以直接从页面中调用网上的一部新影片来代替正在播放的影片：

```
gotoNetMovie("newmovie.dcr")
gotoNetMovie("http://clevermedia.com/newmovie.dcr")
```

可以使用相对和绝对路径名。我们甚至可以指定新影片的一帧并跳到那一帧。只要在文件名后放置#符号和帧标签即可。

```
gotoNetMovie("newmovie.dcr#intro")
```

当使用gotoNetMovie命令时，当前的影片将继续播放直到调入一部新的影片。如果我们发出另一个gotoNetMovie命令，它可以取消第一个命令，替换当前的影片。

参见第16章“控制文本”里的16.9节“使用文本文件和FileIO Xtra”，可以获得关于在硬盘上存储和读取文件的信息。

22.2 从因特网上获取文本

Shockwave Lingo使我们能够从因特网或局域网上获得更多种形式的素材。对于位图来说，要想获取一个外部图像，只要指定它的位置就可以。然而，要想获取文本，就相对复杂一些了。

你也许要从因特网上的某一个文件中获取文本，并把文本数据存储在外部文件中，这样你的公司里的不使用Director人员也能够将数据更新。举例来说，这种技巧可以用于构成图表的数据。你还可以从其他来源输入文本数据，如天气预报或者小的文本数据库。

从因特网上获取文本的重要命令是getNetText。但是这个命令不能独自生效。它只是引发对网络的调用。我们必须使用一系列命令和函数来完成整个操作。

在发出getNetText命令以后，netDone函数会告诉我们何时收到文本。然后，我们可以使用netTextResult来获取文本，并把它们存储在变量或演员里。

然而，我们不能用下面的方法处理，即先使用 getNetText，然后直接将影片锁在一个repeat循环中，直到netDone返回一个TRUE。repeat循环将独占计算机，并且限制计算获取文本的能力。它会繁忙地运行repeat循环，而永远都没有时间执行网络功能。

获取文本的正确方法是先发出getNetText调用，然后让影片运行。它甚至可以在某一帧循

环，表现出暂停的状态。要知道，在某一帧循环与用 repeat命令在一个处理程序中循环是完全不同的。当在某帧循环时，影片允许执行一些网络功能，从而能够获取文本。

下面有一个例子，其中有两帧。每一帧有一个 on exitFrame剧本放置在帧剧本通道内。第一个剧本启动网络调用：

```
on exitFrame
  global gNetID
  gNetID = getNetText("http://clevermedia.com")
end
```

像我们看到那样，getNetText实际上是一个函数，返回一个与该网络函数对应的网络 ID号。由于Director一次可以执行多个网络函数，如 getNetText，因此这个ID号用于在将来引用它们。在本例中，这个数字很可能是 1，除非我们已经执行了一个网络函数。

下一帧所包含的程序将检测与变量 gNetID对应的网络函数是否已完成。如果完成，就获取文本，然后继续前进。如果没有完成，则继续在这一帧循环：

```
on exitFrame
  global gNetID

  -- check to see if text has arrived
  if netDone(gNetID) then
    -- it has, so get it
    text = netTextResult(gNetID)
    put text

    -- move the movie forward
    go to the frame + 1
  else

    -- text is not here yet, keep looping
    go to the frame
  end if
end
```

最好检测系统在获取文本时是否出现了什么问题。在我们用 netDone证实了操作已经完成后，可以立即使用netError函数。例如，可以添加下面这段程序：

```
if netError(gNetID) <> 0 then
  alert "An error occurred trying to get the text."
  halt
end if
```

很明显，“ 0 ”意味着没有错误，而其他任何数字都意味着有问题。表 22-2给出了所有的错误信息。

表22-2 netError代码

代 码	含 义
0	操作成功
4	所需要的网络 Xtra没有安装
5	错误的MOA界面，很可能与 4相同
6	错误的网址，或者可能与 4相同
20	浏览器检测到一个错误

(续)

代 码	含 义
4146	不能与远程主机建立连接
4149	由服务器提供的数据属于未知格式
4150	连接意外地提前关闭
4154	由于时间已到未能完成操作
4155	没有足够的内存来完成该事务
4156	应答请求的协议表明应答里有一个错误
4157	事务验证失败
4159	无效的网址
4164	不能创建套接字(socket)
4165	找不到所要求的对象
4166	类代理失败
4167	传送被客户机故意中断
4242	下载被netAbort停止
4836	下载由于未知的原因而停止，很可能是由于网络错误，或下载被放弃

参见第31章“ Shockwave短程序 ”里的31.4节“ 处理和显示信息 ”，可以看到一个从因特网上获取文本的例子。

22.3 发送文本

自从Shockwave于1995年问世以来，影片已经能够在因特网上发送文本了。然而，直到Director 7，还没有命令来完成这项工作。取而代之的是需要一种技巧 —— 用getNetText命令把信息发送给服务器CGI脚本。这种技巧仍旧是与服务器通讯的好方法。 Director 7也包括一个新命令，它使影片能够用与HTML页面相同的方式把信息post(发送)到服务器上。

22.3.1 用getNetText发送文本

看起来有些混淆，但我们的确可以通过获取文本来发送文本。 getNetText命令是用来从服务器上获取文本信息的。然而，通过这样做，我们也可以将信息传递给服务器。这与 Lingo函数工作的方式类似。函数能返回信息，但也能把信息用作参数。

这里有一个典型的getNetText调用。它调用Web服务器上的一个HTML页面：

```
getNetText("http://clevermedia.com/test.txt")
```

我们也可以用getNetText调用一个CGI程序。CGI程序是一种小的计算机程序，通常用 Perl语言来编写，然后放在服务器上。 Perl程序的输出结果通常为文本，如一个 HTML页面：

```
getNetText("http://clevermedia.com/cgibin/echo.cgi")
```

在这个例子中， CGI程序返回文本，就像前面调用“ test.txt ”时一样。浏览器和 Director都不关心服务器必须运行一个程序，而不是提供一个文本文件。

CGI程序所能够做的事情远远超出仅仅提供静态的文本。它们实际上可以取一些数据为参数，并使用它们。例如，它们可以把数据存储在服务器上的一个文件中。通过在 Web网址后面放置一个“ ？ ”，然后再放置一些文本，可以向CGI程序发送信息：

```
getNetText("http://clevermedia.com/cgibin/echo.cgi?gary")
```

在这个例子中，信息“ gary ”被发送给服务器。在另外一端的 Perl程序仅仅需要查找并得到它，然后用它来做一些事情。下面是一个位于服务器上的 Perl程序：

```
#!/usr/bin/perl
$invar = $ENV{'QUERY_STRING '};
print "Content-type: text/html\n\n";
print "Input: $invar <BR>\n";
```

尽管对Perl的讨论超出了本书的范围，但我们可以找到很多关于 Perl的书籍。这里给出上面这段程序的含义：

1) 第一行告诉服务器这是一个Perl程序，因而当它被调用时，服务器知道要运行 Perl(即与Director相似的程序)，然后把这个文件用作源代码。

2) 第二行获取在对服务器的调用中的问号之后的数据。在这个例子中，信息是“ gary ”。

3) 第三行开始输出。它把 Content-type: text/html这一行外加两个新的字符行放在输出流中。需要用它们来告诉服务器及 Web浏览器即将到来的输出的类型。这一行以及额外的两个新的字符行将不会出现在我们所得到的文本中。然而，其后的所有内容都将出现。

4) 最后一行是输出 Input: ，后面再跟着文本。因此结果得到的即是所发送的内容。这是一个很好的测试，它表明服务器能够获取信息，也能发送信息。在这个例子中，服务器得到了“ gary ”，对它进行处理，并把它发送回去。它还可以得到已打开的文件并存储信息。它甚至可以打开另一个文件(如一个数据库)，并用这个信息来查询其他信息。

22.3.2 用postNetText发送文本

postNetText是Director 7的新添内容，它能够执行与HTML的METHOD=POST相同的功能。使用这个命令的主要优点是它可以比 getNetText方法发送更多的信息(在大多数情况下，getNetText仅限发送4000个字符)。得到的信息的格式也与用 getNetTest所得到的信息的格式不同，而前者是许多CGI程序员所愿意使用的。

postNetText命令所需要的两个参数是CGI剧本的位置和数据。在这个例子中，数据是一个列表。这有一个例子：

```
postNetText("http://clevermedia.com/echopost.cgi", ["name": "Gary", "ID": 1])
```

这个列表应该是属性列表。每个属性与一个项目的名称相对应，每个值是这个项目的值。所有属性都应该是字符串，如果它们不是字符串， Director可以将它们翻译成字符串。

在一个postNetText调用之后，要使用一种与 getNetText函数相同的过程，即必须把该函数返回的值作为一个ID号，检测netDone，然后用netTextResult获得所返回的文本。

注释 即使没有文本返回，或我们不需要文本，也应该完成这些过程。否则，对服务器的调用永远都不能结束，而且，在网络调用结束操作之前，我们只能得到许多开放的调用。

通过Lingo使用postNetText是非常容易的。要得到一个接受和处理数据的 CGI程序却有些困难。如果我们不了解服务器的编程，我们需要同有这个能力的人一起工作。或者，一本好的关于Perl的书或一些Web方面的研究会对我们有帮助。

22.4 浏览器的使用

放置在Web网页中的Shockwave影片具有与HTML页面和Web浏览器通信的能力。它们可以从它们所属的OBJECT标签中读取信息，并且与JavaScript和VBScript对话。

22.4.1 EMBED 和OBJECT标签参数

在Netscape Navigator中显示的影片是 HTML的EMBED标签的一部分。在 Microsoft Internet Explorer中显示的影片是OBJECT标签的一部分。这两种标签在第 36章“ 送货 ”里都将详细讲述。

这两种标签看起来差异很大，但它们有很多相似之处。其中一点是，它们都使用一些额外的参数。这些参数可以用来向 Shockwave影片传递信息。

下面是一个使用额外的参数“ sw1 ”的EMBED/OBJECT标签。注意EMBED标签实际上是在OBJECT标签里面的。 Microsoft Internet Explorer使用OBJECT标签而不使用 EMBED标签，而Netscape Navigator不使用OBJECT标签，而使用EMBED标签：

```
<OBJECT classid="clsid:166B1BCA-3F9C-11CF-8075-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/director/┐
sw.cab#version=7,0,0,0" ID=Credits WIDTH=512 HEIGHT=384 ID="shock">
<PARAM NAME=src VALUE="mymovie.dcr">
<PARAM NAME=sw1 VALUE="testing!">
<EMBED SRC="mymovie.dcr" NAME="shock" WIDTH=512 HEIGHT=384 sw1="testing! ">
</OBJECT>
```

值得注意的是， sw1参数需要两次被加入到程序里：一次是为 Explorer，一次是为 Navigator。而且这是必须的。

Shockwave影片通过使用externalParamValue函数可以从这个标签中获得这个参数。此函数取一个名称(如“ sw1 ”)或一个数字为参数。可以用 externalParamName函数获得一个参数的名称，传递的值是这个参数的号码。我们也可以使用 externalParamCount()函数来取得参数的总数。

尽管在Navigator中我们可以为参数任意命名，但 Explorer要求参数名必从一个预定义的清单中选择。下面是这个清单的完整内容：

sw1，sw2，sw3，sw4，sw5， sw6， sw7， sw8， sw9，swURL，swText，swForeColor，swBackColor，swFrame， swColor， swName，swPassword，swBanner， swSound， swVolume，swPreloadTime，swAUdio，swList

这些参数名都不必用于任何特殊用途。这个集合只是 Macromedia工程师们所使用的很方便的名称，他们是根据创作者们可能会用到的参数而确定的。比如，我们可以通过“ swURL ”参数传送一个声音名，这都是允许的。

由于Explorer需要我们使用已经存在的参数名称，如果我们也能在 Navigator上坚持使用它们，这样最好。这样，我们就不需要从一个平台读取一个参数，而从另一个平台读取另一个参数。

参见第36章里的36.2节“ 制作Shockwave影片 ”，可以获得更多有关使用 Shockwave的信息。

22.4.2 与JavaScript对话

JavaScript是Netscape Navigator 的编程语言。Microsoft Internet Explorer除了使用它自己的语言VBScript外，也可以使用JavaScript。这些语言被嵌入HTML页面。它们都是很复杂的、面向对象的编程语言。要讲述这两种语言需要整整一本书，并且实际上现有的 JavaScript书籍要比Director和Lingo的书多得多。

然而，使这些语言能够与Shockwave对话的命令是十分简单。它们基本上只是给影片发送一些信息。

由于浏览器经常改变，举出一些 JavaScript与Shockwave通信的例子很困难。在写这本书的时候，Netscape Navigator 4.5和Microsoft Internet Explorer 4.0是浏览器的主要版本。然而，在你读到这本书的时候，两种浏览器都可能使用 5.0版本了。

应当把这些例子作为帮助我们编写自己的程序的向导，而不能逐字照抄。同样要记住，一些用户仍在使用旧版本的浏览器。由于浏览器的不同及缺乏有关 JavaScript的丰富资料，如果可能，应当尽量避免与Director一起使用JavaScript，或单独使用它。

要发一个信息给Shockwave，首先需要在OBJECT/EMBED标签为我们的Shockwave对象命名。在本章前面的例子中，通过在 OBJECT标签中使用 ID参数和在EMBED标签中使用NAME参数，对象被命名为“ shock ”。

现在我们需要把对象指定给一个 JavaScript变量。这一过程对于不同的浏览器是不同的。这段JavaScript程序考虑到了其中的差异，并把对象的引用变量存储为 myShock。

```
if (navigator.appName =="Netscape") {
  myMovie = document.shock;
} else {
  myMovie = shock;
}
```

现在，要发送一个消息给这个短程序，只需使用这个对象，后面加上消息就可以了。例如：

```
myMovie.GotoFrame(42)
```

这个语句向Shockwave影片发出一个命令，让它跳到第 42帧。我们可以使用 8种不同的消息类型。下面将它们全部列出。记住这些命令是用在供浏览器使用的 HTML代码中的，而不是用于Director影片的。

- Stop() —— 暂停播放影片。
- Play() —— 从当前的位置开始播放影片。
- AutoStart() —— 返回TRUE或FALSE，确定影片在被调入或在Rewind()命令之后是否播放。
- Rewind() —— 将影片回转到第1帧。
- GotoFrame(x) —— 让影片跳至第x帧。
- GotoMovie(location) —— 调入另一部影片，以代替当前的影片。
- GetCurrentFrame() —— 返回当前帧的编号。
- EvalScript(string) —— 向影片中发一个文本字符串，它将被 on EvalScript处理程序中的Lingo使用。

EvalScript是这些命令中最有用的。它可以将任何字符串送入影片。在另一端应当是 on EvalScript处理程序，下面是一个简单的例子：

```
on EvalScript text
  alert("Message From JavaScript: "&&text)
end
```

当然，使用这些文本并做一些事情的处理程序实际上更有用。我们甚至可以用 do命令让影片执行任何Lingo命令或处理程序：

```
on EvalScript text
```

```
  do text
end
```

提示 do命令得到一个字符串并运行它，好像它是被键入了消息窗口。我们可以调用处理程序，设置全局变量，甚至直接发出命令。

22.4.3 与Shockwave对话

与EvalScript处理程序相对的是externalEvent命令。它会向JavaScript发出一个字符串，以代表一个命令。这里有一个例子：

```
externalEvent ("myJavaScriptFunction (' param ') ")
```

如果用户有Netscape Navigator，这个命令只是运行被命名的JavaScript函数，并带有参数。然而，Microsoft Internet Explorer则试图给VBScript发出命令，而不是给JavaScript。如果我们知道如何使用VBScript，我们可以写一个与Netscape JavaScript功能类似的函数，甚至调用同一个JavaScript处理程序，同时也传递参数信息。这里有一段程序。它假设 OBJECT标签把短程序的ID设置为“ shock ”。

```
<script language="vbscript">
sub shock_ExternalEvent(byVal aMessage)
  call myJavaScriptFunction(aMessage)
end sub
</script>
```

22.5 Shockwave Preferences文件的使用

假设一部Shockwave影片会请求用户输入他或她的名字，然后在整个演出过程使用这个名称。如果用户再次返回此页时，影片最好能够记住用户的名字。这类事情可以在 Preferences文件中完成。

Preferences文件只是一个小的文本文件，可以存在用户的计算机中。它实际上是放置在用户的浏览器文件区域的，像这样把它与其他文件分开来，对安全是有利的。

我们可以使用setPref命令创建一个Preferences文件。例如，要存储用户的名字，可以这样做：

```
setPref("cmprefname.txt", gUserName)
```

在这个例子里，cmprefname.txt是Preferences文件的文件名，变量gUserName中的文本是该文件的内容。给这个文件一个复杂的名称的原因是这个 Preferences文件必须由所有的Shockwave文件共享。如果我们简单命名此文件为“ name.txt ”，其他的一些创作者可能也会使用“ name.txt ” 制作一个Shockwave 短程序，这个文件就会覆盖另一个文件。

使用其他创作者不常用的名称作为前缀是一个非常好的办法。如果公司的名称是CleverMedia，“ cmpref ”就是一个很不错的选择。如果我们在为“ Joe ’ s Multimedia and Burgers ”工作，我们可能想以“ jmb ”或类似的一些字母作为名称的开头。我们还应该使用“ .txt ”作为文件的扩展名，因为Windows需要这个扩展名，我们的部分用户可能使用 Windows。如果我们试图用其他扩展名，而不是“ .txt, ”，就得到一个错误信息。

Preferences文件的内容必须是字符串。如果我们需要存储许多信息，我们可能要考虑将属性列表转换为字符串，然后在读文件时，再使用 value命令将其转换回来。

我们可以用getPref命令来读取文件：

```
text = getPref("cmprefname.txt")
```

如果Preferences文件不存在，我们会获得一个VOID值。我们可以测试一下，看看用户是否曾经看过某个特定的影片。

Preferences文件还可以为一个游戏存储最高分。如果我们的影片是一个游戏并使用了记分系统，我们只要在游戏完成后写上用户的最高分，并且在当用户返回来查询时，可以显示“your best score”或类似的一些东西。这个分数决不会同其他用户的计算机上的分数相比较。家用或商业游戏厅的游戏都是这样的，因为这些计算机没有与其他计算机联网。

在Director和放映机里还可以使用getPref和setPref。这样做可以在与应用程序相同的位置处建立一个Preferences文件夹。任何Preferences文件都可以存储在这里。这个文件夹主要是用于测试，但如果正确使用它，它可以代替简单的FileIO Xtra来存储一些文本文件或信息。

参见第16章里的16.9节“使用文本文件和FileIO Xtra”，可以了解存储和读取文本的另一种方法。

22.6 与服务器和客户机通信

Director 7有一个新的Xtra，即Multiuser Xtra。这使得Shockwave影片、Director以及放映机能够与Director 7 Multiuser Server程序通信。这个Xtra是在Director 7中引入的。

我们可以使用这个Xtra直接与其他用户在因特网上通信，条件是他们也使用带有同一个Xtra的Shockwave影片或放映机。

要讲述Multiuser Xtra就需要整整一本书。有40多个命令、函数和属性都与之有关。此外，随着时间的推移，这个Xtra也很可能变得越来越丰富。参考在线的Director文件可以获得更多的信息和例子。在库面板中的少数几个行为可以帮助我们开始使用这个新的、令人振奋的功能。

22.7 Shockwave Lingo的故障排除

■ 当我们把一个Director影片放到服务器上时，一定要确保我们的FTP程序设置为二进制设置。使用ASCII码设置将会产生一个无效的文件。

■ 如果我们想使用JavaScript通信，一定要在观众可能使用的所有浏览器上对我们的影片进行测试。这是一项烦琐的工作，但是为了保证所有的浏览器对我们的命令能够进行正确处理，我们别无它法。

■ 当我们使用Preference文件时，应该总是使用“.txt”扩展名。如果使用其他扩展名，Shockwave会把这视作一个安全问题，并且会显示错误信息。

■ 当一部影片试图用getNetText从其他服务器获取文本时，将出现一个安全警告。为了避免这种警告，应当把文本和影片放在同一服务器上，并使用相对的路径名。

■ 在过去，如果gotoNetPage命令是由on exitFrame处理程序或被on exitFrame所调用的处理程序发出的，在Microsoft Internet Explorer中通过gotoNetPage使用目标时可能会无效。要进行测试，以确认你的目标能够在Internet Explorer中使用；如果不能，则用on mouseUp处理程序替代它。

■ 对于大多数创作者，使用Perl程序意味着或者控制我们自己的Web服务器，从而创建任

何类型的我们想要的Perl程序，或者与拥有我们的网站的网络服务提供商联系，以了解允许我们做些什么。

■ 如果我们从来都没有使用过Perl程序，使用它时会令我们有挫折感。我们可以通过在浏览器中输入CGI调用来对Perl程序进行测试。这样做是为了保证在使用 Shockwave影片调用它们之前，能够确认它们的功能正常。

22.8 你知道吗

■ 我们可以在Preference文件中放置多行文本，以存储我们想要的存储的足够信息。

■ 在getNetText里可以使用第二个参数，来指定一个不同的服务器字符组：“ JIS ”或“ EUC ”。这个参数缺省值是“ ASCII ”，并且“ AUTO ”这个设置试图自动确定服务器的字符组。

■ 在netDone命令返回TRUE之后，netMIME函数可以用来确定文件的MIME类型。

■ 在netDone命令返回TRUE以后，函数netLastModDate可以用来确定文件的服务器的时间标记。

■ 我们可以在HTTPS(安全服务器)上使用网络Lingo。这是Director 7的新功能。

■ 在 on EvalScript处理程序中可以用return将信息返回JavaScript。

■ 在使用gotoNetMovie命令时，全局变量存在于不同的影片之间。