

第37章 创建Java小程序

与Director 6.5一起发布的发行Director影片的一种新方法是 Save As Java功能。这个功能实际上是一个Xtra，它可以把Director影片、Lingo以及其他所有内容翻译成Java小程序。

这意味着我们可以创建Java小程序了。它实际上是由一系列小文件组成的计算机应用程序。这些小程序需要有Java引擎才能运行。Java小程序最常用于Web网页里。Netscape Navigator和Microsoft Internet Explorer都有能够运行Java小程序的Java引擎。

Java小程序的性能通常比和它等价的Shockwave差。它们通常需要更多的下载时间，而且输入后的Java小程序只支持Director的部分功能。实际上，与Shockwave相比，使用Java小程序的唯一优点是，用户不需安装任何额外的软件就可以在Netscape和Explorer上播放影片。

37.1 Save As Java基础信息

创建Save As Java功能是Macromedia的工程师们完成的一件了不起的事情。他们基本上用Java把Director重新写了一遍，然后又写出一个翻译器，把剪辑室、演员表和Lingo程序都翻译成Java小程序和数据文件。

很多创作者还没有意识到Save As Java Xtra的威力。最起码，当被问道有没有编写Java小程序的能力时，Director创作者或创作公司如今有机会说“有”。

创建Java小程序有两种方法。第一种是对已经制作好的影片进行修改，删除Save As Java Xtra所不支持的功能。

第二种方法是在开始创作影片时就想着Save As Java Xtra。在编程序时就要想到Save As Java Xtra的局限性。下面将详细讨论这两种方法。

37.1.1 把现有的影片翻译成Java小程序

在创作者们试着使用这种方法时，如果不能把旧的Director影片改为Java小程序，会感到非常失望。有很多Director的功能都不被Save As Java Xtra所支持。尽管有许多解决方法，但需要做很多工作才能把Director影片转换成Java小程序。

例如，我们需要把除位图、图形或域以外的所有演员都剔除。这意味着将要把文本演员转成位图，如果文本演员在影片播放时还将变化，则把它转成域。

我们也不能使用任何矢量图形演员或Flash演员了，因此这些演员都必须转成标准的位图。

然后，开始修改Lingo。许多命令都不能被翻译，不过大多数都可以有所变通。例如，子字符串系列命令都必须进行修改。不过，可以用chars和offset等函数写一些处理程序，来模拟line、word和char等函数。length和工具函数numToChar和charToNum等仍然有效。string()和value()函数也仍然有效。它们可以用来把数字转换成字符串或反向转换。

如果我们使用了Director的内置库行为，应当注意它们很可能不能被翻译。不过，我们可以使用另外一组专门为Java设计的库行为。应当只使用Java的这些行为。

如果有一些我们自己编写的行为，并想把它转换为Java，应当逐字逐句地检查，看是否

存在着与 Save As Java 功能不兼容的命令。测试自编行为和其他程序的方法是试着把它们转换成 Java 短程序，再查看在处理过程中会出现什么错误。

尽管把现有的影片翻译成 Java 是可能的，也终将成功，但并不推荐使用这种方法。另一种更好的方法是在开始创作影片之前就要考虑到 Java 的特性。

37.1.2 参考Java的特性制作影片

如果在从零开始创作影片时就记住 Save As Java Xtra 的限制，就可以制作出非常好的 Java 短程序。

即使没有文本或矢量图形等有趣的演员，Director 也是快速创作复杂程序的良好环境。当开始创作将来需要转换为 Java 短程序的影片时，应当忘掉那些不被 Java 短程序所支持的功能和 Lingo。我们将会发现仍旧有许多功能和选项可以使用。

此外，如果我们不打算自己编写 Lingo 处理程序，还可以使用 Java 的库行为以执行多种不同的动作。

当我们制作 Web 广告等简单的影片时，会发现在 Java 的限制下，工作仍然非常容易。但对于 Lingo 程序员来说，却要花一些时间来适应这种新的工作方式。他们必须忘掉一些 Lingo 技巧，只用 Java 所支持的 Lingo 来实现同样的目的。

由 Director 创建 Java 短程序的最好方法是在开始创作影片时就记住 Save As Java Xtra 的限制。

37.2 Save As Java Xtra的支持范围

在创作第一部将要制作成 Java 短程序的影片之前，应当先熟悉 Save As Java Xtra 支持 Director 的哪些功能，不支持哪些功能。而且，即使支持某些功能，在 Java 里的效果也不如在 Director 和 Shockwave 里好。此外，大多数不被支持的功能都可以用其他方法解决。

37.2.1 演员

这个 Xtra 共支持 6 种演员：位图、图形、域、声音和过渡，当然，还有剧本。其余的演员——数字视频文件、文本、链接的影片、影片片断、调色板、按钮、Shockwave 声音、字体演员和 Xtra 演员——则不被支持。

应当记住的第一件事是不能用变量来指代演员。例如，不能把 `member(1)` 赋予变量 `myMem`，然后用这样的表达方式：`myMem.width`，而应当直接引用演员：`member(1).width`。

1. 域

在创建 Java 短程序时，文本演员将被转换为位图。这意味着在影片播放的过程中不能再对它们进行编辑、修改等操作了。但如果把它们转换为域，就可以编辑了。

应当把所有的域都设为 Fixed 或 Scrolling 帧类型。Adjust to Size 帧类型通常不被支持。在 Java 短程序里，域的字号由其中的文字的多少来决定，当文字内容改变后，它不会自行调节。

此外，用来处理域的 Lingo 也是很有限的。应当一次替换域里的所有文字，而不能只替换局部。只要我们预先计划好，这种方法的效果也可以很好。

域可以使用的属性有：`text`、`font`、`fontSize`、`foreColor`、`backColor`、`fontStyle` 和 `border`。能够使用的式样只有粗体和斜体。可以使用任何帮助我们在域里定位字符的 Lingo，如

charPosToLoc。

2. 声音

AU格式是Java所支持的唯一格式。任何内部声音都将被转换为这种格式，任何外部声音也应当是这种格式。外部声音只有与演员链接才能被使用。在使用 sound play命令时，不能完全使用外部声音。

Java里的声音也不支持 Shockwave声音的流式传输、线索点、渐入 /渐出和音量控制。但 puppetSound命令仍然起作用。

3. 数字视频文件

简短的数字视频文件可以被翻译为连续的动画，但却没有一种好的方法能够把真正的视频文件放进Java小程序。此外，链接的影片片断也不被支持。

4. 调色板

不支持调色板，但支持 GIF图像。实际上，我们可以把 GIF用作外部文件。由于每一个 GIF都有它自己的调色板，因此使用 GIF是在Java小程序里使用 8-bit图像的好方法。所有内部图像都被转换为 JPEG，并且可以把它用作外部文件。

5. 链接的素材

如果使用链接的素材，应当记住几条规则。外部图像应当是 GIF或JPEG。如果有BMP或PCT等其他类型的图像，应当把它们转换成内部演员，而不是链接的图像。

所有链接的图像和声音都应当与影片在同一个文件夹里，或在影片的子文件夹里。可以使用Lingo里的filename属性切换演员所链接的图像。

6. 位图

在制作图像时，应当避免使用 1-bit的位深。最好使用 8-bit或32-bit的图像，这样 Save As Java Xtra能够以优化的方式把它们转换为 JPEG或GIF。

不能使用任何能够改动演员表的 Lingo，因此move、erase、duplicate和new（创建新演员）等Lingo都是无效的。此外，也无法获得演员的 picture、media、scriptText、scriptType或size属性。

在修改位图方面，有很多操作都不能做，但是可以读取和设置 regPoint属性。可以读取 rect属性，但却不能改变它。还不能使用 rotation或quad属性，不能在舞台上旋转或扭曲角色。

7. 图形

图形的表现与其在 Director和Shockwave里一样。如果用图案填充图形，只能使用前 15种图案。

至于图形的 Lingo，可以设置 shapeType、filled和lineSize等属性。

37.2.2 角色

在Save As Java Xtra所支持的6种演员里，只有三种可以被制作成角色：位图、域和图形。

1. 油墨

我们仅能使用有限的几种油墨类型，但是，这几种都是很常用的。表 37-1列出了每种角色各自可以使用的油墨。

也不可以使用角色的 trails和movable属性。Java行为库里有一个行为可以模拟可移动属性。

表37-1 Java短程序里的油墨使用情况

油 墨	位 图	域	图 形
Copy	可	可	可
Matte	可	与Background Transparent相同	可
Background Transparent	可	可	可
Blend	可	否	否

2. 角色Lingo

我们仍旧可以用Lingo改变位图或图形角色的位置、尺寸和形状。要做到这些，可以使用loc、locH、locV、rect、height和width等属性。可以改变域角色的位置，但不能改变它们的尺寸和形状。

用member属性可以改变角色所使用的演员。用ink属性可以切换油墨，用foreground和background可以切换颜色。还可以使用blend和visible属性。

要检查两个角色是否相交，可以同正常影片一样使用sprite...intersects句法。

同演员一样，也不能把角色存储在变量里并通过变量引用角色，而应当直接引用角色，如sprite(7).loc。

37.2.3 字体

关于可以使用哪些字体，Java是非常挑剔的。例如，域只能显示几种字体。表37-2列出了它们的Java名称及对应的Mac和Windows名称。

表37-2 Java短程序里可以使用的字体

Java字体名	相应的Windows字体	相应的Mac字体
Helvetica	Arial	Helvetica
TimesRoman	Times New Roman	Times
Courier	Courier-New	Courier
Dialog	MS Sans Serif	Chicago or Charcoal
DialogInput	MS Sans Serif	Geneva
ZapfDingbats	WingDings	ZapfDingbats
Default	Arial	Helvetica

要确定域将显示什么字体，只要查看与我们所使用的计算机类型相对应的那一栏（Mac或Windows）即可。然后，从另一栏就可以看出该字体在另一个平台上的对应字体。不要使用上表未列出的字体。

37.2.4 演员表库

如果我们在影片里建立多个演员表库，虽然Save As Java Xtra能够识别它们，但使用多个演员表库的优点却不存在了。实际上，Java认为所有演员都处于同一个大型演员表内。

为了创作的方便，我们可以使用多个演员表。建议每个演员的名称都是唯一的，而且要用演员的名称引用它们，而不要使用演员表库引用它们。

37.2.5 网络Lingo

由于Java是因特网的编程语言，因此很多网络Lingo命令都能被翻译成Java命令。

getNetText和gotoNetPage命令与它们在Shockwave里的作用是一样的，但netTextResult实际上返回的是最近10次getNetText调用的组合结果。因此最好只对一个文件使用getNetText和netTextResult，而不要同时调用多个文件。

此外，getLatestNetID、netAbort、netDone、netError、netMime和netStatus等都起作用，以支持getNetText。

但gotoNetMovie不起作用。Java小程序只能播放被转换成Java小程序的那一个影片，不能跳到另一个影片。我们可以用gotoNetPage跳到含有另一个小程序的Web网页。

不能使用downloadNetThing、streamStatus和tellStreamStatus、getPref和setPref。此外，getNetText和其他网络命令只能读取处于同一个服务器里的文件。因此我们不能把某个链接的位图的filename设置为不在本服务器的某个文件，这是由Java内部的安全限制所决定的。

与浏览器联络似乎应当是有效的，但还是有一定的限制。externalEvent命令仅能在Netscape上用于向JavaScript发送信息，而不能用于Internet Explorer。不能使用EvalScript与JavaScript联络。

37.2.6 其他Lingo

有些Lingo可以帮助我们判断演员是否已准备好显示。frameReady和mediaReady与它们在Director里的功能一样正常。preloadMember也允许我们把演员调入内存，但条件是演员已从因特网调入了用户的计算机。

不能使用引用其他影片的Lingo，如go to movie或play movie。此外，根本不能使用play。但是，Java行为库里的某些行为的功能与play和play done一样。

不能使用pause、continue或delay等命令。但有经验的Lingo程序员也不使用这几个命令。库行为能提供足够的浏览功能。

支持节奏的变化。这种支持不仅表现在剪辑室里，而且还表现在可以使用Lingo命令puppetTempo。

尽管支持行为，但不支持行为间的联络。因此不存在sendSprite、sendAllSprites、call、callAncestor、ancestor或scriptInstanceList等命令。

仅能部分支持键盘Lingo。on keyUp和on keyDown事件处理程序的功能照常。但是，shiftDown、optionDown、controlDown或commandDown等属性如果不是在那个键被按下时被调用的处理程序里，就不起作用。不能使用keyPressed属性。

关于键盘操作，应当记住浏览器将首先拦截所有键盘命令。这意味着小程序先要被用户点击才能接受键盘命令，而且，浏览器所使用的键盘快捷键命令永远都不能到达小程序。因此，Command+D或Ctrl+D只用于在浏览器里添加书签，而永远不能到达小程序。

此外，鼠标的位置只能在处理程序之间被更新，因此在repeat循环里它不能被更新。

还有，只支持-1、0、1、2和4光标。有关timeout的Lingo都不起作用。on idle处理程序和鼠标右键处理程序都不起作用。

37.2.7 过渡

并不是所有过渡都被Save As Java Xtra所支持。Lingo命令puppetTransition无效，只有剪辑室里的过渡才有效。以下列出支持的过渡：

Center out (horizontal、vertical和square)

Checkerboard

Dissolve (bits、bits fast、boxy rectangles、boxy squares、patterns、pixels和pixels fast)

Edges in (horizontal、vertical和square)

Random (column和row)

Venetian blinds

Vertical blinds

Wipe (right、left、down和up)

37.3 走进Java

Save As Java Xtra实际上是把Director影片转换成一系列的文件。这些文件里有几个.class文件。它们是纯粹的Java。其中有些文件名是Sprite.class和Member.class，因此我们可以猜出其中的内容。

此外，还有一个.djr文件。其中是影片里的素材的另一个版本。该文件的尺寸与原始影片的尺寸相似。

还有一个.class文件，它的名称与影片的名称相同，其中含有被翻译的Lingo代码。

Lingo的翻译几乎是逐行进行的。如果我们使用Source Java选项，而不是Compiled Java选项进行输出，再查看生成的源代码，就可以发现这一点。

创作者应当意识到，Java和Lingo处理变量的方式是很不同的。在Lingo里，我们可以让某个变量有时存储数字，有时存储字符串；但在Java里，某个变量只能存储某种类型的数字，如整数。

为了补偿这个差异，工程师们已经创建了LVal变量类型。这种Java变量类型可以存储一种结构，其中可以包含任何类型的值，如整数和字符串，甚至是列表。

LVal结构的弊端在于它很慢。毕竟它要考虑多种不同类型的值，并时刻准备着处理任何类型的值。我们可以创建只使用整数的变量，这样可以提高速度。

把一个变量做成真正的整数变量的方法是用`x=0`这样一些简单的命令“声明”该变量。然后，在整个处理程序或影片内，只把它设置为整数。

这件事做起来并不像说的那样简单。如果我们写下这样的代码`x=myFuntion()`，Save As Java Xtra肯定会认为该函数可以返回一个非整数值，于是它会把`x`定为LVal类变量。但是，如果我们用`x=integer(myFuntion())`，Java就知道`x`将接受整数值。

使用这一技巧的唯一好处是能提高一些速度。但速度在Java短程序里是最重要的，因为Java并不是速度很快的环境。如果想制作arcade风格的游戏，就应当极力地进行优化。

37.4 使用Save As Java Xtra

尽管Save As Java功能是一个Xtra，而且出现在Xtras文件夹里，但并不是在Xtra菜单里调用它，而是选择File | Save As Java来调用它。

首先看到的是Save As Java对话框。图37-1就是这个对话框，其中只有几个选项。

我们可以选择输出Java源代码还是输出编译后的Java短程序。如果你会用Java编程序，或是出于好奇，就可以输出源代码，以便看到Lingo被翻译后的结果。

但是，管理角色等执行 Director 的非凡功能的Java代码已看不到了，因为它们已经被编译了，创作者不可以使用。

如果选择 Minimize Player Size，Director 的Java播放器里的任何没有被影片所使用的部分都将自动被删除。选择这个选项后，Save As Java功能将需要更长时间，但最终生成的短程序的文件尺寸却小得多。当生成最终的短程序时，应当选择这个选项。

Embed Linked Media选项将把外部演员放在.djr文件里。

在生成短程序之前，应当按 Options按钮。这个按钮将调出如图 37-2所示的对话框。

第一个选项是JPEG质量。由于大多数位图演员都将被转换为 JPEG格式，在此应当设置为最高质量，位图才能保持原来的外观。不过，可以试着使用较低的设置，看看它对文件的尺寸有何影响。

Playback选项有Loop和Pause At End，它们很容易理解。

下一个是No Multi-Line Text Fields选项。

如果选择该项，所有域都将只包含一行。选择该选项可以提高速度。

其余大部分选项的目的也都是为了提高速度，条件是我们愿意牺牲一些功能。有些选项我们几乎不用，如 No List-to-String Conversion、No Arithmetic On Lists和No Colorized Bitmaps等。如果没有使用相关的功能，就把它们关闭。

Lists Contain Only Integers选项可以提高列表的处理速度。不过，我们只能使用含有整数的线性列表。如果在编程时可以避开这个问题，也可以关闭该选项。

完成对这些选项的选择后，点击 OK，回到主对话框。先用 Check按钮检查演员表、剪辑室和Lingo，查看那些不被支持的项目。我们将得到一个清单，其中列着不被支持的项目（如果有的话）。我们需要返回头进行一些修改，在继续操作之前找到一些解决方法。

如果Check没有检查出任何问题，点击 Save，制作Java短程序。在它被编译之后，点击 Run，可以在浏览器上测试它。

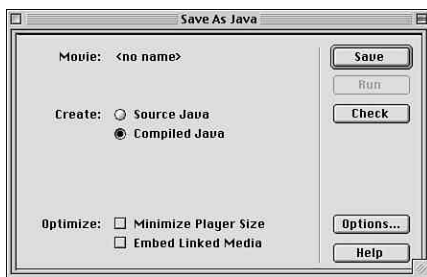


图37-1 Save As Java对话框里仅有几个选项，但大部分选项都隐藏在Options按钮里

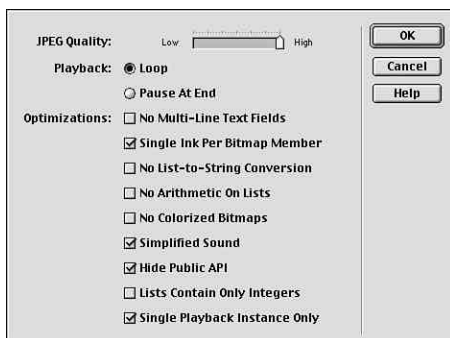


图37-2 Save As Java的选项对话框，在这里可以决定短程序所支持的项目

37.5 生成Java的技巧

除了使用常规方法创建 Director影片外，在制作Java短程序前，还应当了解一些非常规的技巧。首先，应当熟悉特殊的Java库行为。

较熟练的用户可能需要了解如何制作可以同时被用作 Shockwave短程序和Java短程序的影片，然后可以嵌入Save As Java Xtra能够识别的Lingo代码段。

此外，Java编程员需要知道如何在Lingo处理程序里嵌入真正的Java代码。

37.5.1 Java行为

尽管有经验的 Lingo 程序员不喜欢使用 Director 内置的现成的行为，但我们还是应当仔细地研究一下面向 Java 的库行为。

这些库行为使我们能用简单的方法在影片里设置一些复杂的功能，却又不必担心 Java 是否支持它们。在编写这些行为时，程序员们已经考虑到了 Java 的各种限制，因此我们只需把它们拖到角色和帧上即可。

在行为监察窗里，可以查看每个行为的描述。我们也可以把行为拖到演员表里，然后在 Script Editor 里打开它们，以查看 Macromedia 在编写这些程序时是如何避开那些限制的。

除了行为外，库里还有两个影片剧本。第一个是一个处理程序，它把字符串转换为列表。这是很有用的，因为只使用 value 不能像它在 Director 里正常发挥功能时那样地把字符串转换成列表。

还有两个处理程序允许我们在 Java 短程序里模拟 play 和 play done 命令。

37.5.2 对Java隐藏Lingo

在我们制作 Java 短程序时，可能也需要把它用作 Shockwave 影片。毕竟 Shockwave 的速度更快而且更可靠。

要创建 Shockwave 影片，只要把影片存储为 Shockwave 文件即可。但是，偶尔还需要在 Shockwave 影片里添加额外的功能。这些功能可能是 Java 所不支持的功能，也可能是采用了实现同一种功能的比 Java 方法速度更快的方法。

因此，有一种方法可以对 Save As Java Xtra 隐藏 Lingo。可以使用简单的命令行 “--Lingo begin” 表示这部分的开始，并用 “--Lingo end” 结束它。请看下面的例子：

```
list = [1,3,2,5,12,6,3,7,3,7,3]
--Lingo begin
sort(list)
--Lingo end
```

在这个例子里，sort 命令对于 Java Xtra 隐藏起来。如果不隐藏，Save As Java 进程将被中止，并报出一个错误信息，因为它不支持 sort 命令。在这个例子里，对于 Shockwave 短程序，列表被分类，作为对用户的“奖励”，但对于 Java 短程序，列表却不被分类。

37.5.3 嵌入Java代码

也可以在影片里嵌入真正的 Java 代码。该代码被 Director 忽略，因为它以注释行出现。但是，它却可以被 Save As Java Xtra 识别和使用。其源代码出现在 Java 源程序里的“注释”的位置。

当然，只有专业人员才善于使用 Lingo/Java 嵌入。这些代码应当出现在 “--Java begin” 和 “--Java end” 命令行之间，而且每行都应当用双连字符开头。在 Macromedia 的网站查找 technote(s)，可以看到一些实例。

37.6 制作Java短程序的收尾工作

同 Shockwave 影片一样，直到把 Java 短程序放到 Web 网站上，才算完成产品的制作。这需要几个步骤，包括捆绑短程序、确认它与所有可能的素材都兼容以及短程序的故障排除。

37.6.1 捆绑短程序

在完成输出后，得到的是许多小的 .class 文件和 .djr 文件。还有一个 HTML 样例文件。应当把所有这些文件都放到服务器上，但这并不是推荐使用的发行产品的方法。

取而代之的是应当把这些文件捆绑为一种压缩格式。我们需要使用一个用于老版本浏览器的未压缩的 ZIP 文件、一个用于 Netscape 4.x 的 JAR 文件和一个用于 Internet Explorer 的 CAB 文件。这三个文件都是把 Java 短程序放在一个单独的文件里。

尽管生成这三个文件十分费时间，不过 Macromedia 已经使这个过程变得简单了。只要使用 Director 7 所附带的 AfterShock 程序就可以了。除了完成 Director Shockwave 影片的许多收尾工作外，它还负责创建这三个文件。它甚至还生成一些现成的 HTML 代码可供我们直接使用，或拷贝、粘贴到我们的 HTML 文件框架里去。

关于 AfterShock 的更多信息，请参见它的说明书，或试着使用它。这个软件非常具有新意。

37.6.2 兼容问题

Save As Java Xtra 输出纯正的 Java。它应当能运行在任何支持 Java 1.02 的环境里，这意味着我们的 Director 内容甚至可以运行在基于 UNIX 的计算机或 Java 计算机上。

对于浏览器来说，由 Director 生成的 Java 短程序可以运行在 Netscape Navigator 和 Microsoft Internet Explorer 3.0 及以上版本上。它也可以运行在这些浏览器的 Mac 和 Windows 版本上。

不过，和 Shockwave 一样，Java 短程序也要经过尽可能多的测试。如果我们想让短程序运行在非标准的浏览器上时尤其如此。

37.6.3 问题及更多信息

在此郑重建议你仔细阅读 Macromedia 关于 Save As Java Xtra 的说明书。你不但能从其中了解到更多信息，而且还将读到关于升级的内容。

由于这个功能是用 Xtra 的方式添加的，Macromedia 有可能在 Director 的另一个重要版本出台前继续改进它。时常访问 Macromedia 网站，看看这个 Xtra 有没有被更新。

用 Director 创建 Java 短程序是一件微妙又棘手的事情。你应当预料到它会比创建 Shockwave 所花费的时间要长。

37.7 Java短程序的故障排除

Director 7 所附带的 Java 行为中有个别的行为有程序错误。当你对它们进行拖动操作时，可能会收到错误信息，因为它们没有使用关于 member 的恰当句法。如果你了解 Lingo，可以很容易地改正这些错误。也可以访问 Macromedia 网站，以获取更新后的版本。

即使 Save As Java Xtra 没有检测出错误，但 Java 编译器却可能检测出错误。在这种情况下，你需要花一些时间查找是哪些 Lingo 程序有问题。

在需要软件 100% 地正确的场合，我建议你不要使用 Java。Java 仍是一种不很成熟的语言，而且有很多版本，使得我们难以生成 100% 正确的短程序。短的动画和演示比 Lingo 密集型的影片的正确率更高些。

如果 Java 短程序没有被放到服务器上，并在那里进行过测试，而只是在你的本地硬盘上做过测试，Internet Explorer 的某些版本可能就不能正常地运行它。Shockwave 影片就不是这样，而是可以用各种浏览器在本地进行测试。