



Applet

进入词条

全站搜索

帮助

声明：百科词条人人可编辑，词条创建和修改均免费，绝不存在官方及代理商付费代编，请勿上当受骗。详情>>

首页

分类

特色百科

用户

权威合作

手机百科

个人中心

Applet 编辑

收藏 770 13

Applet是采用Java编程语言编写的小应用程序，该程序可以包含在HTML（标准通用标记语言的一个应用）页中，与在页中包含图像的方式大致相同。^[1]含有Applet的网页的HTML文件代码中部带有<applet>和</applet>这样一对标记，当支持Java的网络浏览器遇到这对标记时，就将下载相应的小应用程序代码并在本地计算机上执行该Applet。

中文名	小应用程序	编写语言	Java
外文名	Applet	适用范围	通常用于网页程序

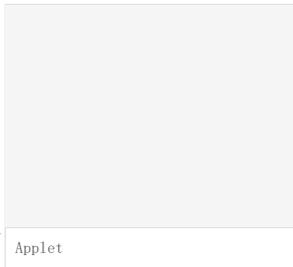
目录	1 工作原理	· 有关参数说明	4 实现方法
	· 安全性限制	3 调用Applet	· 服务器端
	· 开发步骤	· 重要	· 客户端
	2 常用方法	· 缺陷	· 监控平台
	· 运行状态	· 优化	

工作原理

Java Applet 是用Java 语言编写的一些小应用程序，这些程序是直接嵌入到页面中，由支持Java的浏览器(IE 或 Netscape)解释执行能够产生特殊效果的程序。它可以大大提高Web页面的交互能力和动态执行能力。包含Applet的网页被称为Java-powered页，可以称其为Java支持的网页。

当用户访问这样的网页时，Applet被下载到用户的计算机上执行，但前提是用户使用的是支持Java的网络浏览器。由于Applet是在用户的计算机上执行的，所以它的执行速度不受网络带宽或者Modem存取速度的限制，用户可以更好地欣赏网页上Applet产生的多媒体效果。

在Java Applet中，可以实现图形绘制，字体和颜色控制，动画和声音的插入，人机交互及网络交流等功能。Applet还提供了名为抽象窗口工具箱（Abstract Window Toolkit，AWT）的窗口环境开发工具。AWT利用用户计算机的GUI元素，可以建立标准的图形用户界面，如窗口、按钮、滚动条等等。目前，在网络上有非常多的Applet范例来生动地展现这些功能，读者可以去调阅相应的网页以观看它们的效果。



Applet 小应用程序的实现主要依靠java.applet包中的Applet类。与一般的应用程序不同，Applet应用程序必须嵌入在HTML页面中，才能得到解释执行；同时Applet可以从Web页面中获得参数，并和Web页面进行交互。

Applet是一种Java的小程序，它通过使用该Applet的HTML文件，由支持Java的网页浏览器下载运行。也可以通过java开发工具的appletviewer来运行。Applet 程序离不开使用它的HTML文件。这个HTML文件中关于Applet的信息至少应包含以下三点：

- 1)字节码文件名(编译后的Java文件，以.class为后缀)
- 2)字节码文件的地址
- 3)在网页上显示Applet的方式。

一个HTML文件增加Applet有关的内容只是使网页更加富有生气，如添加声音、动画等这些吸引人的特征，它并不会改变HTML文件中与Applet无关的元素。

安全性限制

为了防止这样的问题出现，在缺省的情况下，大多数浏览器在Applet安全方面受到诸多的限制，几乎不能对系统进行任何“读”或“写”的操作：

1. 不允许Applet访问本地文件系统
2. Applet不能执行任何本地计算机上的程序



Applet图册

V百科

往期



相关人物



程序设计类书籍



词条统计

浏览次数：364663次
编辑次数：50次历史版本
最近更新：2017-06-23
创建者：lym13

3. 不允许Applet尝试打开除提供Applet的服务器之外的任何系统的Socket
4. 不允许Applet访问用户名、电子邮件地址等与本地系统有关的信息
5. 对于Applet而言，Java环境中的某些标准系统属性是只读的
6. Applet不能创建或装载Applet的机器的网络连接
7. Applet不能充当网络服务器，监听或接收来自远程系统的连接请求
8. 不允许Applet装载动态库或定义本地方法调用
9. Applet不能操纵不在自己线程组中的任何线程
10. Applet不能关闭JVM。

开发步骤

Applet程序开发主要步骤如下：

- 1)选用EDIT或Windows Notepad等工具作为编辑器建立Java Applet源程序。
- 2)把Applet的源程序转换为字节码文件。
- 3)编制使用class 的HTML文件。在HTML文件内放入必要的<OBJECT>语句。

下面举一个最简单的HelloWorld 例子来说明Applet程序的开发过程：

(1) 编辑Applet 的java源文件

创建文件夹C:\ghq，在该文件夹下建立 HelloWorld.java

文件的源代码如下：

```
import java.awt.*;

import java.applet.*;

public class HelloWorld extends Applet //继承Applet类，这是Applet Java程序的特点
{

public void paint(Graphics g )

{

g.drawString("Hello World!",5,35);

}

}
```

保存上述程序在C:\ghq\HelloWorld.java文件里。

(2)编译Applet

编译HelloWorld.java源文件可使用如下JDK命令：

```
C:\ghq>javac HelloWorld.java<Enter>
```

注意：如果编写的源程序违反了Java编程语言的语法规则，Java编译器将在屏幕上显示语法错误提示信息。源文件中必须不含任何语法错误，Java编译器才能成功地把源程序转换为appletviewer和浏览器能够执行的字节码程序。

成功地编译Java applet之后生成响应的字节码文件HelloWorld.class的文件。用资源管理器或DIR命令列出目录列表，将会发现目录C:\ghq中多了一个名为HelloWorld.class的文件。

(3)创建HTML文件

在运行创建的HelloWorld.class 之前，还需创建一个HTML文件，appletviewer或浏览器将通过该文件访问创建的Applet。为运行HelloWorld.class，需要创建包含如下HTML语句的名为HelloWorld.html的文件。

```
<HTML>

<TITLE>HelloWorld! Applet</TITLE>

<APPLET

CODE="HelloWorld.class" //这里应该是HelloWorld.class
```

分享



```
WIDTH=200  
HEIGHT=100>  
</APPLET>  
</HTML>
```

本例中，<APPLET>语句指明该Applet字节码类文件名和以像素为单位的窗口的尺寸。虽然这里HTML文件使用的文件名为HelloWorld.HTML，它对应于HelloWorld.java的名字，但这种对应关系不是必须的，可以用其他的任何名字(比如说Ghq.HTML)命名该HTML文件。但是使文件名保持一种对应关系可给文件的管理带来方便。

(4)执行 HelloWorld.html

如果用appletviewer运行HelloWorld.html,需输入如下的命令行:

```
C:\ghq>appletviewer HelloWorld.html<ENTER>
```

可以看出，该命令启动了appletviewer并指明了HTML文件，该HTML文件中包含对应于HelloWorld的<APPLET>语句。

如果用浏览器运行HelloWorld Applet，需在浏览器的地址栏中输入HTML文件URL地址。

至此，一个Applet程序的开发运行整个过程结束了(包括java源文件、编译的class文件、html文件以及用appletviewer或用浏览器运行)。

//还可应用程序:

```
import java.awt.*;  
import java.applet.*;  
public class Test extends Applet //继承Applet类，这是Applet Java程序的特点  
{  
    public void paint(Graphics g)  
    {  
        people zhang;  
        zhang=new people();  
        zhang.setname("Robot");  
        zhang.setage(12);  
        g.drawString(zhang.talk(),5,35);  
    }  
}  
class people  
{  
    private String name ;  
    private int age ;  
    String talk()  
    {  
        return "我是"+name+";今年"+age+"岁";  
    }  
    public void setname(String str)  
    {  
        name = str ;  
    }  
    public void setage(int a)
```

分享



```

{
if(a>0)

age = a ;

else age=-a;

}

public String getname()

{

return name ;

}

public int getage()

{

return age ;

}

}

```

常用方法

[编辑](#)

applet的超类是panel.

```
public final void setStub(AppletStub stub)
```

//设置Applet的stub.stub是Java和C之间转换参数并返回值的代码位，它是由系统自动设定的。

```
public boolean isActive();//判断一个Applet是否处于活动状态。
```

```
public URL getDocumentBase();//检索表示该Applet运行的文件目录的对象。
```

```
public URL getCodeBase();//获取该Applet 代码的URL地址。
```

```
public String getParameter(String name); //获取该Applet 由name指定参数的值。
```

```
public AppletContext getAppletContext(); //返回浏览器或小应用程序观察器。
```

```
public void resize(int width,int height); //调整Applet运行的窗口尺寸。
```

```
public void resize(Dimension d); //调整Applet运行的窗口尺寸。
```

```
public void showStatus(String msg); //在浏览器的状态条中显示指定的信息。
```

```
public Image getImage(URL url); //按url指定的地址装入图象。
```

```
public Image getImage(URL url,String name); //按url指定的地址和文件名加载图像。
```

```
public AudioClip getAudioClip(URL url); //按url指定的地址获取声音文件。
```

```
public AudioClip getAudioClip(URL url, String name); //按url指定的地址和文件名获取声音。
```

```
public String getAppletInfo(); //返回Applet应用有关的作者、版本和版权方面的信息；
```

```
public String[][] getParameterInfo();
```

//返回描述Applet参数的字符串数组，该数组通常包含三个字符串：参数名、该参数所需值的类型和该参数的说明。

```
public void play(URL url); //加载并播放一个url指定的音频剪辑。
```

```
public void destroy(); //撤消Applet及其所占用的资源。若该Applet是活动的，则先终止该Applet的运行。
```

运行状态

Applet类中的四种基本方法用来控制其运行状态：init()、start()、stop()、destroy()

init()方法

这个方法主要是为Applet的正常运行做一些初始化工作。当一个Applet被系统调用时，系统首先调用的就是该方法。通常可以在该方法中完成从网页向Applet传递参数，添加用户界面的基本组件等操作。

[分享](#)


start()方法

系统在调用完init()方法之后，将自动调用start()方法。而且，每当用户离开包含该Applet的主页后又再返回时，系统又会再执行一遍start()方法。这就意味着start()方法可以被多次执行，而不像init()方法。因此，可把只希望执行一遍的代码放在init()方法中。可以在start()方法中开始一个线程，如继续一个动画、声音等。

stop()方法

这个方法在用户离开Applet所在页面时执行，因此，它也是可以多次执行的。它使你可以在用户并不注意Applet的时候，停止一些耗用系统资源的工作以免影响系统的运行速度，且并不需要人为地去调用该方法。如果Applet中不包含动画、声音等程序，通常也不必实现该方法。

destroy()方法

与对象的finalize()方法不同，Java在浏览器关闭的时候才调用该方法。Applet是嵌在HTML文件中的，所以destroy()方法不关心何时Applet被关闭，它在浏览器关闭的时候自动执行。在destroy()方法中一般可以要求收回占用的非内存独立资源。(如果在Applet仍在运行时浏览器被关闭，系统将先执行stop()方法，再执行destroy()方法。)

有关参数说明

利用Applet来接收从HTML中传递过来的参数,下面对这些参数作一简单说明:

*** CODE标志**

CODE标志指定Applet的类名; WIDTH和HEIGHT标志指定Applet窗口的像素尺寸。在APPLET语句里还可使用其他一些标志。

*** CODEBASE 标志**

CODEBASE标志指定Applet的URL地址。Applet的通用资源定位地址URL，它可以是绝对地址。也可以是相对于当前HTML所在目录的相对地址，如/AppletPath/Name。如果HTML文件不指定CODEBASE标志，浏览器将使用和HTML文件相同的URL。

*** ALT 标志**

虽然Java在WWW上很受欢迎，但并非所有浏览器都对其提供支持。如果某浏览器无法运行Java Applet，那么它在遇到APPLET语句时将显示ALT标志指定的文本信息。

*** ALIGN 标志**

ALIGN标志可用来控制把Applet窗口显示在HTML文档窗口的什么位置。与HTML语句一样，ALIGN标志指定的值可以是TOP、MIDDLE或BOTTOM。

*** VSPACE与HSPACE 标志**

VSPACE和HSPACE标志指定浏览器显示在Applet窗口周围的水平和垂直空白条的尺寸，单位为像素。

*** NAME 标志**

NAME标志把指定的名字赋予Applet的当前实例。当浏览器同时运行两个或多个Applet时，各Applet可通过名字相互引用或交换信息。如果忽略NAME标志，Applet的名字将对应于其类名。

*** PARAM 标志**

通用性是程序设计所追求的目标之一。使用户或者程序员能很方便地使用同一个Applet完成不同的任务是通用性的具体表现。从HTML文件获取信息是提高Applet通用性的一条有效途径。

假设编制了一个把某公司的名字在屏幕上卷动的Applet。为了使该Applet更加通用，则可以使该Applet从HTML文件获取需要卷动的文本信息。这样，若想显示另一个公司的名字，用不着修改Java Applet本身，只需修改HTML文件即可。

PARAM 标志可用来在HTML文件里指定参数，格式如下所示:

```
PARAM Name="name" Value="Liter"
```

Java Applet可调用getParameter方法获取HTML文件里设置的参数值。

上面这个例子就是一个简单主页的HTML文件代码。代码第五行中的<P>，是为了确保Applet出现在新的一行，若没有它，Applet将会紧接着上一行的最后一个单词出现。代码第六、七两行是关于Applet的一些参数。其中第六行是必需的Applet参数，定义了编译后的包含Applet字节码的文件名，后缀通常为“.class”；和以像素为单位的Applet的初始宽度与高度。第七行则是附加的Applet参数，它由一个分离的标记来指定其后的名称和值，在这里是img的值为“example.gif”，它代表了一个图形文件名。

Applet的下载与图形文件一样需要一定的时间，若干秒后它才能在屏幕上显示出来。等待的时间则取决于Applet的大小和用户的网络连接的速度。一旦下载以后，它便和本地计算机上的程序以相同的速度运行了。

分享



Applet在用户的计算机上执行时，还可以下载其它的资源，如声音文件、图像文件或更多的Java代码，有些Applet还允许用户进行交互式操作。但这需要重复的链接与下载，因此速度很慢，这是一个亟待解决的问题，可以想到的一个好办法是采用类似高速缓存的技术，将每次下载的文件都临时保存在用户的硬盘上，虽然第一次使用时花的时间比较多，但当再次使用时，只需直接从硬盘上读取文件而无需再与Internet连接，便可以大大提高性能了。

调用applet

[编辑](#)

大家都知道由于java2的安全性，对applet的正常调用的html文件已经不能再使用了，而改为ActiveX类型的调用。具体的又分ie和nescape的不同写法，这一些在sun网上都能找到现成的教程。

重要

其中我要强调一点，因为applet每一次的改动都需要重新打包签名，手续非常繁琐，所以在具体的实现中要将一些会变化参数放到html文件中来，传到applet中去，这一点网上文章好多，自己去看看吧。

缺陷

另外一个就是有朋友问我，那这样不是太麻烦了，每一个客户端都要进行复杂的dos命令操作，我只能说我的水平只能将一个已经做好的客户端文件cer文件和java.policy以及cacerts文件直接拷贝到客户端，当然这也有缺陷，如果别人的计算机已经有了认证，就会丢失。就这些问题我们可以一起探讨。

优化

另外还有一点优化，就是在打包的时候，我这儿只讲了把所有要用的涉及到安全性的包和源程序都要打到一个包中。这样如果包非常大的话，会非常影响下载的速度，如果可以使用本地计算机的包就好了，这一点jre也做到了，具体的要到控制面板的jre控制台上设置。这个就留着读者自己去摸索吧。

实现方法

[编辑](#)

服务器端

1. 将程序需要用到的各种包文件全部解压(要用到WEBLOGIC的JMS包使用命令jar xf weblogic.jar)，然后使用JDK的打包命令将编译好的监控程序.class和刚才解压的包一起打包到一个包中。(前提我已经将监控程序和解开的包都放在同一个目录下)，都是dos状态下的命令，具体命令见jdk1.3(1.2)的bin目录下，

命令如下：

```
jar cvf monitor.jar *.class
```

此命令生成一个名为monitor.jar的包

2. 为刚才创建的包文件（monitor.jar）创建keystore和keys。其中，keystore将用来存放密匙(private keys)和公共钥匙的认证，alias别名这儿取为monitor。

命令如下：

```
keytool -genkey -keystore monitor.keystore -alias monitor
```

此命令生成了一个名为monitor.keystore的keystore文件，接着这条命令，系统会问你好多问题，比如公司名称，地址，要设置的密码等等，都可以随便写。

3. 使用刚才生成的钥匙来对jar文件进行签名

命令如下：

```
jarsigner -keystore monitor.keystore monitor.jar monitor
```

这个命令将对monitor.jar文件进行签名，不会生成新文件。

4. 将公共钥匙导入到一个cer文件中，这个cer文件就是要拷贝到客户端的唯一文件。

命令如下：

```
keytool -export -keystore monitor.keystore -alias monitor -file monitor.cer
```

此条命令将生成monitor.cer认证文件，当然这几步都有可能问你刚才设置的密码。这样就完成了服务器端的设置。这时就可以将jar文件和keystore文件以及cer文件(我这儿是monitor.jar,monitor.keystore,monitor.cer)拷贝到服务器的目录下了。

客户端

1. 首先应该安装jre1.3.0_01，然后将服务器端生成的monitor.cer文件拷贝到jre的特定目录下，我这儿是：

c:\program files\jvasoft\jre\1.3.0_01\lib\security目录下。

2. 将公共钥匙倒入到jre的cacerts（这是jre的默认keystore）

[分享](#)

命令如下：

```
keytool -import -alias monitor -file monitor.cer
```

```
-keystore cacerts
```

注意这儿要你输入的是cacerts的密码，应该是changeit，而不是你自己设定的keystore的密码。

3. 修改policy策略文件，在dos状态下使用命令 policytool

系统会自动弹出一个policytool的对话框，如图4所示，在这里面首先选择file菜单的open项，打开c:\program files\javasoft\jre\1.3.0_01\lib\security目录下的java.policy文件，然后在edit菜单中选择Change keystore，在对话框中new keystore url:中输入

file:/c:/program files /javasoft/jre/1.3.0_01/lib/security/cacerts, 这儿要注意反斜杠，在new keystore type 中输入JKS，这是cacerts的固定格式，然后单击Add Policy Entry，在出现的对话框中CodeBase中输入：

```
http://URL:7001/*
```

其中的URL是服务器的IP地址，7001是weblogic的端口，如果是在别的应用服务器上比如说是apache，那端口号就可以省略掉。

在SignedBy中输入(别名alias):这儿是Monitor然后单击add peimission按钮，在出现的对话框中permission中选择你想给这个applet的权限，这儿具体有许多权限，读者可以自己找资料看看。这儿就选用allpeimission，右边的signedBy中输入别名：monitor

最后保存，在file菜单的save项。

这样客户端的设置就完成了。在客户端用ie运行该applet程序时，会询问你是不是对该签名授权，选择授权后，包会自动从服务器上下载到本地计算机，而且ie会自动启动jre，在右下栏中可以看见，相当于ie的java控制台。

4. 调用applet的html文件

applet的正常调用的html文件已经不能再用了。

```
<html>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=gb2312">
<center>
```

监控平台

```
<hr>
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
width="900" height="520" align="baseline" codebase="http://192.168.2.217:7001/j2re-1_3_0_01-win-
i.exe#Version=1,3,0,0">
</OBJECT>
</center>
</html>
```

参考资料

1. 在 Web 应用程序中集成小应用程序 . netbeans[引用日期2013-12-24]

词条标签： 软件

分享



猜你喜欢

快讯通

墙体彩绘机

饺子皮机

天津港进口车报价

原油交易所

北京新房房价

毕业论文代写

天平

垃圾桶

创业小项目

🔍 新手上路

成长任务

编辑入门

📖 我有疑问

我要质疑

我要提问

🗨️ 投诉建议

举报不良信息

未通过词条申诉

分享

