第10章 部署应用程序和applet

## 10.3 Applet

10.3.6 访问图像和音频文件

在编写这本书时， 图像必须是GIF、PNG或JPEG格式。音频文件必须是AU、AIFF、WAV或MIDI格式。动画支持GIF，并且能显示动画效果。

需要利用需要利用相对的URL指定图像和音频文件的位置。通常，由调用getDocumentBase或getCodeBase方法获得基URL。前者可以获得包含applet的HTML网页的URL；后者可以获得applet的codebase的URL。

可以通过getImage和getAudioClip方法获得基URL和文件的存储位置。例如：

Image cat = getImage(getCodeBase(), “images/cat.gif”);

AudioClip meow = getAudioClip(getCodeBase(), “audio/meow.au”);

要想播放音频剪辑，只需要调用play方法即可。还可以调用Applet类中的play方法而不用加载音频剪辑。

play(getCodeBase(), “audio/meow.au”);

**java.applet.Applet 1.0**

1. URL getDocumentBase()

获得包含applet的网页的URL

1. URL getCodeBase()

获得codebase目录的URL，applet是由这个目录加载的。返回的URL既可以是由codebase属性指定的引用目录的绝对URL，在没有指定codebase的情况下，也可以是HTML文件的目录。

1. void play(URL url)
2. void play(URL url, String name)

前一个方法播放由URL指定的音频文件。第二个方法利用字符串提供相对于第一个参数URL的路径。第二个方法利用字符串提供相对于第一个参数URL的路径。如果没有找到音频剪辑，则不做任何操作。

1. AudioClip getAudioClip(URL url)
2. AudioClip getAudioClip(URL url, String name)

前一个方法获得URL指定的音频剪辑。第二个方法使用字符串来提供相对于第一个参数URL的路径。如果没有找到音频剪辑，则返回null.

1. Image getImage(URL url)
2. Image getImage(URL uril, String name)

返回一个由URL指定的图像对象，这个对象封装一个图像。如果图像不存在，则立即返回null。否则，将启动一个独立的纯种来装载图像。

10.3.7 Applet上下文

如果想在浏览器之间进行通信，那么需要applet调用getAppletContext方法。这个方法将返回一个实现了AppletContext接口的对象。可以将AppletContext接口的具体实现认为是打开了一条applet与环境浏览器之间的通信道路。

1. Applet间的通信

一个网页可以包含多个applet。如果网页中包含多个来自于同一个codebase的applet，则它们之间就可以互相通信。

如果为HTML文件中的每个applet都指定一个name属性，就可以使用AppletContext接口中的getApplet方法来得到对applet的引用。例如，若HTML文件包含下列标记：

<applet code=”Chart.class” width=”100” height=”100” name=”Chart1”>

则调用：

Applet chart1 = getAppletContext().getApplet(“Chart1”);

就会得到对applet的引用。如果Chart类包含一个利用获得的新数据重新绘制图表的方法，那么就可以在进行相应的类型转换后调用这个方法〉。

((chart) chart1).setData(3, “Earth”, 9000);

还可以把所有的applet都列在网页上，而不管它们是否有name属性。getApplets方法返回一个枚举对象（enumeration object）。下面的循环可以打印出当前页面中包含的全部applet的类名。

Enumberation<Applet> e = getAppletContext().getApplets();

while (e.hasMoreElements()) {

Applet a = e.nextElement();

System.out.println(a.getClass().getName());

}

Applet不能与其他网页上的applet进行通信。

1. 在浏览器中显示信息

通过使用AppletContext类的方法可以访问环境浏览器的两个区域：状态行和网页显示区域。

可以使用showStatus消息在浏览器底部的状态行中显示一个字符串，例如：

showStatus(“Loading data … please wait”);

**提示：**经验证明，showStatus方法有一定的局限性。这是因为浏览器也经常使用状态行，并会用Applet running之类的话覆盖住applet在状态行上显示的有用信息，所以，应该在状态行中显示不太重要的信息，例如，“Loading data … please wait”，而还要显示对用户来说非常重要的信息。

利用showDocument方法，可以请求浏览器显示不同的网页。最简单的一种方法是调用只需要提供一个参数的showDocument方法，其参数就是要显示的URL：

URL u = new URL(<http://java.sun.com/index.html>);

getAppletContext().showDocument(u);

这种调用方式的问题在于新打开的页面将会出现在当前网页的窗口中。因此，applet会被替换掉。

如果在调用showDocument方法时提供第二个参数，就可以告诉浏览器在另一个窗口中显示文档，请参看10－2。

表10－2 showDocument方法

|  |  |
| --- | --- |
| 目标参数 | 位置 |
| “\_self”或者none | 在当前框架中显示文档 |
| “\_parent” | 在父框架中显示文档 |
| “\_top” | 在最顶层框架中显示文档 |
| “\_blank” | 在新的、未命名的、顶层窗口中显示文档 |
| 其他字符串 | 在指定名称的框架中显示文档。如果没有相应名称的框架，就打开一个新窗口，并给这个窗口指定名称 |

1. 既是applet，又是应用程序

GUI应用程序会构造一个框架对象，并调用setVisible(true)方法。由于不能在光秃秃的applet上调用setVisible，所以必须将applet放在框架中，并在AppletFrame类提供的构造器中将applet添加到内容空格中。

public class AppletFrame extends JFrame {

public AppletFrame(Applet anApplet) {

applet = anApplet;

add(applet);

….

}

…

}

在applet应用程序的make方法，构造并显示AppletFrame.

class MyAppletApplication extends MyApplet {

public static void main(String args[]) {

AppletFrame frame = new AppletFrame(new MyApplet());

frame.setTitle(“MyAppletApplication”);

frame.setSize(DEFAULT\_WIDTH, DEFAULT\_HEIGHT);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

}

}

注意，直接扩展于已存在的applet。

在applet启动时，必须调用它的init方法和start方法。通过覆盖AppletFrame类中的setVisible方法可以达到这个目的：

public void setVisible(boolean b) {

if (b) {

applet.init();

super.setVisible(true);

applet.start();

} else {

applet.stop();

super.setVisible(false);

applet.destroy();

}

}

但有一个问题。如果程序以应用程序的方式启动，并且调用getAppletContext方法，就会得到一个null指针，其原因是这个程序不能在浏览器中启动。这就需要实现两个接口：AppletStub和AppletContext。AppletStub接口的主要用途是定位applet的上下文。下面提供了实现这两个接口的最小的基本功能：

public class AppletFrame extends JFrame implements AppletStub, AppletContext {

…

// AppletStub methods

public Boolean isActive() { return true;}

public URL getDocumentBase() { return null; }

public URL getCodeBase() { return null; }

public String getParamter(String name) { return “”; }

public AppletContext getAppletContext() { return this; }

public void appletResize(int width, int height) { }

// AppletContext methods

public AudioClip getAudioClip(URL uri) { return null; }

public Image getImage(URL url) {

return ToolKit.getDefaultToolkit().getImage(url);

}

public Applget getApplet(String name) { return null; }

public Enumeration<Applet> getApplets() { return null; }

public void showDocument(URL url) {}

public void showDocument(URL url, String target) {}

public void showStatus(String status) {}

public void setStream(String key, InputStream stream) {}

public InputStream getStream(String key) { return null; }

public Iterator<String> getStreamKeys() { return null; }

}

接下来， 框架类的构造器调用applet的setStub方法，以便将applet设定为自己的桩.

public AppletFrame(Applet anApplet) {

applet = anApplet;

Container contentPane = getContentPane();

contentPane.add(applet);

applet.setStub(this);

}