



Java GUI

Allen Long

Email: allen@huihoo.com

<http://www.huihoo.com>

2004-04

概述



在 Java 2 平台发行之之前，抽象窗口工具包（Abstract Window Toolkit，AWT）是 Java 平台的图形功能的范围。例如 Swing 等各种技术都是作为可选的扩展被引入的。在 Java 2 平台上，这些扩展中的大部分都可以在平台的核心中找到它们的一席之地，成为 Java 基础类（Java Foundation Classes，JFC）的一部分。JFC 指的是包含在 Java 2 平台内的一整套图形和用户界面技术，包括 AWT 和 Swing 等。

JFC(Java Foundation Classes)



包括以下API:

- AWT
- Java 2D
- Accessibility
- Drag and Drop
- Swing



JDK1.4.1的JFC



运行JFC的例子

C:\j2sdk1.4.1\demo\jfc

JFC的核心: Swing



Swing 是一个带有一套丰富组件的 GUI 工具包，它组成了 JFC 的用户界面功能的核心部分。它不仅是 AWT 所提供的组件的替代品，并且在这些组件的基础上有了很大进步。

先看看Swing的能力吧

运行JAVA 2D例子

C:\j2sdk1.4.1\demo\jfc\swingset2

Java -jar swingset2.jar



AWT (Abstract Window Toolkit)



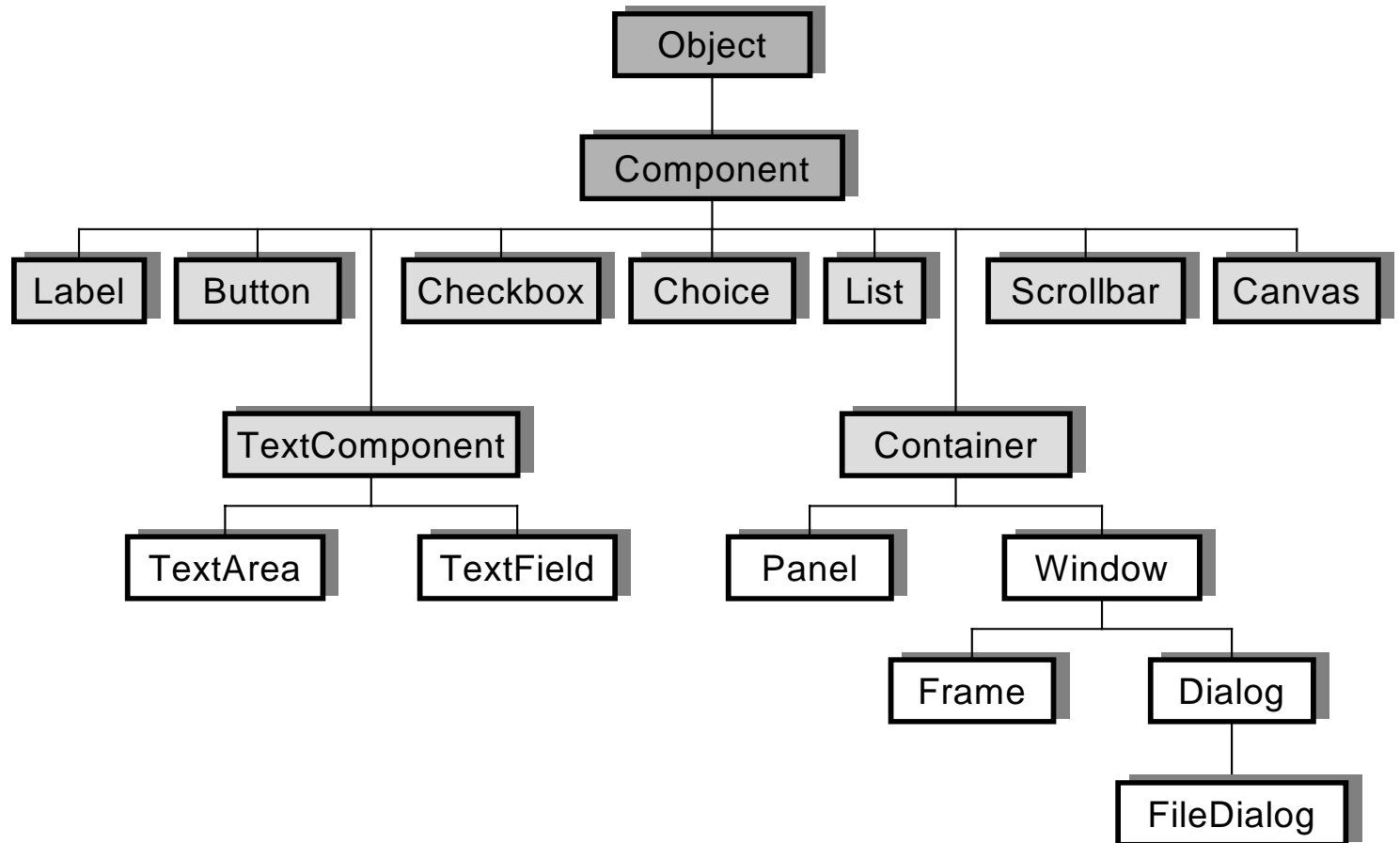
AWT提供用于所有Java applets及应用程序中的基本GUI组件，还为应用程序提供与机器的界面。这将保证一台计算机上出现的东西与另一台上的相一致。

在学AWT之前，简单回顾一下对象层次。记住，超类是可以扩展的，它们的属性是可继承的。而且，类可以被抽象化，这就是说，它们是可被分成子类的模板，子类用于类的具体实现。

显示在屏幕上的每个GUI组件都是抽象类组件的子类。也就是说，每个从组件类扩展来的图形对象都与允许它们运行的大量方法和实例变量共享。

Container是Component的一个抽象子类，它允许其它的组件被嵌套在里面。这些组件也可以是允许其它组件被嵌套在里面的容器，于是就创建了一个完整的层次结构。在屏幕上布置GUI组件，容器是很有用的。Panel是Container的最简单的类。Container的另一个子类是Window。

Java.awt包



Container



- 。 Container的两个主要类型是Window和Panel
- 。 Window是Java.awt.Window.的对象
- 。 Panel是Java.awt.Panel的对象
- Window有两种形式：Frame(框架)和Dialog（对话框）。Frame和Dialog是Window的子类。Frame是一个带有标题和缩放角的窗口。对话没有菜单条。尽管它能移动，但它不能缩放。
- Panel是Java.awt.Panel的对象。Panel包含在另一个容器中，或是在Web浏览器的窗口中。Panel确定一个四边形，其它组件可以放入其中。Panel必须放在Window之中（或Window的子类中）以便能显示出来。



Frames



- 是Window的子类
- 具有标题和缩放角
- 从组件继承并以add方式添加组件
- 能以字符串规定的标题来创建不可见框架对象
- 能将Border Layout当做缺省布局管理器
- 用setLayout方式来改变缺省布局管理器



第一个Frame 例子



```
1.  import java.awt.*;
2.  public class MyFrame extends Frame {
3.  public static void main (String args[]) {
4.          MyFrame fr = new MyFrame("Hello Frame!");
5.          fr.setSize(500,500);
6.          fr.setBackground(Color.yellow);
7.          fr.setVisible(true);
8.      }
9.  public MyFrame (String str) {
10.      super(str);
11.  }
12. }
```

● 运行这个例子：



参考：

Using Swing Components: Examples (1.4)

<http://java.sun.com/docs/books/tutorial/uiswing/components/example-1dot4/index.html>



Panels



- 为组件提供空间
- 允许子面板拥有自己的布局管理器
- 以add方法添加组件



Panels例子



创建了一个小的黄色面板，并将它加到一个框架对象上

```
import java.awt.*;
public class FrameWithPanel extends Frame {
    // Constructor
    public FrameWithPanel (String str) {
        super (str);
    }
    public static void main (String args[]) {
        FrameWithPanel fr =
            new FrameWithPanel ("Frame with Panel");
        Panel pan = new Panel();

        fr.setSize(200,200);
        fr.setBackground(Color.blue);
        fr.setLayout(null); //override default layout mgr
        pan.setSize (100,100);
        pan.setBackground(Color.yellow);

        fr.add(pan);
        fr.setVisible(true);
    }
}
```

运行这个例子：



布局管理器



- Flow Layout—Panel和Applets的缺省布局管理器
- Border Layout—Window、Dialog及Frame的缺省管理程序
- Grid Layout
- Card Layout
- GridBag Layout



Flow Layout的一个简单例子



```
import java.awt.*;
public class FrameWithPanel extends Frame {
    // Constructor
    public FrameWithPanel (String str) {
        super (str);
    }
    public static void main (String args[]) {
        FrameWithPanel fr =
            new FrameWithPanel ("Frame with Panel");
        Panel pan = new Panel();

        fr.setSize(200,200);
        fr.setBackground(Color.blue);
        fr.setLayout(null); //override default layout mgr
        pan.setSize (100,100);
        pan.setBackground(Color.yellow);

        fr.add(pan);
        fr.setVisible(true);
    }
}
```

运行这个例子：



使用JFC/Swing创建GUI



功能	描述
Swing GUI Components	Includes everything from buttons to split panes to tables.
Pluggable Look-and-Feel Support	Gives any program that uses Swing components a choice of look and feel. For example, the same program can use either the Java or the Windows look and feel. Many more look-and-feel packages are available from various sources. As of v1.4.2, the Java platform supports the GTK+ look and feel, which makes hundreds of existing look and feels available to Swing programs.
Accessibility API	Enables assistive technologies, such as screen readers and Braille displays, to get information from the user interface.
Java 2D API	Enables developers to easily incorporate high-quality 2D graphics, text, and images in applications and applets. Java 2D includes extensive APIs for generating and sending high-quality output to printing devices.
Drag-and-Drop Support	Provides the ability to drag and drop between Java applications and native applications.
Internationalization	Allows developers to build applications that can interact with users worldwide in their own languages and cultural conventions. With the input method framework developers can build applications that accept text in languages that use thousands of different characters, such as Japanese, Chinese , or Korean.

Swing API



- Swing API有17个包

`javax.accessibility`

`javax.swing`

`javax.swing.border`

`javax.swing.colorchooser`

`javax.swing.event`

`javax.swing.filechooser`

`javax.swing.plaf`

`javax.swing.plaf.basic`

`javax.swing.plaf.metal`

`javax.swing.plaf.multi`

`javax.swing.table`

`javax.swing.text`

`javax.swing.text.html`

`javax.swing.text.parser`

`javax.swing.text.rtf`

`javax.swing.tree`

`javax.swing.undo`

常用

- `javax.swing`
- `javax.swing.event` (not always required)



Swing例子



```
import javax.swing.*;
```

```
public class HelloWorldSwing {  
    private static void createAndShowGUI() {  
        JFrame.setDefaultLookAndFeelDecorated(true);  
        JFrame frame = new JFrame("HelloWorldSwing");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JLabel label = new JLabel("Hello World");  
        frame.getContentPane().add(label);  
        frame.pack();  
        frame.setVisible(true);  
    }  
  
    public static void main(String[] args) {  
        javax.swing.SwingUtilities.invokeLater(new Runnable() {  
            public void run() {  
                createAndShowGUI();  
            }  
        });  
    }  
}
```

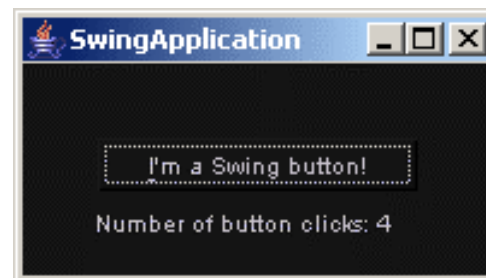
运行这个例子：



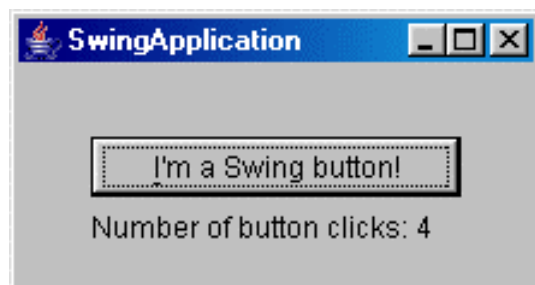
Look and Feel



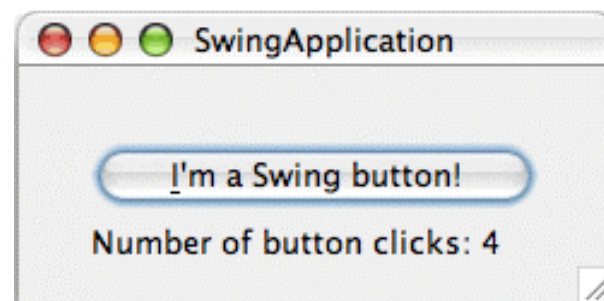
Java look and feel



GTK+ look and feel



Windows look and feel



Mac OS look and feel



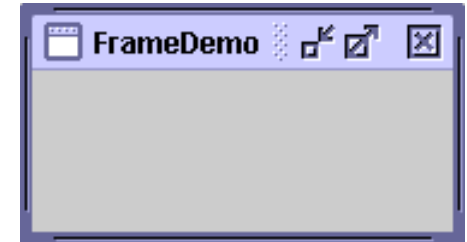
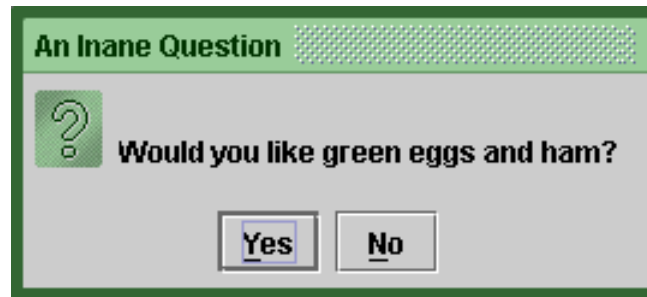


使用Swing组件

Top-Level Containers



The components at the top of any Swing containment hierarchy.



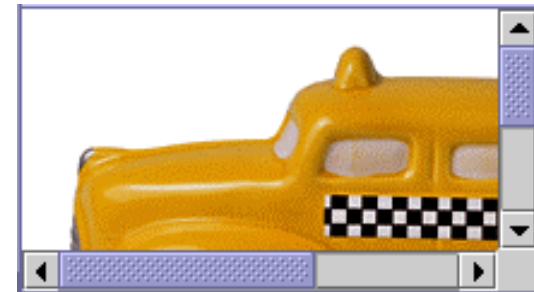
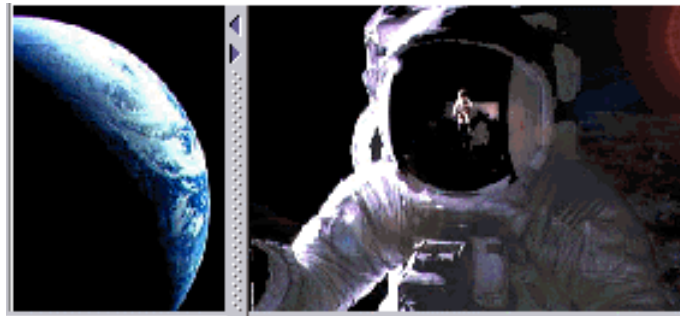
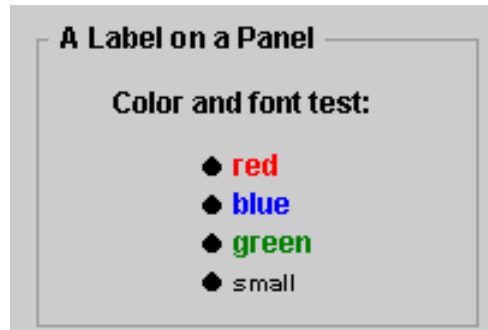
演示 : `c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar`



General-Purpose Containers



Intermediate containers that can be used under many different circumstances.

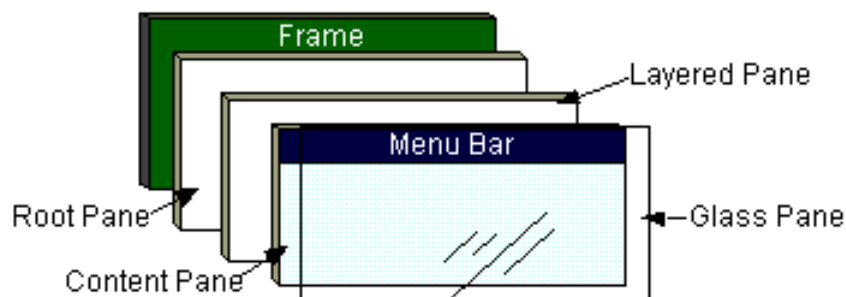
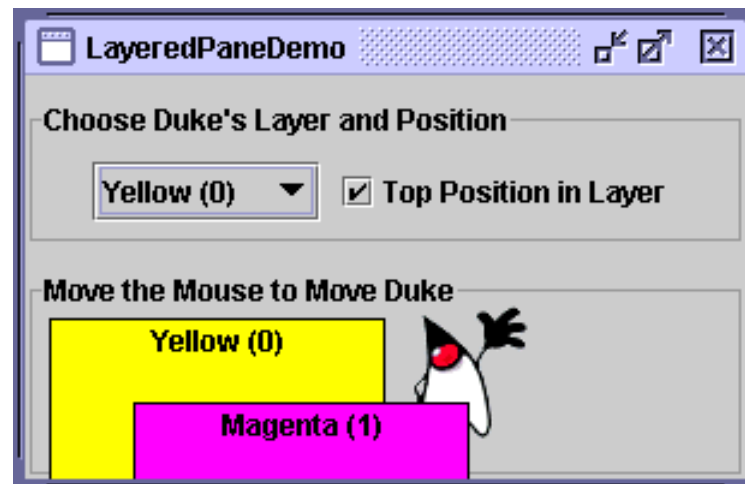
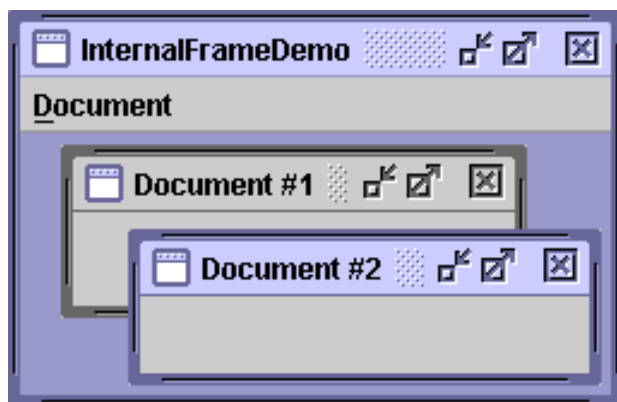


演示 : `c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar`

Special-Purpose Containers



Intermediate containers that play specific roles in the UI.



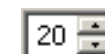
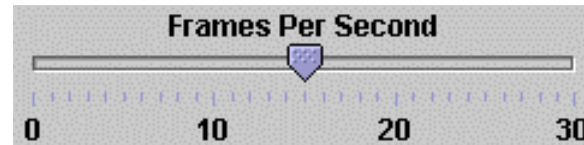
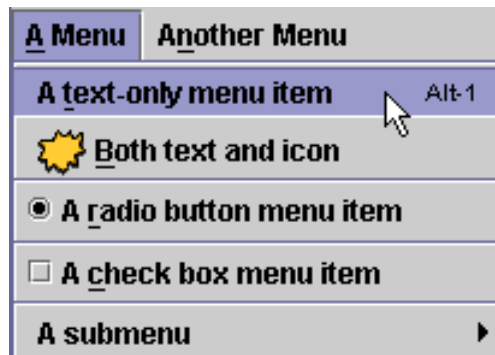
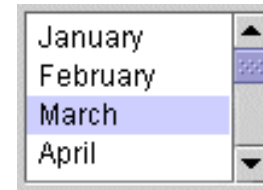
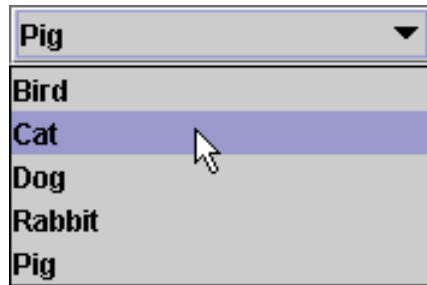
演示 : `c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar`



Basic Controls



Atomic components that exist primarily to get input from the user; they generally also show simple state.

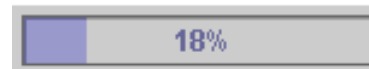
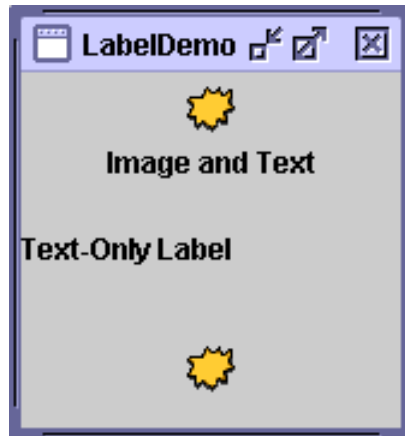


演示 : `c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar`

Uneditable Information Displays



Atomic components that exist solely to give the user information.



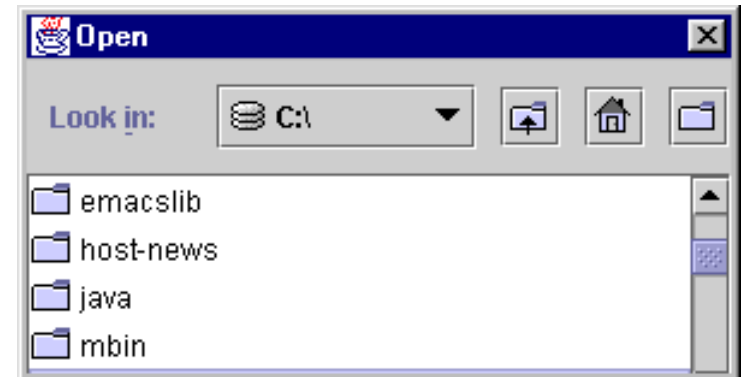
演示 : `c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar`



Interactive Displays of Highly Formatted Information

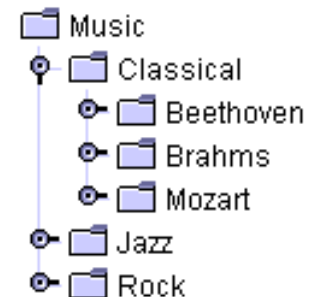


Atomic components that display highly formatted information that (if you choose) can be modified by the user.



- ◆ red
- ◆ blue
- ◆ green
- ◆ small
- ◆ large
- ◆ *italic*
- ◆ **bold**

First Name	Last Name	Favorite Food
Jeff	Dinkins	
Ewan	Dinkins	
Amy	Fowler	
Hania	Gajewska	
David	Geary	



演示 : `c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar`

Drag and Drop



Drag and Drop enables data transfer both across Java™ programming language and native applications, between Java programming language applications, and within a single Java programming language application.



独立于平台的拖放



。JDK 1.1 增加了一个通用机制，这种机制放在 `java.awt.datatransfer` 包中，实现了应用程序之间和应用程序内部的数据传输，以及对系统剪贴板的操作能力。

。Java 2 版本引入了 `java.awt.dnd` 包。此包建立在数据传输的机制上，提供了拖放功能，这些功能可以在一个单独的 Java 应用程序内或两个 Java 应用程序之间以一种平台无关的方式执行。为了集成本机平台的拖放功能，它也可以以一种平台 相关的方式运行。

。使用拖放（Drag and Drop，DND）API 很具有挑战性，因为它要操作在很高的抽象度上来支持它工作的不同方式，同时也因为它被设计成操作被 `java.awt.datatransfer.Transferable` 接口所指定的任意数据类型。



DND API 如何工作



。首先，需要一个 `DragSource` 对象来初始化拖放操作。该对象被用于创建一个 `DragGestureRecognizer`，它规定标明拖动操作开始点的事件。例如，在一个特定组件上按下鼠标左键然后移动鼠标。

。系统一识别到此动作，可利用 `DragSource` 对象通过要传输的信息来定义操作的开始点。

。下一步，拖动操作的有效目的地可通过向 `DropTarget` 对象传送一个组件或者调用 `Component` 的新的 `setDropTarget()` 方法来定义。

。实现 `DropTargetListener` 接口的类可用来响应拖放操作中可能出现的各种事件。主要事件是放操作，一个 `DropTargetDropEvent` 对象被传送给该操作。可向该对象查询被拖动的数据的类型。

。侦听器可以拒绝或者接受拖动并传输数据，完成拖放操作。



Swing Data Transfer



The [JLabel](#) component does not support DnD by default. The following code fragment creates a JLabel that supports drag and drop on the "text" property (a String):

```
JLabel componentType = new JLabel();  
componentType.setTransferHandler(new  
TransferHandler("text"));  
MouseListener ml = new MouseAdapter() {  
    public void mousePressed(MouseEvent e) {  
        JComponent c = (JComponent)e.getSource();  
        TransferHandler th = c.getTransferHandler();  
        th.exportAsDrag(c, e, TransferHandler.COPY);  
    }  
};  
componentType.addMouseListener(ml);
```



演示 : c:\j2sdk1.4.1\demo\jfc\SwingSet2\java -jar SwingSet2.jar

Java 2D



The Java 2D API introduced in JDK 1.2 provides enhanced two-dimensional graphics, text, and imaging capabilities for Java programs through extensions to the Abstract Windowing Toolkit (AWT). This comprehensive rendering package supports line art, text, and images in a flexible, full-featured framework for developing richer user interfaces, sophisticated drawing programs and image editors.

Java 2D API提供：

- A uniform rendering model for display devices and printers
- A wide range of geometric primitives, such as curves, rectangles, and ellipses and a mechanism for rendering virtually any geometric shape
- Mechanisms for performing hit detection on shapes, text, and images
- A compositing model that provides control over how overlapping objects are rendered
- Enhanced color support that facilitates color management
- Support for printing complex documents





java.awt

Class Graphics2D

java.lang.Object

--java.awt.Graphics

--java.awt.Graphics2D

```
public void Paint (Graphics g) {  
    Graphics2D g2 = (Graphics2D) g;  
    ...  
}
```



Displaying Graphics with Graphics2D



```
import java.awt.*;
import java.awt.geom.*;

public class j2d extends Frame {
    public static void main(String args[]) { new j2d(); }
    public j2d() {
        setSize(220,180);
        setVisible(true);
    }
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D)g;

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
            RenderingHints.VALUE_ANTIALIAS_ON);

        GradientPaint gp = new GradientPaint(0, 60, Color.red,
            0, 120, Color.yellow);
        Ellipse2D r = new Ellipse2D.Float(30, 60, 160, 60);
        g2.setPaint(gp);
        g2.fill(r);

        g2.transform(AffineTransform.getRotateInstance(Math.PI/8));
        g2.setFont(new Font("Serif", Font.BOLD, 85));
        g2.setPaint(Color.blue);
        g2.setComposite(AlphaComposite.getInstance(
            AlphaComposite.SRC_OVER, 0.5f));
        g2.drawString("J2D",50,70);
    }
}
```



Displaying Graphics with Graphics2D



演示 : j2d.java



更多演示



演示 : `c:\j2sdk1.4.1\demo\jfc\Java2D\java -jar java2demo.jar`





方便残疾人士：辅助选项（Accessibility）

- 。Java 应用程序配备有 JFC Accessibility API（JFC 辅助选项 API），所以可以被具有各种能力的用户访问，包括视力、听力或行动不便的用户。这些可能包括不能分辨视觉或听觉信号或者不能操作定位设备。
- 。辅助选项支持的两大重要功能就是屏幕读取器和屏幕放大镜。屏幕读取器允许用户通过创建界面的 off-screen 表示，并将其传送到一个语音合成器或盲文终端来与 GUI 进行交互。屏幕放大镜提供一个屏幕的放大窗口，典型情况下是正常大小的 2 到 16 倍。它们通常不断跟踪指针的移动和输入焦点的改变，然后相应地调整放大的窗口。另外，象平滑字体这样的技术可能会用于生成更清晰的图像。

Java Accessibility Bridge（Java 辅助选项 Bridge）

- 。一些主机系统，如 Microsoft Windows，提供它们自己的辅助选项功能。缺省情况下，Java 应用程序并不完全支持它们。例如，对于本机应用程序，屏幕放大器探测输入焦点何时转换到另一个不同的用户界面组件（例如通过使用 Tab 键），然后调整放大的屏幕部分，来显示当前拥有输入焦点的组件。
- 。然而，Swing 应用程序使用的是轻量级的组件，操作系统将这些组件作为图像处理，而不作为离散的组件来处理。这就意味着屏幕放大器不能象本机应用程序那样在输入焦点中跟踪发生的改变。
- 。这正好就是面向 Windows 的 Java Accessibility Bridge 所要解决的问题。它在与轻量级组件有关的事件和本机系统事件之间建立一个映射。使用这个 Bridge，支持 Accessibility API 的 Java 应用程序就可以与 Windows 辅助选项支持完全集成。



Resources



- <http://java.sun.com/products/jfc/JFC/Swing>
- <http://java.sun.com/products/jfc/tsc/>
The Swing Connection
- <http://www-900.ibm.com/developerWorks/cn/java/j-j2int/index.shtml>

Java 2 用户界面





Q&A



Thank You

