



Indian Institute of Technology Kharagpur

---

## Java Applets – Part II

Prof. Indranil Sen Gupta  
Dept. of Computer Science & Engg.  
I.I.T. Kharagpur, INDIA



### Lecture 29: Java applets – Part II

On completion, the student will be able to:

1. Illustrate how a given java application can be converted into a java applet.
2. Explain the applet life cycle, with examples.
3. Illustrate the embedding of multiple applets in the same HTML page.
4. Demonstrate the execution of more complex applets.



## Converting Java Applications to Java Applets



### Introduction

- It is fairly straightforward to convert a graphical Java application into an applet.
- Why so?
  - Both the *Applet* and the *Frame* classes descend from *Container*.
  - Thus the same methods may be used to add the user interface components.



## Steps to be Followed

1. Create a HTML file with an **APPLET** tag which specifies the name of the applet class file, applet window size, and other relevant information.
2. Drop the **main** method.
  - In a Java application, the **main** method usually contains code to create a new frame object.
  - In an applet, however, creation of an applet object is done by the browser automatically.
  - The **main** method defines the frame size. For the applet, the size information is provided by the **WIDTH** and **HEIGHT** attributes of **APPLET** tag.



3. Instead of deriving the class from **Frame**, derive it from **Applet**.
4. Replace the constructor of the Java application with a method called **init()**.
  - After the browser creates an object of the applet class, the **init()** method is automatically called.
5. Take special care regarding the default layout manager.
  - Java applications use **BorderLayout** manager as default, while applets use **FlowLayout**.
  - The following must be included in **init()** method:  
**setLayout (new BorderLayout());**



6. Applets do not have title bars, and so any call to **setTitle** method must be omitted.
7. Add the following line at the beginning of the program:

```
import java.applet.* ;
```



## An Example

- We take an example to illustrate the conversion process.
- About the example:
  - A set of menu buttons are displayed in a frame.
  - The buttons are labeled with different colors which when clicked changes the background color to that value.



## An Example :: the application

```
import java.awt.*;
public class ButtonDemo extends Frame
{
    public ButtonDemo()
    {
        setTitle ("Button Demonstration");
        setLayout (new FlowLayout());

        add (new Button ("Red");
        add (new Button ("Blue");
        add (new Button ("Green");
        add (new Button ("White");
    }
}
```



```
public boolean handleEvent (Event v)
{
    if (v.id == Event.WINDOW_DESTROY)
        System.exit (0);
    return super.handleEvent (v);
}
public boolean action (Event v, Object but)
{
    if (but.equals ("Red"))           setBackground (Color.red);
    else if (but.equals ("Blue"))     setBackground (Color.blue);
    else if (but.equals ("Green"))    setBackground (Color.green);
    else if (but.equals ("White"))    setBackground (Color.white);
    else return false;
    repaint ();
    return true; }
}
```



```
public static void main (String args [ ])  
{  
    Frame f = new ButtonDemo ();  
    f.resize (300, 300);  
    f.show ();  
}
```



## The Applet :: after conversion

```
import java.awt.*;  
import java.applet.*;  
public class ColorApplet extends Applet  
{  
    public void init ()  
    {  
        setLayout (new FlowLayout());  
  
        add (new Button ("Red"));  
        add (new Button ("Blue"));  
        add (new Button ("Green"));  
        add (new Button ("White"));  
    }  
}
```



```
public boolean handleEvent (Event v)
{
    if (v.id == Event.WINDOW_DESTROY)
        System.exit (0);
    return super.handleEvent (v);
}
public boolean action (Event v, Object but)
{
    if (but.equals ("Red"))           setBackground (Color.red);
    else if (but.equals ("Blue"))     setBackground (Color.blue);
    else if (but.equals ("Green"))    setBackground (Color.green);
    else if (but.equals ("White"))    setBackground (Color.white);
    else return false;
    repaint ();
    return true; }
}
```



## Applet Life Cycle



## Introduction

- When writing an applet, it may be necessary to override methods in the Applet class.
- What we need to know?
  - The possible side effects of overriding.
  - When the methods are called.
  - What code should be placed inside the methods.



## About the Methods

- **init**
  - This method is invoked first.
  - All the initializations needed for the applet are done here.
  - Called only once when the applet is loaded.
- **start**
  - This is called after *init()*, and as a starting point after an applet was stopped (say, visit some page and come back again later).
  - *start()* is invoked every time the applet's HTML code is displayed on the screen.





- **paint**
  - Called every time the window is damaged.
- **update**
  - This first fills an applet window with the default background color, and then calls *paint()* .
- **stop**
  - Called when the browser moves to some other document.
  - Can be used to suspend time consuming activities (animation, threads, etc.) that need not be run when the applet is not visible.
  - The activities can be restarted by calling *start()* .



- **destroy**
  - Called when the browser determines that the applet needs to be removed completely from memory.
  - Can be used to release any resource which the applet may be using.



## The Life Cycle



## Multiple Applets on the Same Page



## Inter-applet Communication

- It is possible for a HTML page to have more than one applet.
  - These applets may interact with each other.
  - One applet may access the public variables and methods of other applets.
  - How?
    - By calling *getAppletContext()* method to communicate with the browser, which returns an object of type *AppletContext*.



- By giving *NAME* tags to each applet in the HTML document, we can use the *getApplet()* method of the *AppletContext* class in order to refer to the applet.
- An applet cannot communicate with another applet on a different web page.



## An Example

- The example considered is as follows:
  - There are two applets *GUI.class* and *Compute.class*, which have been assigned names *first* and *second* using the NAME attribute in HTML file.
  - The *GUI.class* applet reads in the amount in rupees, and the currency (dollar, say).
  - The *Compute.class* applet contains a method *convert (x, y)* to convert *x* rupees into currency type *y*.



## Example : the HTML file

```
<HTML>
  <TITLE> Currency Conversion Demo </TITLE>
  <BODY>

    <APPLET CODE = "GUI.class" NAME="first" WIDTH=300
              HEIGHT=200>

    </APPLET>

    <APPLET CODE="Compute.class" NAME="second"
              WIDTH=10 HEIGHT=10>

    </APPLET>
  </BODY>
</HTML>
```



## Example : the GUI.java applet

```
import java.awt.*;
import java.applet.*;

public class GUI extends Applet
{
    public void init()
    {
        Panel pan = new Panel();
        pan.setLayout (new.FlowLayout());

        pan.add (new Label ("Rupees"));
        TextField r = new TextField (" ", 8);
        pan.add (r);

        pan.add (new Label ("Currency"));
        TextField nc = new TextField (" ", 10);
        pan.add (nc);
    }
}
```



## Contd.

```
pan.add (new Label ("Result of conversion:"));
TextField result = new TextField (" ", 50);
pan.add (result);

add ("South", pan)
}

public boolean handleEvent (Event v)
{
    if (v.id == Event.WINDOW_DESTROY)
        System.exit (0);
    return super.handleEvent (v);
}
```



## Contd.

```
public boolean action (Event v, Object arg)
{
    int x = Format.atoi (r.getText());
    String y = nc.getText();
    Compute p = (Compute) getAppletContext . getApplet
("second");

    result.setText (r.getText().trim() + "Rupees = " + p.convert
(x, y) + nc.getText().trim());
}

private TextField r, nc;
}
```



## Example : the Compute.java applet

```
import java.awt.*;
import java.applet.*;
public class Compute extends Applet
{
    public void init()
    {
        Panel outp = new Panel();
        pan.setLayout (new.FlowLayout());
        outp.add (new Label ("Result of conversion:"));
        TextField result = new TextField (" ", 50);
        outp.add (result);
        add ("South", outp);
    }
}
```



## Contd.

```
public int convert (int rupee, String curr)
{
    if (curr .equals "dollar")
        return (rupee / 45);
    else if (curr .equals "baht")
        return (rupee / 1.2);
    else if (curr .equals "xyz")
        return (rupee * 11);
    }
}
```



## Accessing Image Files



## Introduction

- An image file can be retrieved from an applet using the *getImage()* method, which has two different forms.

```
Image getImage (URL a)
Image getImage (URL a, String path)
```

- First form gets an image file specified by the URL.
  - Second form uses the string to provide a path relative to the URL.
- Typical use:

```
Image logo = getImage (getDocumentBase(),
                        "images/logo.gif");
```



## An Example

- Displays two images alternately in response to some window event.

```
<HTML>
  <APPLET CODE = "ImageView.class"
          WIDTH = 650 HEIGHT = 300>
  <PARAM NAME = "M1" VALUE = "face1.gif">
  <PARAM NAME = "M2" VALUE = "face2.gif">
  </APPLET>
</HTML>
```





```
import java.awt.*;
import java.applet.*;
public class ImageView extends Applet
{
    boolean flag = true;
    Image x, y;
    public void init ( )
    {
        x = getImage (getDocumentBase(), getParameter("M1"));
        y = getImage (getDocumentBase(), getParameter("M2"));
    }
}
```



```
public void paint (Graphics g)
{
    if (flag)
    {
        g.drawImage (x, 0, 0, this);
        flag = false;
    }
    else
    {
        g.drawImage (y, 0, 0, this);
        flag = true;
    }
} }
```



## Accessing Audio Files



## Introduction

- An audio file can be retrieved using the ***getAudioClip()*** method, which has two different forms:

**AudioClip getAudioClip (URL a)**

**AudioClip getAudioClip (URL a, String path)**

- The first form gets an audio file from the location specified by the URL.
- The second form uses the string to provide a path relative to the URL in the first argument.



- To play an audio clip, we can use the ***play()*** method, which also has two different forms:

***void play (URL a)***

***void play (URL a, String path)***

with the same interpretation.



- Typical uses of ***play()*** :

```
AudioClip anthem = getAudioClip  
    (getDocumentBase(), "anthem.au");  
Anthem.play();
```

```
play (getDocumentBase(), "anthem.au")
```



---

# End of Lecture 29



---

## SOLUTIONS TO QUIZ QUESTIONS ON LECTURE 28



## Quiz Solutions on Lecture 28

---

### 1. Why is Java considered to be secure?

**For the following reasons:**

- **They cannot invoke any local executable.**
- **They cannot access the local file system.**
- **They can communicate with only the web server from where they were fetched.**
- **They cannot access any sensitive information on the local host**



## Quiz Solutions on Lecture 28

---

### 2. What makes Java programs platform independent?

**Java programs are compiled into a platform independent byte code format, which are then interpreted using JVM.**

**Versions of JVM exist on all platforms.**



## Quiz Solutions on Lecture 28

---

3. What is the difference between Java byte code and Java run time?

**Java byte code is the intermediate format generated by the compiler.**

**Java run time interprets the byte code.**

4. What is the difference between a Java application and a Java applet?

**Java application is a stand-alone program, while an applet is linked to a web page.**



## Quiz Solutions on Lecture 28

---

5. How can you pass parameters to an applet from the HTML page?

**Using the PARAM tag, with the NAME and VALUE attributes.**

**Parameters can be accessed by an applet using the getParameter() method.**



## Quiz Solutions on Lecture 28

---

6. Is it possible to invoke a method of some other applet from another applet on the same HTML page?

**Yes, by using the getApplet() method. A complete example was discussed today.**



## QUIZ QUESTIONS ON LECTURE 29



## Quiz Questions on Lecture 29

---

1. Why do we need to sometime convert a Java application into an applet?
2. What is the purpose of the `init()` method?
3. What is the purpose of the `start()` method?
4. What is the purpose of the `paint()` method?
5. How can an applet A invoke a method of applet B, where both A and B are included in the same HTML page?
6. How do you change the displayed image on an applet?