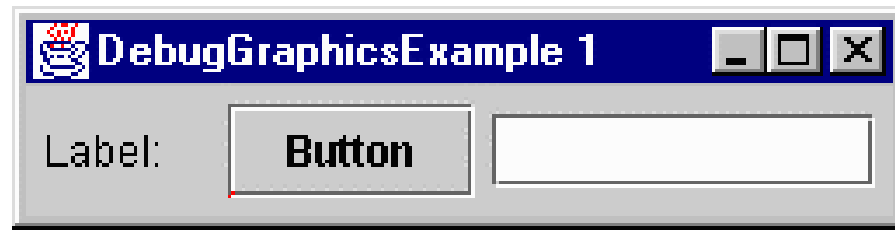


DebugGraphics

- Debugging utility included with Swing
- Allows you to watch or log graphics drawing as it occurs
- Since all rendering operations in Java are performed by operations on a Graphics object, it is possible to capture all of these operations and control they way they are performed by providing a subclass of Graphics that intercepts every operation on a component.
- Use on any JComponent, but must not be double buffered
- Use the setDebugGraphicsOptions(int option) method
 - DebugGraphics.FLASH_OPTION
 - DebugGraphics.LOG_OPTION
 - DebugGraphics.BUFFERED_OPTION
 - DebugGraphics.NONE_OPTION

DebugGraphicsExample1.java



More on DebugGraphics

- You can change the color, number of flashes and the duration of each flash with
 - `Public static void setFlashColor(Color aColor);`
 - `Public static void setFlashTime(int flashTime);`
 - `Public static void setFlashCount(int flashCount);`
- Logging goes to standard output, unless you tell it otherwise
 - `Public static void setLogStream(PrintStream stream);`
 - Note that `PrintStream` is deprecated in JDK1.1, but in JDK1.2 `PrintStream` has been “un-deprecated”

Logged DebugGraphics

- `cd c:/class/Fall 01/Section11_DebugGraphics/`
- `javaw DebugGraphicsExample1`

- Graphics(0-1) Enabling debug
- Graphics(1-1) Enabling debug
- Graphics(1-1) Translating by: `java.awt.Point[x=58,y=5]`
- Graphics(1-1) Setting new clipRect: `java.awt.Rectangle[x=0,y=0,width=71,height=26]`
- Graphics(1-1) Setting font: `javax.swing.plaf.FontUIResource[family=dialog,name=Dialog,style=bold,size=12]`
- Graphics(2-1) Enabling debug
- Graphics(5-1) Enabling debug
- Graphics(5-1) Setting color: `javax.swing.plaf.ColorUIResource[r=0,g=0,b=0]`
- Graphics(5-1) Setting font: `javax.swing.plaf.FontUIResource[family=dialog,name=Dialog,style=bold,size=12]`
- Graphics(7-1) Setting color: `javax.swing.plaf.ColorUIResource[r=204,g=204,b=204]`
- Graphics(7-1) Filling rect: `java.awt.Rectangle[x=0,y=0,width=71,height=26]`
- Graphics(7-1) Setting font: `javax.swing.plaf.FontUIResource[family=dialog,name=Dialog,style=bold,size=12]`
- Graphics(7-1) Setting color: `javax.swing.plaf.ColorUIResource[r=0,g=0,b=0]`
- Graphics(7-1) Drawing string: "Button" at: `java.awt.Point[x=17,y=18]`
- Graphics(7-1) Setting color: `javax.swing.plaf.ColorUIResource[r=153,g=153,b=204]`
- Graphics(7-1) Drawing rect: `java.awt.Rectangle[x=16,y=4,width=38,height=17]`
- Graphics(6-1) Translating by: `java.awt.Point[x=0,y=0]`
- Graphics(6-1) Setting color: `javax.swing.plaf.ColorUIResource[r=102,g=102,b=102]`

Current Bugs

- Note that with each use, the documentation states that the mask supplied is “or’d” with the current settings, however in early versions of JDK1.2 it merely uses the last mask provided.
- In JDK 1.2.1/2 and JDK 1.3 this works, but in JDK 1.4 it is broken again. Only the last command option is used, they are not “or’d” as it states in the API.
- BUFFERED_OPTION shows any off-screen buffer graphics in a separate window for items like JPanel which are double buffered. (However this option has not worked as of JDK1.2.1 thru 1.4B)