



the
POWER
of
JAVA™



Desktop Java™ Technology: Deep Dive

Thorsten Laux
Chet Haase
Oleg Sukholdosky
Scott Violet

Sun Microsystems
TS-1593

Copyright © 2006, Sun Microsystems Inc., All rights reserved.

2006 JavaOne™ Conference | Session TS-1593 |

java.sun.com/javaone/sf

Goal

Learn about Java™ technology's continued momentum on the desktop and how to take advantage of the new desktop features in Mustang

Agenda

State of Desktop Java Technology

Mustang

2D

AWT

Internationalization

Deployment

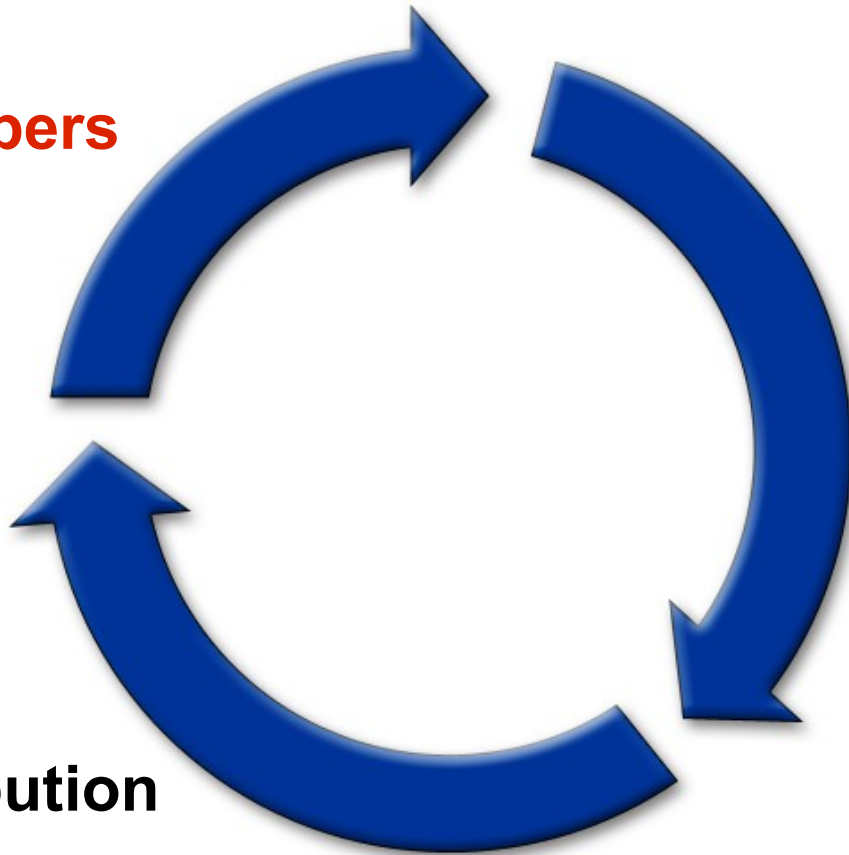
Swing

State of the Desktop Java Ecosystem

1.) Developers

2.) Applications

3.) Distribution



Developers—Adoption

- How do Java technology developers spend their time*
 - Desktop applications and Applets 41%
 - Servlets and EJB™ beans 37%
 - Mobile applications 4%
 - Other 18%
- Swing surpasses WinForms as the dominant GUI development toolkit*
- Which Java SE 6 features appeal to you most? **
 - Desktop enhancements 63.3%
 - Integrated web services 14.1%
 - Debugging enhancements 3.3%
 - All other (Security, Management, etc.) 19.3%

*Evans Data, 2005 **Java.net survey, 2006

Developers—Community

- Java.sun.com—Articles, Chats, Technical Tips
- Javadesktop.org
 - Desktop developer community site on java.net
 - 438 Projects—79 since January 1
 - Blogs, Articles, Swing Sightings, etc.
- Sun sponsored projects at Javadesktop.org
 - SwingLabs (<http://swinglabs.dev.java.net>)
 - Exploratory desktop Java projects
 - Incubator for future Java SE features
 - JavaMedia (Java 3D, Java Advanced Imaging, JOGL)
 - Project Looking Glass (Top java.net Project)

Developers—JDK Community

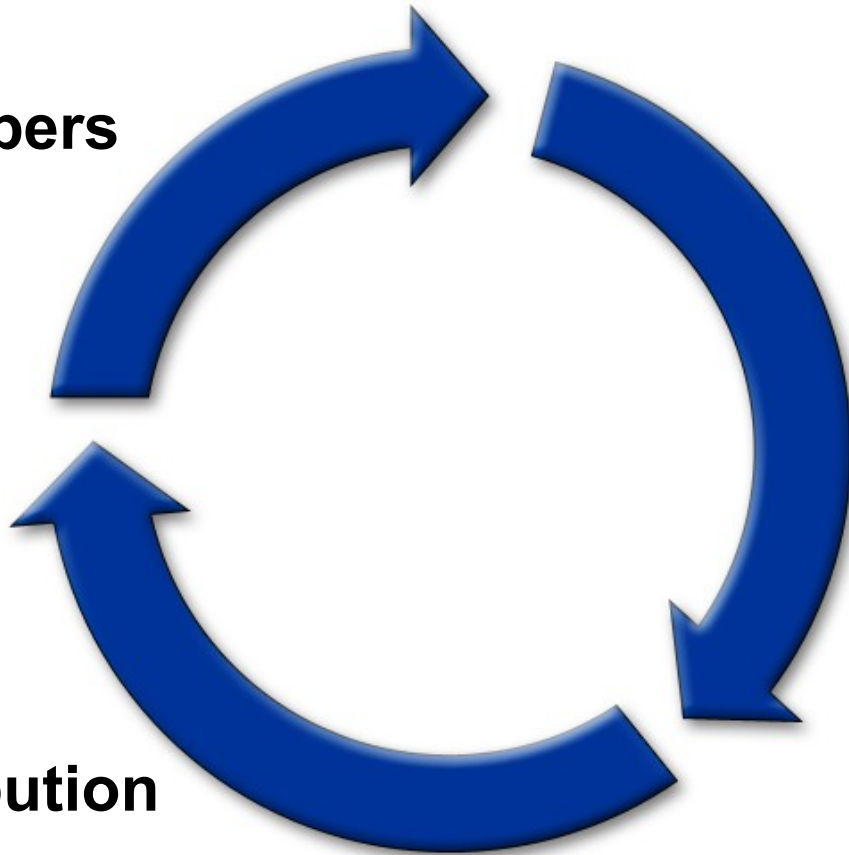
- Java SE community site (Forums, Blogs, etc.)
 - <http://community.java.net/jdk/>
 - 100,000+ visitors per month
- Weekly Java SE binary and source drops
 - 14,000 Downloads/month (average)
- Contribute features / bug fixes to Java SE
 - 1,100+ Bugfix forum posts
 - 750+ Java 6.0 bugs reported
 - 300+ Contributed bugfixes (200+ for the Java client)

State of the Desktop Java Ecosystem

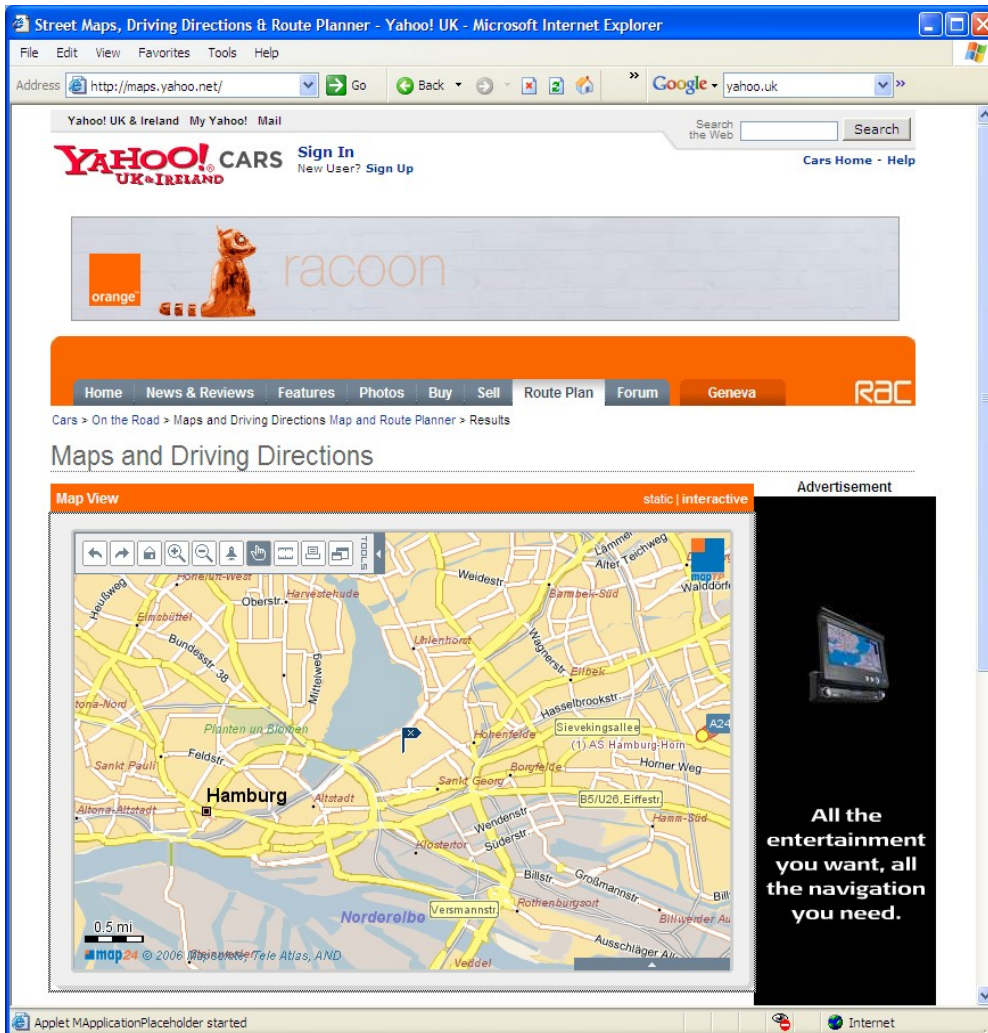
1.) Developers

2.) Applications

3.) Distribution

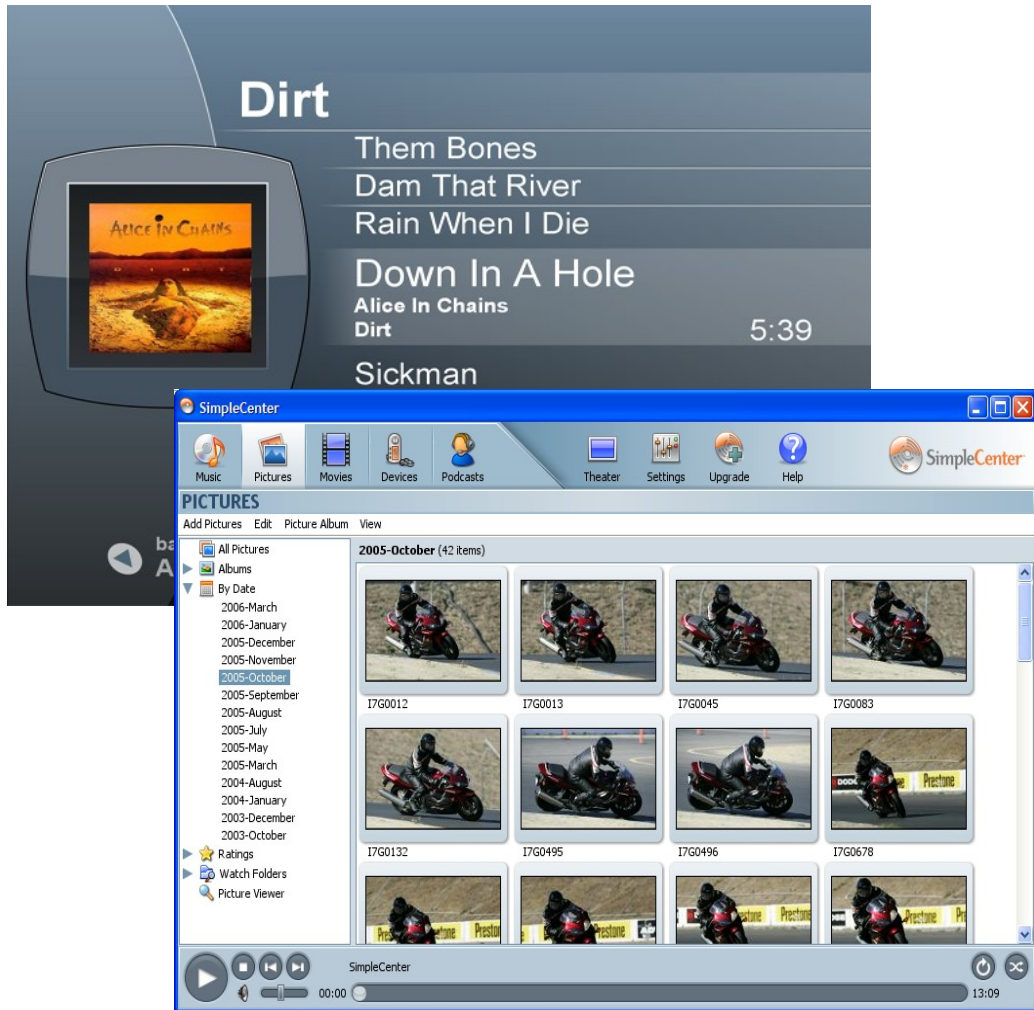


Applications: Map24



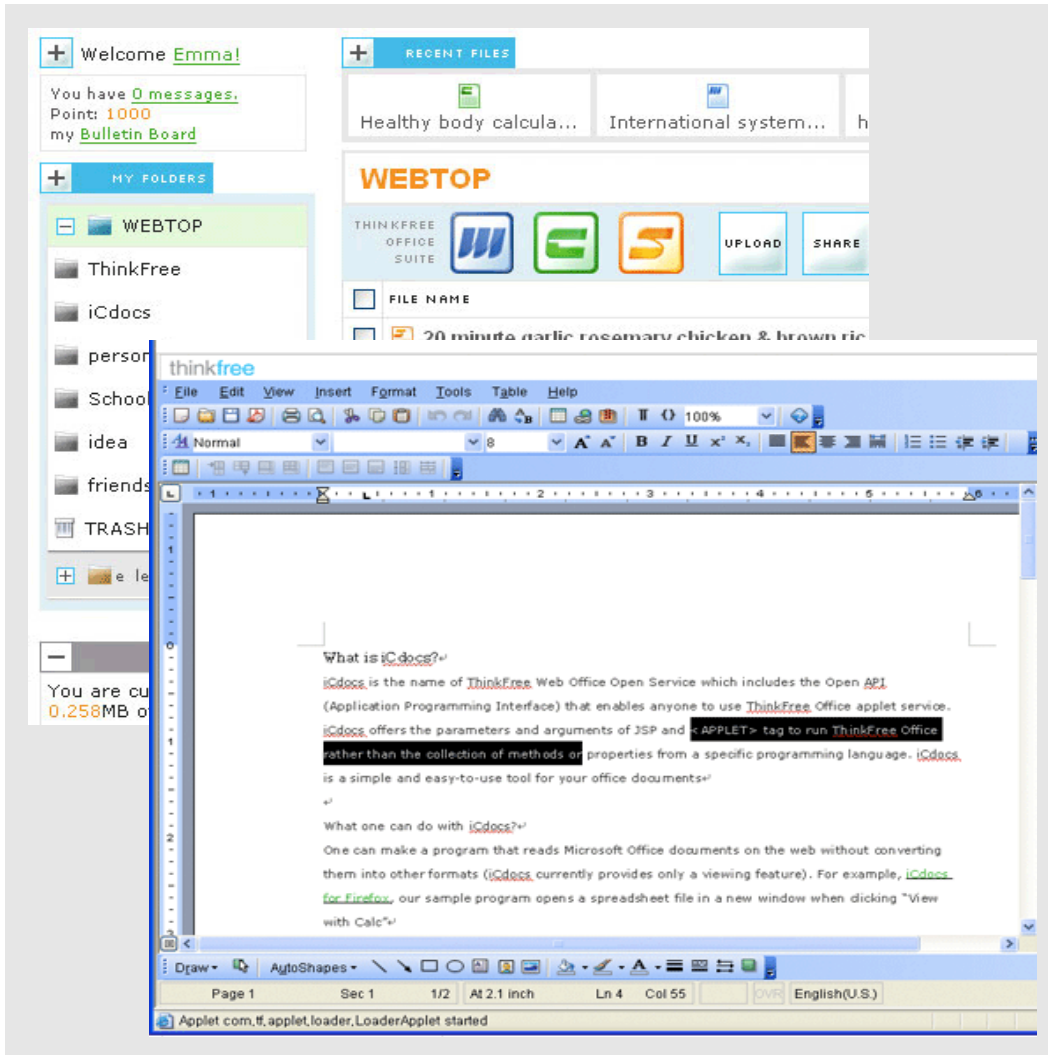
- Maps, Directions, Local Search as Java Applet
- Used for Maps, Directions and Local Search for Yahoo! Europe
- Up to 1 million Applets delivered to end users everyday

Applications: SimpleCenter



- Media management/ P2P sharing application
- Sharing of music, pictures, movies between PCs and uPnP devices
- Bundled with Phillips Streamium
- Bundled with every Nokia N80 phone

Applications: ThinkFree



- Browser based office suite
- Access from anywhere via Applet mode
- Central document storage
- Document sharing
- Blogging

Applications: Tribal Trouble



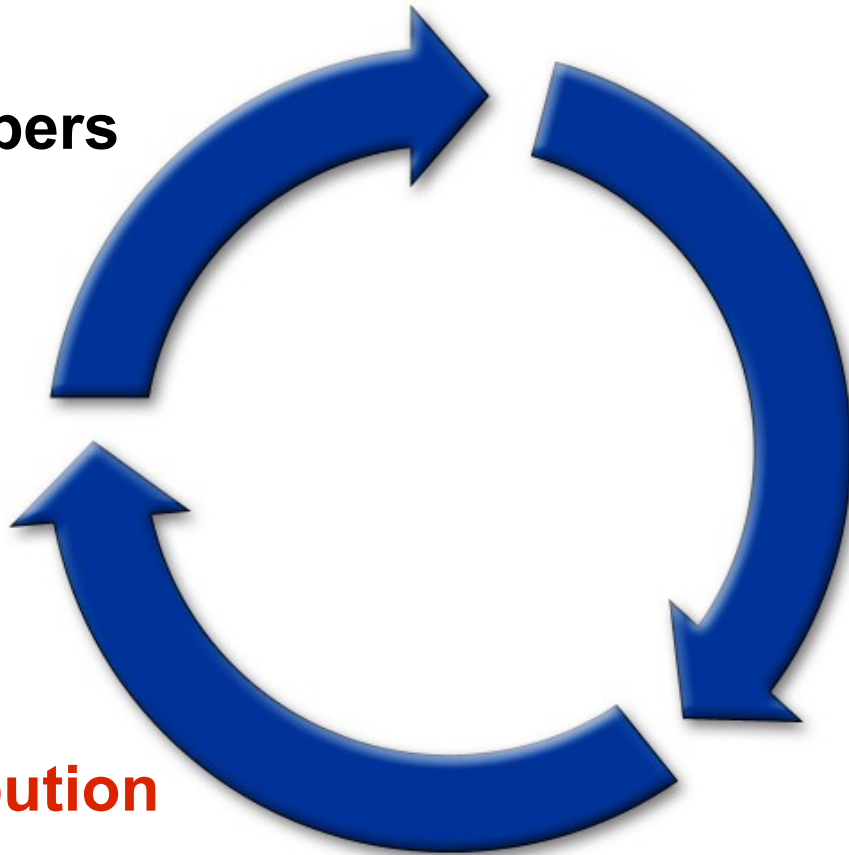
- Realtime multi-user strategy game
- Independent Games Festival 2006 finalist
- GameTunnel: Second best strategy game in 2005
- Quote: “The smartest thing was without a doubt choosing Java as a development platform”

State of the Desktop Java Ecosystem

1.) Developers

2.) Applications

3.) Distribution

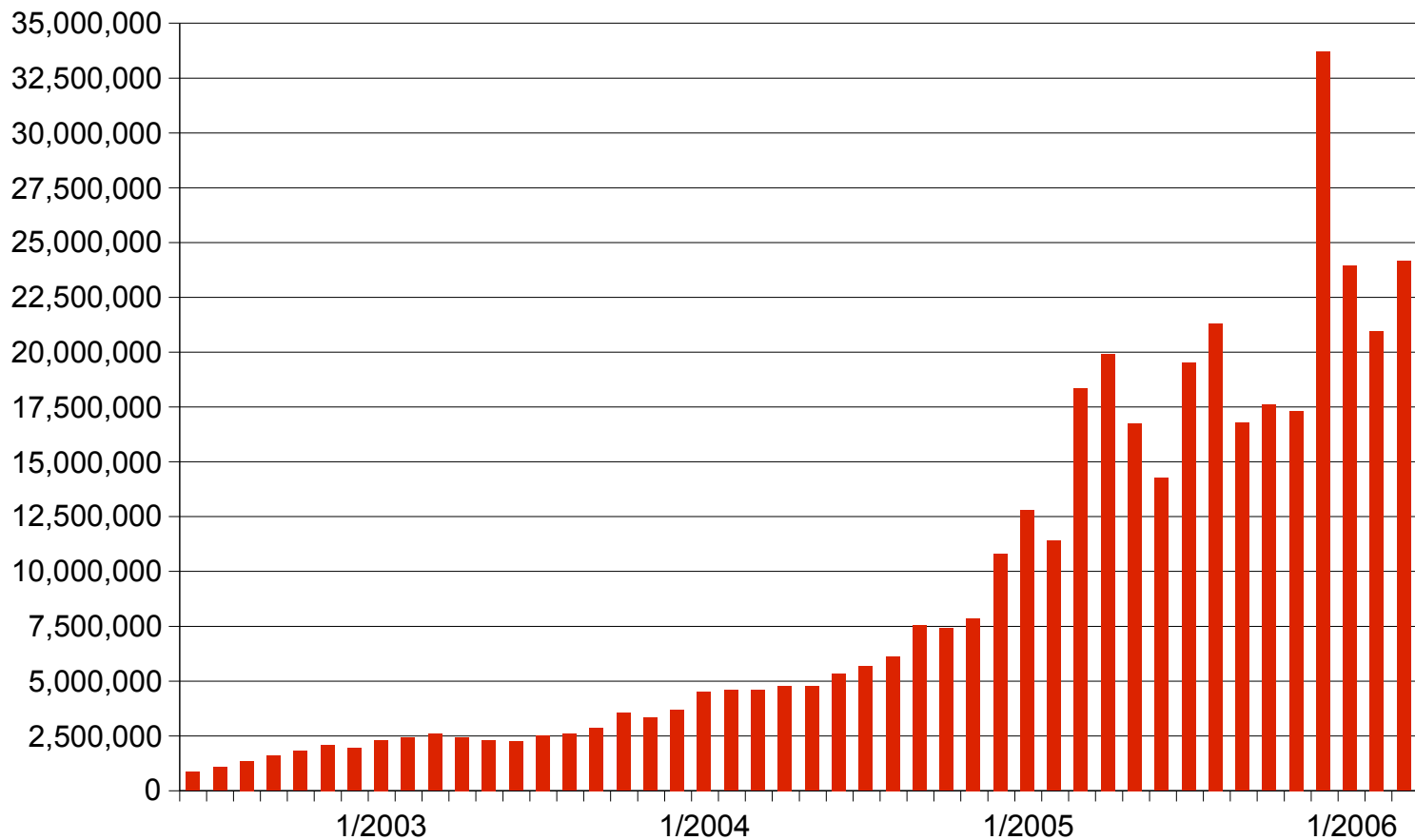


Distribution

- ~87% of all PCs run Java
- ~65% of all Java enabled PCs run Java SE
- Distribution through PC OEMs
 - All top ten PC OEMs ship the JRE
 - Representing >60% of all shipped PCs
 - New Windows PC OEM licensing program
 - New Linux Distribution licensing program
- Downloads from java.com
 - Monthly Windows JRE downloads >30M in 1/06

Completed Java SE Platform Downloads

Windows Numbers Only



Agenda

State of Desktop Java Technology

Mustang

2D

AWT

Internationalization

Deployment

Swing

Java 2D™ API in Mustang

Overview

- Sub-pixel text, aka LCD text
- LinearGradientPaint, RadialGradientPaint
- Faster ImageIO image loading
- Better small curve quality
- More Mustang 2D features, not covered here
 - Improved OpenGL® pipeline performance
 - Linux fullscreen support
 - New Image I/O plugin: GIF writer
 - New applet animation

Sub-Pixel Text

Smoother Text

- Plain



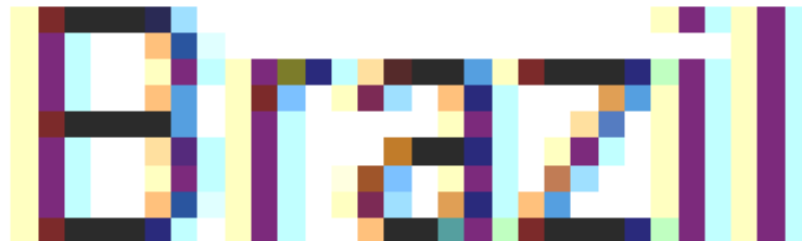
Brazil

- Antialiased



Brazil

- Sub-pixel



Brazil

Sub-Pixel Text: The Geeky Details

<http://today.java.net/pub/a/today/2005/07/26/lcdtext.html>

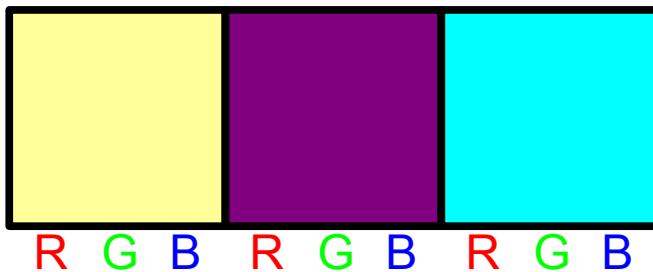
Plain:



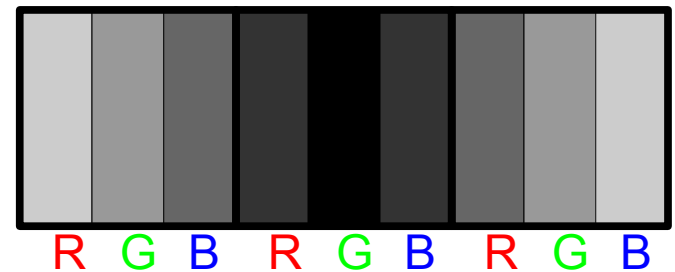
Anti-Aliased:



Sub-Pixel AA:



We see
as this →



- Our eyes combine the colors to get the same effect as a smooth ramp through the sub-pixels

Sub-Pixel Text: How to Use It

http://weblogs.java.net/blog/chet/archive/2005/06/phils_font_fixe.html

- In most cases, it will just work
 - Swing picks up desktop and font settings
 - Standard components will use same font rendering properties as native applications
- Custom components/rendering
 - `java.awt.RenderingHints.KEY_TEXT_ANTIALIASING`
- In settings where desktop properties do not exist (e.g., Windows 2000)
 - `-Dawt.useSystemAAFontSettings=lcd`

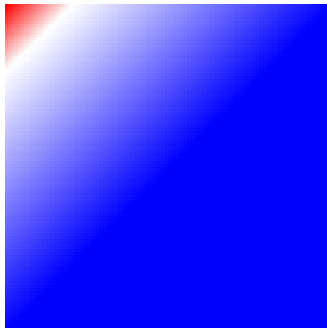
New Gradients

- **MultipleGradientPaint**
 - Abstract for gradients with multiple colors
 - Superclass for gradients below
 - CycleMethod for areas outside endpoints
 - NO_CYCLE: Extend color at endpoints outside area
 - REFLECT: start-end, end-start
 - REPEAT: start-end, start-end
- **LinearGradientPaint**
- **RadialGradientPaint**

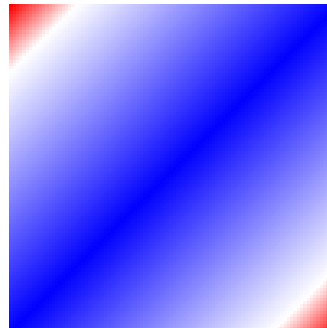
LinearGradientPaint

- Values from 0-1 determine where colors change
- Colors for each of those values
- Start and end points on Shape
- CycleMethod

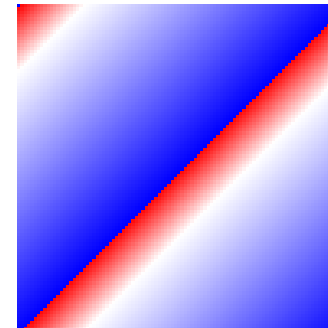
NO_CYCLE:



REFLECT:



REPEAT:



LinearGradientPaint

```
Point2D start = new Point2D.Float(0, 0);
Point2D end = new Point2D.Float(getWidth(), 0);
float[] dist = {0.0f, 0.2f, 1.0f};
Color[] colors = {Color.RED, Color.WHITE, Color.BLUE};
LinearGradientPaint p =
    new LinearGradientPaint(start, end, dist, colors);
g2d.setPaint(p);
g2d.fillRect(0, 0, getWidth(), getHeight());
```

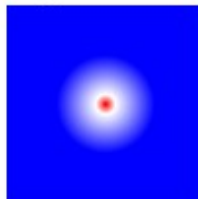


RadialGradientPaint

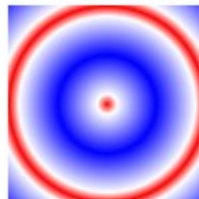
- Similar to LinearGradientPaint, only the gradient proceeds in a circle from the start point

```
Point2D center = new Point2D.Float(50, 50);  
float radius = 25;  
float[] dist = {0.0f, 0.2f, 1.0f};  
Color[] colors =  
    {Color.RED, Color.WHITE, Color.BLUE};  
RadialGradientPaint p =  
    new RadialGradientPaint(center, radius,  
                            dist, colors);
```

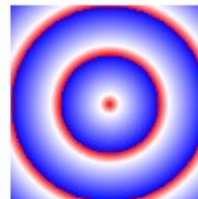
NO_CYCLE:



REFLECT:



REPEAT:

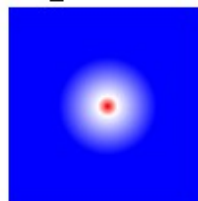


RadialGradientPaint

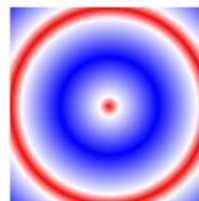
- Can also have focus that differs from center

```
Point2D center = new Point2D.Float(50, 50);  
float radius = 25;  
Point2D focus = new Point2D.Float(40, 40);  
float[] dist = {0.0f, 0.2f, 1.0f};  
Color[] colors =  
    {Color.RED, Color.WHITE, Color.BLUE};  
RadialGradientPaint p =  
    new RadialGradientPaint(center, radius,  
                            focus, dist, colors,  
                            CycleMethod.NO_CYCLE);
```

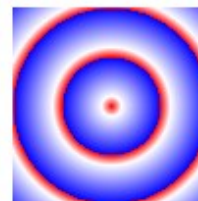
NO_CYCLE:



REFLECT:



REPEAT:

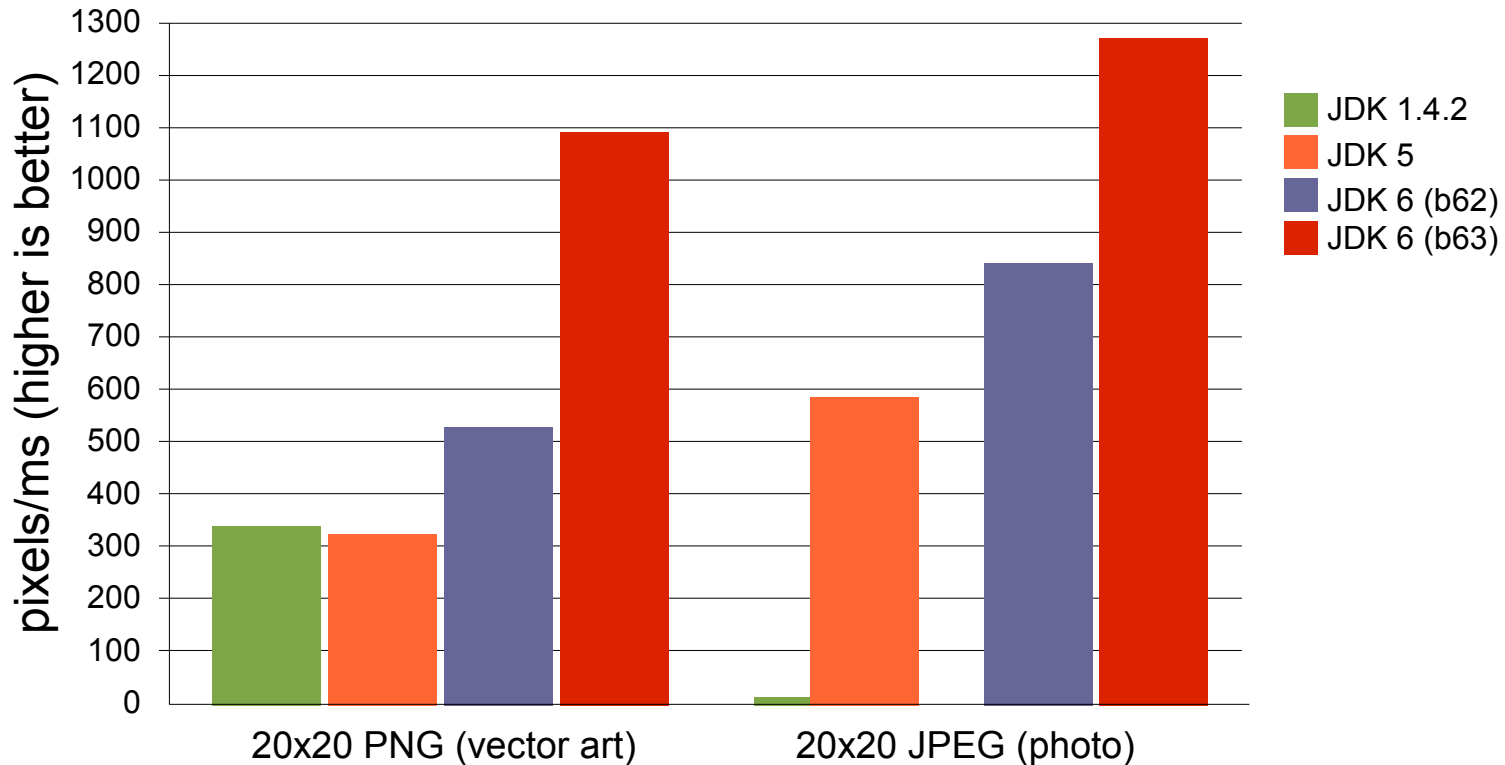


ImageIO Performance Improvements

<http://weblogs.java.net/blog/campbell/archive/2006/01>

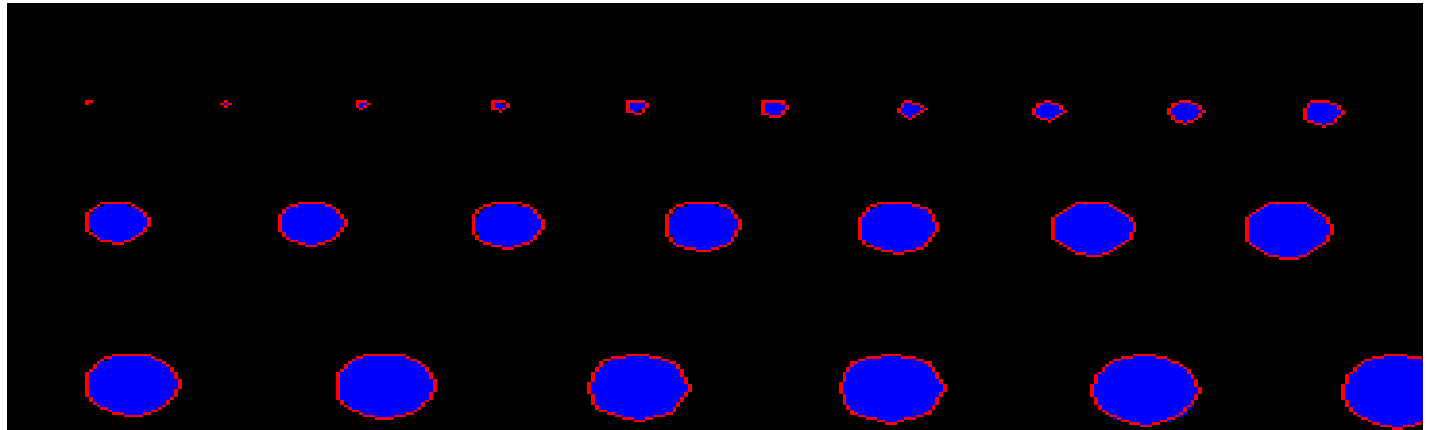
Image I/O Improvements in Mustang

(2x2.8 Ghz P4, 1 GB RAM, Windows XP, 32-bit Client VM)

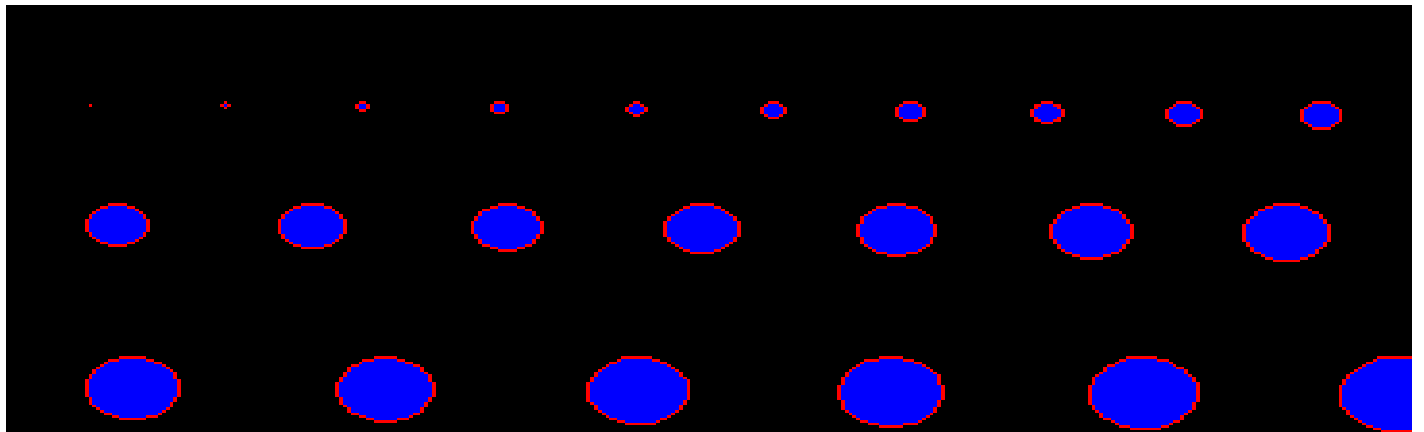


Better Curve Quality

Tiger



Mustang



Related Sessions

- Wednesday 8:30 PM: 2D and JAI BOF

Agenda

State of Desktop Java Technology

Mustang

2D

AWT

Internationalization

Deployment

Swing

AWT Enhancements in Mustang

- Major enhancements
 - Tray icon
 - Splash screen
 - Desktop class
- Other changes
 - Focus
 - Top levels
 - Text input
 - Modality enhancements and API
- Stability

Tray Icon

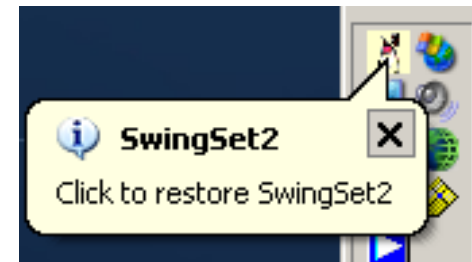
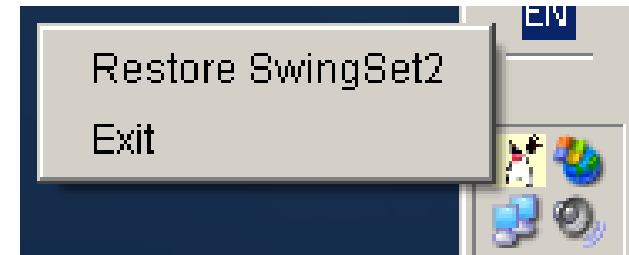
Overview

- Based on JDIC's TrayIcon
- Allows Java technology-based Apps to have an image icon (TrayIcon class) in the System Tray area
- Multiple icons per application

Tray Icon

Functionality

- ToolTip and image
- Popup menu
- Message
- Listener for various events



Tray Icon

Usage

```
// Construct a TrayIcon
TrayIcon trayIcon = new TrayIcon(image, "Tray Demo",
                                popup);

// Set the TrayIcon properties
trayIcon.addActionListener(actionListener);

// add the tray icon
SystemTray.getSystemTray().add(trayIcon);
```

Splash Screen

Overview

- Allows displaying a splash screen for the application instantly—before the JVM™ software starts!
 - GIF, PNG and JPEG images supported
 - Transparency, translucency and animation supported
 - Closed automatically on first top-level window display
- Java API to control the splash screen from the application
 - Change image, get position, close
 - Paint over the splash screen via standard `java.awt.Graphics2D`—e.g., for progress indication

Splash Screen

Usage

- Display from command line

```
java -splash:image.gif TheApp
```

- Display from MANIFEST.MF

```
Splashscreen-Image: image.gif
```

- Painting

```
SplashScreen splash =  
    SplashScreen.getSplashScreen();  
Graphics2D g = splash.createGraphics();  
// your painting code here  
splash.update();
```

Desktop Class

Overview

- New class `java.awt.Desktop` based on JDIC's Desktop class
- File processing
 - Opening/editing/printing files with applications registered in native system
- Browsing
 - Opening a URL with the default browser
- Email
 - Sending a message with the default mail client
- Depends on platform capabilities to work
 - `Desktop.isDesktopSupported()`

Desktop Class

API Overview

```
package java.awt;

public class Desktop {
    static boolean isDesktopSupported();
    static Desktop getDesktop();
    boolean isSupported(Desktop.Action action);
    void open(File file);
    void edit(File file);
    void print(File file);
    void browse(Url url);
    void mail();
    void mail(URI mailtoURI);
}
```

Desktop Class

Usage Examples

- Edit file

```
Desktop desktop = Desktop.getDesktop();  
if (desktop.isSupported(Desktop.Action.EDIT)) {  
    desktop.edit(file);  
}
```

- Browse

```
Desktop desktop = Desktop.getDesktop();  
if (desktop.isSupported(Desktop.Action.BROWSE)) {  
    desktop.browse(url);  
}
```

Related Sessions

- Wednesday 7:30 PM:
Swing/AWT/Internationalization BOF

Agenda

State of Desktop Java Technology

Mustang

2D

AWT

Internationalization

Deployment

Swing

I18N in Mustang

Overview

- **Locale Sensitive Services SPI**
 - Provides ability to plug-in locale specific data and services for `java.text` and `java.util` packages
- **ResourceBundle** enhancements
 - Flexible **ResourceBundle** loading and cache control
- **Normalizer API**
 - A new API set that conforms to the Unicode Standard (Annex #15)
- **Japanese calendar support**
 - A new implementation of the **Calendar** class that supports Japanese imperial era based calendar

Locale Sensitive Services SPI

- Provides SPIs for the `java.text` and `java.util` packages
- SPIs conform to the standard Java Extension Mechanism
- SPI Providers can
 - Offer their own locale sensitive service implementations for locales that the JRE does not support
 - Provide localized names for locales that the JRE currently does not provide

Locale Sensitive Services SPI Example

See This Demo at the AWT/2D/I18N/Swing Pavilion Pod

```
// Date Format example
DateFormat df =
    DateFormat.getDateInstance(PigLatinLocale);
Date today = new Date();
System.out.println(df.format(today));

// Locale name example
String name = Locale.US.getDisplayName(PigLatinLocale);
System.out.println(name);
```

Output:

```
"Uesdaytay, Aymay 16, 2006"
"Englishway (Unitedway Atesstay)"
```

Related Sessions

- Wednesday 7:30 PM:
Swing/AWT/Internationalization BOF

Agenda

State of Desktop Java Technology

Mustang

2D

AWT

Internationalization

Deployment

Swing

Deployment in Mustang

Ease of Deployment!

- Deployment toolkit for browsers
 - JRE detection, JRE deployment, and Java NLP application launching from the browsers
- Unified download engine for Java Web Start and Java Plug-in software
 - Improve download and caching support
 - JAR indexing, HTTP compression, HTTP cache control, offline support, flexible update policy, CD install,...
 - Java technology cache viewer

Deployment in Mustang

Even More Ease of Deployment!

- Improve user experience in Java Web Start and Java Plug-in software
 - Goodbye, scary security dialog box!
 - New download and launching experience for Java NLP applications
- Improve user experience in JDK/JRE software installer
- Improve security
 - Certificate validation—OCSP and CRL
 - Password protected keystore
 - Save password option for login

Deployment in Mustang

New Default Splash Screen for Java NLP Application

Old



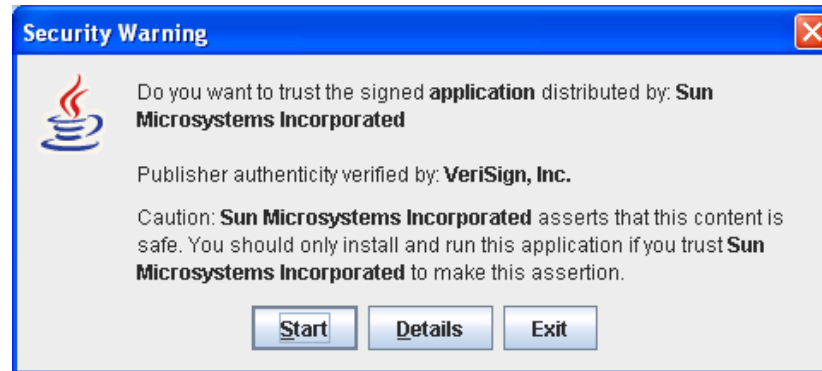
New



Deployment in Mustang

New Security Warning Dialog Box

Old



New



Deployment in Mustang

Yet More Ease of Deployment!

- Improve desktop integration
 - ICO support
 - PNG support
 - Better Windows and GNOME shell integration
- Complete Mozilla™ Firefox browser support
- Automatic proxy detection with WPAD
- Default Java technology on Linux
- Direct execution of JAR files on Linux

Related Sessions

- Tuesday 5:45 PM: Large-Scale Client Deployment Using Java™ Web Start Software
- Wednesday 9:30 PM: Meet the Deployment Team BOF
- Friday 10:45 AM: Deployment Tips and Tricks for Using Java™ Web Start and Java Plug-in

Agenda

State of Desktop Java Technology

Mustang

2D

AWT

Internationalization

Deployment

Swing

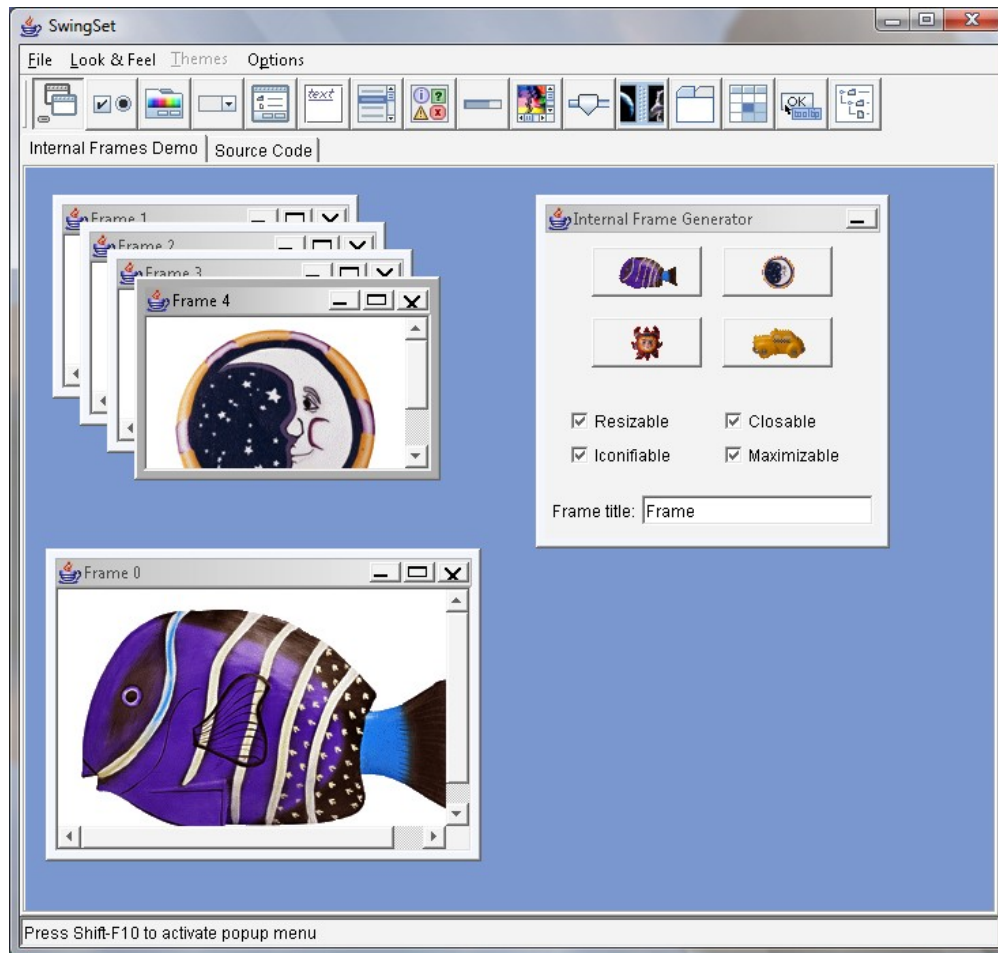
JFC/Swing API

Major Features in Mustang

- Enhanced and improved platform look and feels
- SwingWorker
- Layout enhancements
 - Including GroupLayout—basis for Matisse
- Drag and drop enhancements
- True double buffering
- Text printing
- JTable sorting and filtering
- Tabs as components

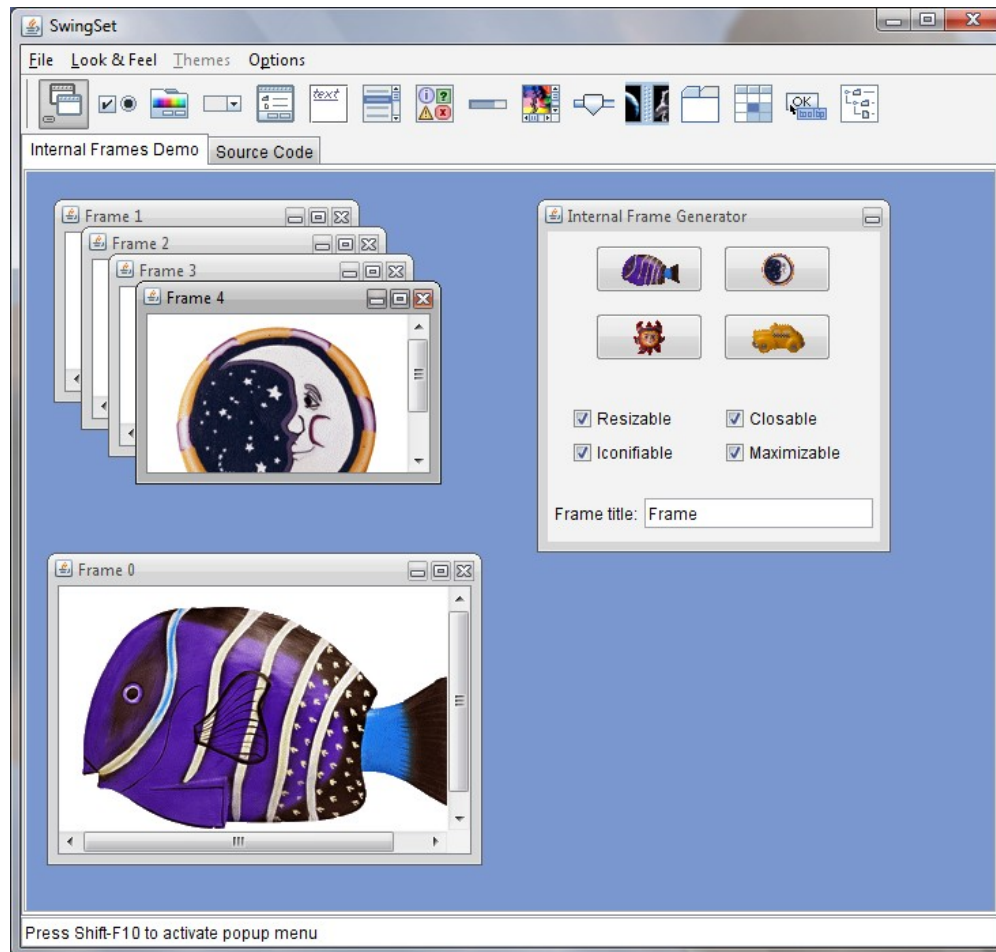
Windows Look and Feel Improvements

SwingSet on Vista with 5.0



Windows Look and Feel Improvements

SwingSet on Vista with 6



GTK Look and Feel Improvements

Crux Theme

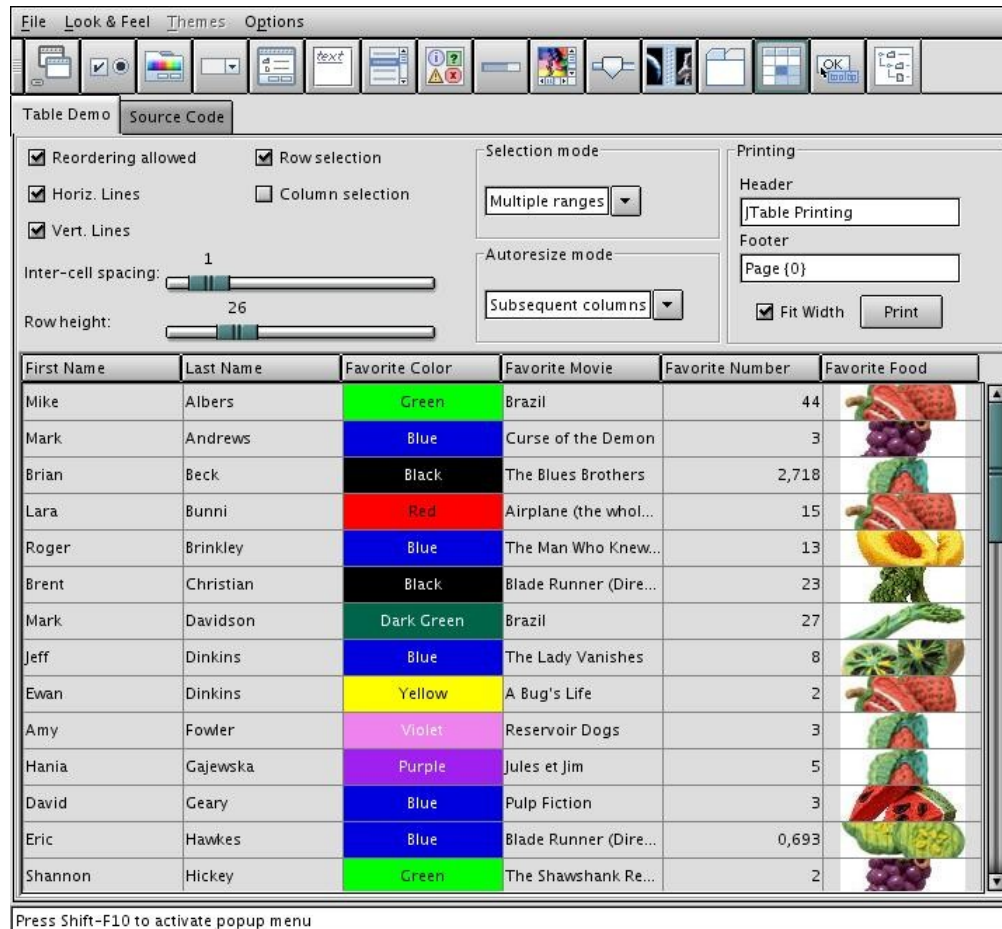


Table Demo Source Code

Reordering allowed Row selection
 Horiz. Lines Column selection
 Vert. Lines















Inter-cell spacing: 1

Row height: 26

Selection mode: Multiple ranges

Autosize mode: Subsequent columns

Printing: Header: jTable Printing, Footer: Page {0}, Fit Width, Print

First Name	Last Name	Favorite Color	Favorite Movie	Favorite Number	Favorite Food
Mike	Albers	Green	Brazil	44	
Mark	Andrews	Blue	Curse of the Demon	3	
Brian	Beck	Black	The Blues Brothers	2,718	
Lara	Bunni	Red	Airplane (the whol...	15	
Roger	Brinkley	Blue	The Man Who Knew...	13	
Brent	Christian	Black	Blade Runner (Dire...	23	
Mark	Davidson	Dark Green	Brazil	27	
Jeff	Dinkins	Blue	The Lady Vanishes	8	
Ewan	Dinkins	Yellow	A Bug's Life	2	
Amy	Fowler	Violet	Reservoir Dogs	3	
Hania	Gajewska	Purple	Jules et Jim	5	
David	Geary	Blue	Pulp Fiction	3	
Eric	Hawkes	Blue	Blade Runner (Dire...	0,693	
Shannon	Hickey	Green	The Shawshank Re...	2	

Press Shift-F10 to activate popup menu

GTK Look and Feel Improvements

Glacier Theme

Table Demo Source Code

Reordering allowed Row selection
 Horiz. Lines Column selection
 Vert. Lines
 Inter-cell spacing: 1
 Row height: 26

Selection mode: Multiple ranges
 Autosize mode: Subsequent columns
 Printing: Header: jTable Printing, Footer: Page {0}, Fit Width, Print

First Name	Last Name	Favorite Color	Favorite Movie	Favorite Number	Favorite Food
Mike	Albers	Green	Brazil	44	
Mark	Andrews	Blue	Curse of the Demon	3	
Brian	Beck	Black	The Blues Brothers	2,718	
Lara	Bunni	Red	Airplane (the whol...	15	
Roger	Brinkley	Blue	The Man Who Knew...	13	
Brent	Christian	Black	Blade Runner (Dire...	23	
Mark	Davidson	Dark Green	Brazil	27	
Jeff	Dinkins	Blue	The Lady Vanishes	8	
Ewan	Dinkins	Yellow	A Bug's Life	2	
Amy	Fowler	Violet	Reservoir Dogs	3	
Hania	Gajewska	Purple	Jules et Jim	5	
David	Geary	Blue	Pulp Fiction	3	
Eric	Hawkes	Blue	Blade Runner (Dire...	0,693	
Shannon	Hickey	Green	The Shawshank Re...	2	

Press Shift-F10 to activate popup menu

DEMO

Java File Explorer



Demo

Makes Use of Numerous Mustang Features

- File loading done in background with SwingWorker
 - User can continue to use UI while files are loaded
- Enhanced drag and drop support to determine valid drop location
 - And avoid changing selection during drag and drop
- Custom component to render tab
 - Allows user to close tab by clicking on close button in tab
 - File loading indicator rendered on tab
- Sorting via JTable's sorting API

SwingWorker

Easing Multi-threaded Apps with Swing

- Makes it easy to offload work to separate thread
- Makes use of concurrency package
- Generified
- Supports partial results
- Supports PropertyChangeListener

Loading Files Using SwingWorker

Files Loaded in Background

```
class FileLoader extends SwingWorker<List<File>,File>{
    private final File path;

    // Invoked on a background thread. Loads all files.
    protected List<File> doInBackground() throws Exception {
        return Arrays.asList(path.listFiles());
    }
}
```

Loading Files Using SwingWorker

UI Updated Once Files Loaded

```
class FileLoader extends SwingWorker<List<File>,File>{
    private final File path;

    // Invoked on the EDT when done (or canceled)
    protected void done() {
        try {
            // Get the list of files and update model
            List<File> files = get();
            tableModel.setFiles(files);
        } catch (Exception e) {}
    }
}
```

Drag and Drop Enhancements

- Drag no longer requires selection to be made first!
- **setDropMode ()** to specify how drop location is tracked
 - Enum: `USE_SELECTION` (old behavior), `ON`, `INSERT`, `ON_OR_INSERT...`
 - Implemented on `JTree`, `JList`, `JTable` and `JTextComponent`
- API to fetch drop location
 - As a point
 - As appropriate for component

Specifying The Drop Mode

```
// Install TransferHandler
table.setTransferHandler(new TreeTransferHandler());

// Configure how the table determines drop locations
// and provides drag over feedback.
table.setDropMode(DropMode.ON);
```


Determining Legal Drop Location

```
class TreeTransferHandler extends TransferHandler {
    // Called repeatedly during drag and drop. Passed in
    // object is queryable for dnd information.
    public boolean canImport(TransferSupport ts) {
        if (ts.isDrop() && ts.isDataFlavorSupported(FILE_F)) {
            JTree.DropLocation dropLocation =
                (JTree.DropLocation)ts.getDropLocation();
            // Get the path the mouse is over
            TreePath pathOver = dropLocation.getPath();
            if (validPath(pathOver)) {
                // If the path the mouse is over is valid for the
                // specified files, return true.
                return true;
            }
        }
        return false;
    }
}
```

Processing Drop

```
class TreeTransferHandler extends TransferHandler {
    // Called to accept the drop (or paste).
    public boolean importData(TransferSupport ts) {
        if (canImport(ts)) {
            JTree.DropLocation dropLocation =
                (JTree.DropLocation)ts.getDropLocation();
            // Get the path the user dropped over
            TreePath pathOver = dropLocation.getPath();
            // Extract the files from the Transferable
            List<File> files = getFiles(ts.getTransferable());
            processDrop(files, pathOver);
            return true;
        }
        return false;
    }
}
```

JTable Sorting and Filtering

- Add sorting to your JTable with one method call
 - `setAutoCreateRowSorter(true)`
- Ability to specify your own Comparators
- Also supports secondary, tertiary sort columns
- Can specify a filter to limit what is shown
 - Regular expression, number and date implementations provided

JTable Sorting

```
TableRowSorter rowSorter = new TableRowSorter(model);

// Using a different Comparator
rowSorter.setComparator(0, new FileNameComparator());

// Sort the contents on the first column
rowSorter.setSortKeys(Arrays.asList(
    new SortKey(0, SortOrder.ASCENDING)));

// Install the RowSorter.
// If you don't need to customize the RowSorter, use
// JTable.setAutoCreateRowSorter(true);
table.setRowSorter(rowSorter);
```

Related Sessions

- Wednesday 11 AM: Swing Threading 101
- Wednesday 7:30 PM:
Swing/AWT/Internationalization BOF
- Thursday 8:30PM: Drag and Drop BOF

Future Directions

- Swing/2D across devices (JSR 209)
- Ease of development
 - Tools support (NetBeans™ GUI Builder, Java Web Start plugin)
 - JSR 295 (Beans Binding)
 - JSR 296 (Swing Application Framework)
- Modular deployment (JSR 277)
- Desktop integration
 - Native dialogs, browser embedding
- User experience
 - Animation framework
 - New Applet startup animation

Desktop Java Technology in Mustang

- Native Desktop Integration
- Better User Experience
- Performance
- Try it out today!

<http://mustang.dev.java.net>

Related Sessions

- Tuesday 5:45 PM: Large-Scale Client Deployment Using Java™ Web Start Software
- Wednesday 11 AM: Swing Threading 101
- Wednesday 12:15P: Expanded Capabilities for Mobile Development with JSR-209
- Wednesday 7:30-10:30 PM: Swing/AWT/i18n, 2D/JAI, Deployment BOFs
- Thursday 2:45P: “The Incredible Shrinking Application”: Making Desktop Applications Mobile with CDC
- Thursday 7:30 PM: Desktop Futures Panel
- Thursday 8:30PM: Drag and Drop BOF
- Friday 10:45 AM: Deployment Tips and Tricks for Using Java™ Web Start and Java Plug-in

...as well as the rest of the desktop sessions!

Q&A



the
POWER
of
JAVA™



JavaOne
Part of the Network and Business Solutions

Desktop Java™ Technology: Deep Dive

Thorsten Laux
Chet Haase
Oleg Sukholdosky
Scott Violet

Sun Microsystems

TS-1593