

Cross-Browser Java

Making Java Applets Work On All Web Browsers

Gunther Birznieks

<http://www.extropia.com/Scripts/>

<http://www.gunther.web66.com/crossjava/>

Cross-Browser Java

- Introduction to the issues
- Solutions
 - Software Engineering
 - General Guidelines
 - Specific Browser Bug Fixes
- Questions

Cross-Browser Java

- The Problems
 - Different browser VMs
 - Different platform VMs
 - Technical Problems versus Bugs
- Java is not the only problem
 - JavaScript
 - HTML Layout
 - DHTML

Client-Side Java Cases

- Famous “Failure”
 - Corel’s Java Office Suite
- But there are successes
 - Network Flight Recorder
 - <http://www.nfr.net>
 - Citrix Java client
 - ApplixAnywhere

Browsers

- Internet Explorer 3 & 4
- Netscape 3 & 4
- Appletviewer/HotJava
- <http://www.browserwatch.com/>

Browsers - Case Study

- Netscape
 - NSPR 2.0
 - Netscape Portable Runtime
 - Threading support for Win32, Mac, 20+ Unix
 - Part of Mozilla CVS Tree
 - <http://www.mozilla.org>

Browsers - JDK Versions

- JDK 1.0.2
 - Old AWT, But lowest common denominator
 - Netscape 3, IE 3
 - Netscape 4 w/o patch, < 4.05

Browsers - JDK Versions

- JDK 1.1
 - Adds inner classes (useful) -- Can use with JDK 1.0.2 if you like walking in dark alleys at night.
 - JDBC Standard -- can use in JDK 1.0.2 though!
 - Bean/New Event Model support
 - Netscape 4 w/patch, 4.05, IE 4, Java Plug-In
- JDK 1.2
 - Swing -- lightweight components
 - And more...

Browsers - Plug-Ins (JDK 1.1)

- Java JDK Plug-In
- MRJ PlugIn
 - Problems
 - Not tightly coupled to the browser
 - Lack SSL Support
 - Lack Authentication support
 - Cookies not accessible from plug-in

Browsers - Platform?

- What Platform Are We On?
 - System.getProperty()
 - java.vendor
 - os.name
 - Most other properties give Security Exceptions under some browsers

Browsers - Platform?

- Sample Code (Check For Z-Order)

```
String osName =  
    System.getProperty( "os.name" )  
        .toLowerCase( ) ;  
String browser =  
    System.getProperty( "java.vendor" )  
        .toLowerCase( ) ;
```

Browsers - Platform?

- Sample Code Continued

```
if ( (osName.indexOf("window") != -1 &&
      browser.indexOf("netscape") != -1) ||
      (osName.indexOf("mac") != -1 &&
      browser.indexOf("microsoft") != -1)) {
    add(label_1); add(label_2); add(label_3);
} else {
    add(label_3); add(label_2); add(label_1);
}
```

Cross-Browser Java

- Solutions
 - Software Engineering Principles
 - Reduce Java Usage To A Minimum
 - But can't always do that so...
 - General Cross-Platform Guidelines
 - Peer issues
 - Threading
 - Cross-Platform Bug Workarounds

Software Engineering Perspective

- Tips on minimizing Java use
 - Less Java = Less debugging
- HTML Tricks

AntiPattern

- Software Development Mini-AntiPattern
 - Continuous Obsolescence
 - Technology is changing so rapidly that finding compatible releases of products that interoperate is difficult. Sound familiar?
 - AntiPattern Solution:
 - Minimize use of risky technology.
- Anti-Patterns book
 - Brown, Malveau, McCormick, Mowbray

Minimize Java Use

- Pattern your application around CGI/HTML Forms
- Use Application Servers
 - CORBA
 - JavaCGIBridge (~6k overhead)
 - Marshals data to and from CGI scripts
 - JDBC helps a bit
 - But is usually a 2-tier solution or a pseudo-3-tier solution with stored procedures

Minimize Java Use

- Less Java Code, Less Problems
 - Applications Server Helps
 - Less memory problems
 - JDBC is a bit of a bandwidth hog, so are CORBA ORBs
 - ~200-400k
- Use showDocument() to call up reports for printing
 - Fancy Graphs and Reports can be printed as HTML output from CGI
 - Beware of platform bug...discussed later

Use HTML Tricks

- JavaScript/CGI can sometimes do the trick
- Misconception: Graphics must be done in Java
 - Can use other libraries
 - CGI Scripts can create gif/jpegs using libraries such as GD lib
 - Simple Bar charts can be simulated using IMG SRC HEIGHT, WIDTH tags to scale.

General Cross Browser Guidelines

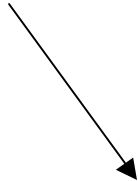
- AWT
- Threading
- Finalization/Garbage Collection
- JDK Compiling
- Archive Files
- JDBC Usage

Guidelines - AWT

- Fonts
 - Use standards
 - Times, Helvetica, Courier
 - JDK 1.1 uses Serif, Sans Serif, Monospaced
 - Font Sizes differ on platforms
 - UNIX tends to be larger than Win32
 - Mac tends to be smaller than Win32
 - Case Study -- Using Netscape 4.0

Font Case Study

```
Label label = new Label("This is a message.");  
add("South", label);  
  
Font font = label.getFont();  
  
FontMetrics metrics =  
label.getFontMetrics(font);
```



Test Font
metrics with
this Label

Font Case Study

```
String s1 = "abcdefghijklmnopqrstuvwxy" +  
            "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
String s2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
String s3 = "abcdefghijklmnopqrstuvwxy";  
  
System.out.println(metrics.stringWidth  
                    (label.getText()));  
  
int size = label.getText().length();
```

Font Case Study

```
int w1 = metrics.stringWidth(s1);  
int w2 = metrics.stringWidth(s2);  
int w3 = metrics.stringWidth(s3);
```

```
int factor1 = w1 / (float)s1.length();  
int factor2 = w2 / (float)s2.length();  
int factor3 = w3 / (float)s3.length();
```

```
System.out.println(factor1);  
System.out.println(factor2);  
System.out.println(factor3);
```

Font Case Study Results

- Windows
 - Base Width = 87
 - Scaling Factors = 111, 130, 93
- Mac
 - Base Width = 82
 - Scaling Factors = 99, 106, 91
- UNIX (Solaris)
 - Base Width = 97
 - Scaling Factors = 118, 126, 109

Font Case Study Results

- UNIX (IRIX)
 - Base Width = 108
 - Scaling Factors = 108, 108, 108 (Fixed!)
- UNIX (LINUX)
 - Base Width = 110
 - Scaling Factors = 135, 145, 124

Font Case Study

- Windows Height = 15
- Mac Height = 13
- UNIX (Solaris) Height = 14
- UNIX (IRIX) Height = 13
- UNIX (LINUX) Height = 14

Guidelines - AWT

- Colors
 - Many components ignore color changes especially on JDK 1.0.2
 - Best to use defaults
- Keystrokes
 - Some keys are special on some platforms
 - Mac's Forward Delete Key Bug
- Focus
 - Basically only consistent with Text components

Guidelines - AWT

- Peer Classes
 - Limitations
 - TextArea on Windows has a limit (~30k)
 - Solaris is more forgiving (> 200k)
 - Focus & other events
- Using JDK 1.1 and 1.0.2 at the same time
 - JavaSoft's "Writing Compatible Programs"
 - <http://java.sun.com/products/jdk/1.1/compatible/index.html>

Guidelines - AWT

- JPEGs
 - Older browsers sometimes do not recognize certain JPEG formats for loading
 - Strip out application data block (APPn)
 - Turn off save thumbnails preference in Photoshop
 - If the image is previously cached it will work (since the connection is fast again).
 - Bad to rely on this.

Guidelines - Threading

- Use `Thread.sleep()`
- `Thread.yield()`
 - When not wasting CPU
- Don't use too many threads
 - Try Thread pooling

Guidelines - Garbage Collection

- Don't rely on Finalization() to clean up
 - JDK is not guaranteed to garbage collect
 - Implementation Specific
 - Especially if you have a lot of RAM
- Don't necessarily rely on stop() or destroy() for critical resources
 - Browser could crash

Guidelines - JDK Compiling

- Use the lowest common denominator JDK
 - javac version 1.0.2 for backwards compatibility
- Feel like taking a risk?
 - javac version 1.1.x for inner classes

Guidelines - Using Archive Files

- IE 3 Supports CAB
- Netscape 3 supports uncompressed zip
- Netscape 4 and IE 4 supports compressed jar
- Solution:
 - Use zip before jar in Archive tag
 - Or use JavaScript on the next slides
 - `<PARAM NAME="CABBASE">` does not hurt for supporting Microsoft.

Guidelines - Using Archive Files

- JavaScript Code To Output Different Applet Tag

```
<SCRIPT LANGUAGE="JavaScript">
<!--
if (navigator.appName.indexOf("Netscape") >= 0
    && parseFloat(navigator.appVersion) >= 4) {
document.writeln(
    '<APPLET CODE="ArchiveTest3.class" CODEBASE="./Java"
ARCHIVE="archivetest3.jar" HEIGHT=50 WIDTH=300>');
```

Guidelines - Using Archive Files

- JavaScript Code To Output Different Applet Tag

```
} else {  
  document.writeln(  
    '<APPLET CODE="ArchiveTest3.class" CODEBASE="./Java"  
    ARCHIVE="archivetest3.zip" HEIGHT=50 WIDTH=300>');  
}  
  
//-->  
  
</SCRIPT>
```

Guidelines - JDBC

- JDBC is an official part of JDK 1.1
 - java.sql.*
 - JDK 1.0.2 Drivers exist
 - java.sql.* hierarchy renamed so applets can download the classes
 - Download is longer for the java.sql.* classes to transfer
 - Sybase example

Sybase JDBC - JDK 1.1

Includes:

```
import java.io.*;  
import java.sql.*;  
import com.sybase.utils.Debug;  
import java.util.*;
```

Driver:

```
Class.forName("com.sybase.jdbc.SybDriver");
```

Sybase JDBC - JDK 1.0.2

Includes:

```
import jdbc.sql.*;  
import jdbc.math.*;  
import jdbc.sybase.utils.Debug;  
import jdbc.sybase.jdbc.*;  
import java.util.*;
```

Driver:

```
Class.forName("jdbc.sybase.jdbc.SybDriver");
```

Bug Workarounds

- AWT Bugs
 - Modal Dialog
 - Forward Delete Key
 - Drop Down Choice Bug
 - Z-Ordering
 - URL Target showDocument()

Bug Workarounds

- Networking, Threading, Classes
 - Thread.join()
 - ThreadGroup Security
 - URLConnection/HTTP 1.1
 - Static Initializer (JDBC)

Modal Dialog Bug

- Thread of execution is supposed to stop if a dialog is modal
- Not Always True
 - JDK 1.0.2 Browsers
 - Modal Dialogs thrown from an applet (instead of a real frame)

Modal Dialog Workaround

- Use a callback!
 - If you need to do something immediately after getting the data
 - Messy design -- Dialog has to KNOW about the caller
 - Can mediate this by using an interface for the action method instead
- Using Threads
 - Launch dialog within a second thread and then `Join()` that thread until it ends.

Mac Forward Delete Key Bug

- UNIX/Win32 use the forward delete key to delete characters
- Mac does not.
 - Worse, it inserts an invisible character! Yuck!

Mac Forward Delete Key Workaround

- Check for the character and ignore...or
- Simulate the forward delete key
- Code on the next page
 - Algorithm:
 - Check for `keypress == 127`
 - Check for Mac (expensive operation deferred)
 - If no selection, just delete 1 char
 - If selection, delete whole selection
 - `get/setCaretPosition()` works better but JDK 1.1 only

```

if (event.id == event.KEY_PRESS) {
    if (event.key == 127) {
        String osName =
            System.getProperty("os.name").toLowerCase();
        if (event.target instanceof TextComponent
            && osName.indexOf("mac") != -1) {
            TextComponent tf = (TextComponent)event.target;
            int start = tf.getSelectionStart();
            int end = tf.getSelectionEnd();
            String text = tf.getText();
            if (end - start == 0) {
                text = text.substring(0,start) +
                    text.substring(end + 1,text.length());
            } else {
                text = text.substring(0,start) +
                    text.substring(end,text.length());
            }
            tf.setText(text);
        }
    }
}

```

Drop Down Choice Bug

- Works when number of choices is small
- When number of choices expand, UNIX doesn't expand the choice box with it and does not provide a scroll bar.
 - Win32 adds scroll bar
 - Mac adds little arrows

Drop Down Choice Workaround

- Change your coding style if you have a lot of items to fill the choice with
- Use a button + a dialog with a listbox to scroll choices
 - REMEMBER the Modal Dialog problem!
 - Use an interface to make the choice listbox more generic

Z-Order Bug

- Items are not always drawn in the order they were added
 - Platform dependant
- Worse
 - Applet Panel and Frame exhibit different z-order behavior
- Primary effect: third party libraries with owner drawn combo boxes
 - As opposed to choice() peers

Z-Order Workaround

- Use code to compensate and add the components in an order conducive to the way you want them drawn.
- Frame Z-Order Workaround Code

```
if ( (osName.indexOf("window") != -1 &&
      browser.indexOf("netscape") != -1) ||
      (osName.indexOf("mac") != -1 &&
      browser.indexOf("microsoft") != -1)) {
    add(l1); add(l2); add(l3);
} else {
    add(l3); add(l2); add(l1);
}
```

Z-Order Workaround

- Applet Panel Z-Order Workaround Code

```
if (osName.indexOf("window") != -1 &&
    browser.indexOf("microsoft") != -1) {
    add(l3); add(l2); add(l1);
} else {
    add(l1); add(l2); add(l3);
}
```

URL Target Bug

- Common Web Design Pattern
 - Use applet for data entry
 - CGI + HTML output for reporting
- `showDocument()` to a target does not work on Mac VMs

URL Target Workaround

- `showDocument()` does work the second time around if the window is already open.
- `_blank` has no hope because a new browser is opened all the time.
- `_namedTarget` can be called twice.
 - First time, a blank browser is opened
 - Second time, the URL gets filled in
- Can simulate `_blank` by creating random name targets.

URL Target Workaround

```
getAppletContext().showDocument(u, target);  
String osName =  
    System.getProperty("os.name")  
        .toLowerCase();  
String browser =  
    System.getProperty("java.vendor")  
        .toLowerCase();  
if ((osName.indexOf("mac") != -1 &&  
    browser.indexOf("netscape") != -1)) {  
    getAppletContext().showDocument(u, target);  
}
```

Does it again for Mac.

Thread.join() Bug

- Broken in all Netscape browsers
- Used for timeouts
- Very common in communications programming
 - Create Thread that does blocking socket call
 - join() the thread for a given period of time to timeout if communications are down.
- Raw Join code on next page.

```
long base = System.currentTimeMillis();
long now = 0;
if (millis == 0) {
    while (isAlive()) {
        wait(0);
    }
} else {
    while (isAlive()) {
        long delay = millis - now;
        if (delay <= 0) {
            break;
        }
        wait(delay);
        now = System.currentTimeMillis() - base;
    }
}
```

Thread.join() Workaround

- wait() for a delay period.
- if notifyAll() is called, wait is stopped and checks if thread is still alive
 - Requires a thread variable
 - Can't use isAlive() -- race condition

```
long base = System.currentTimeMillis();
long delay = 0;
synchronized(this) {
    try {
        while(!_isThreadReallyAlive) {
            delay = timeOut -
                (System.currentTimeMillis()
                 - base);
            if (delay <= 0) break;
            wait (delay);
        }
    } catch (InterruptedException e) {}
} // End synchronized block
// TIMEOUT ENDS HERE
```

Thread.join() Workaround

```
synchronized(this) {  
    _isThreadReallyAlive = false;  
    notifyAll();  
}
```



Add to end of run()

ThreadGroup Bug

- Security Exception Thrown if suspend, resume, interrupt, destroy, setPriority, setName, setDaemon on Netscape 3 for non-applet ThreadGroups
- Problem for threads created inside AWT Event Thread
 - eg Button Click

ThreadGroup Workaround

- Get a handle to the applet's threadgroup
- Use it in the Thread creation. ThreadGroup is a parameter to the constructor

 ↗ New instance variable

```
private ThreadGroup _appletThreadGroup;
```

```
_appletThreadGroup =  
    Thread.currentThread().getThreadGroup();
```

 ↘ Add to init() method

URLConnection/HTTP 1.1 Workaround

- Turn off HTTP 1.1 in the browser
- Use nph- scripts without buffering
- Filter out HTTP 1.1 in your web server
 - Apache Solution
 - HTTP.CONF
 - BrowserMatch "MSIE 4\.0;" nokeepalive force-response-1.0 downgrade-1.0

JDBC Bug Workaround

- Internet Explorer Bug (also AIX JDK 1.1)
 - Download singleton class with static methods
 - Then, class with static initializer that calls a static method, it creates a temp new version of class.
- Change registration code:

```
DriverManager.registerDriver(  
(Driver)Class.forName  
("jdbc.sybase.jdbc.SybDriver").newInstance());
```

instead of

```
Class.forName("com.sybase.jdbc.SybDriver");
```

Static Initializer (JDBC) Bug

- Internet Explorer Problem (3 and 4 even SP1)
- If a static initializer calls a static method in another class, the other class is duplicated internally to the JVM and the initializer is actually not calling the original class so the registration gets discarded.
- This is how JDBC works though...
- An example...

Driver Code

```
public class MyStaticDriver {
    private boolean _registered = false;

    static {
        MyStaticDriverManager.register(
            new MyStaticDriver());
    }

    public MyStaticDriver () {
        _registered = true;
    }
}
```

DriverManager Code

```
public class MyStaticDriverManager {
    private static int _registerCount;

    public static void register(Object o) {
        _registerCount++;
    }

    public static int getRegisterCount() {
        return _registerCount;
    }
}
```

Static Initializer (JDBC) Workaround

- Do not simply load the class.
 - `Class.forName("MyDriver.class");`
- Instead instantiate the class and manually do what the static initializer was doing

```
MyStaticDriverManager.register(  
    (MyStaticDriver)Class.forName(  
        "MyStaticDriver").newInstance());
```

More Information...

- [Http://gunther.web66.com/crossjava/](http://gunther.web66.com/crossjava/)
 - Updated talk plus details
 - Example code for the talk
- <http://www.extropia.com/Scripts/>
 - JavaCGIBridge class examples and source
- Acknowledgements
 - Joseph Ryan, Erik Ferlanti, Anthony Masiello, Mark McDonald, Peter Chines, Selena Sol