Welcome

To Advance through Presentation
Use Page Up and Page Down Keys

99 | Worldwide
Developers
Conference

99 Worldwide Developers Conference

# Inside MRJ

Peter A. Steinauer
Jens Alfke
John Burkey

Java Platform Engineering

# Overview

- Building Java into your Mac OS Application
- Building the Mac OS into your Java Application
- MRJ Debugging
- Thinking toward the future

99 Worldwide Developers Conference

# Building Java into Your Mac OS App

# What Is JManager?

- Java VM Invocation
- UI Embedding and Event Delegation
- Access to Java Native Interface (JNI)

# JManager 2.1

- JNI Support / JNI Object Access
    - JMJNIToAWTContext
- AWT Support
    - JMMenuSelectWithModifiers and JMFrameClickWithEventRecord
    - JMDrawFrameInPort

# JMAppletPage APIs

```
typedef struct OpaqueJMAppletPageRef*           JMAppletPageRef;


OSStatus JMNewAppletPage(
        JMAppletPageRef *                       page,
        JMSessionRef                            session );

OSStatus JMDisposeAppletPage(
        JMAppletPageRef                         page );

OSStatus JMNewAWTContextInPage (
        JMAWTContextRef *                       context,
        JMSessionRef                            session,
        JMAppletPageRef                         page,
        const JMAWTContextCallbacks *           callbacks,
        JMClientData                            data);
```

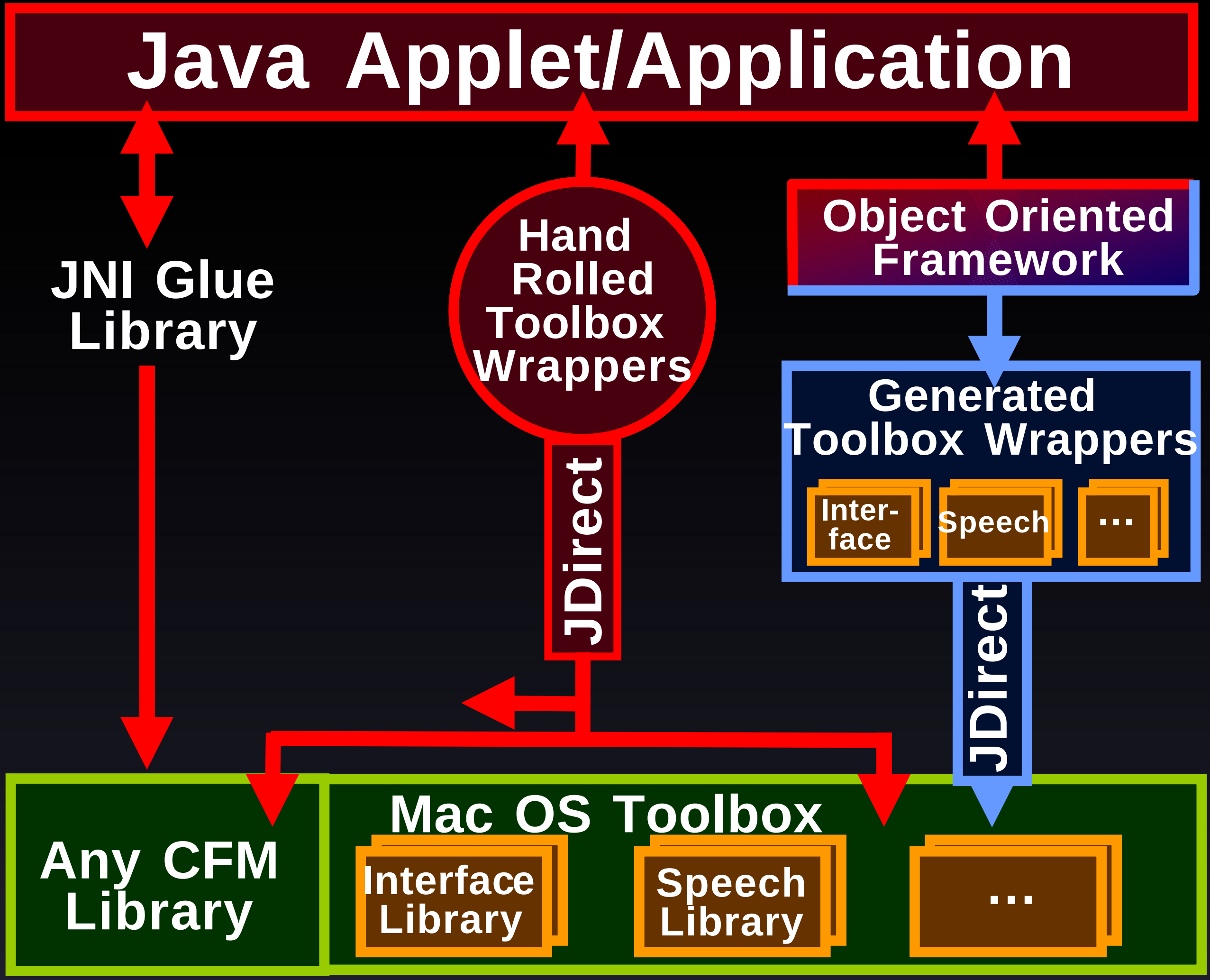
99 Worldwide Developers Conference

# Building the Mac OS into Your Java App

# JDirect: What It Is

- Direct access to existing native code
- No need for C Glue
- Standard Java Syntax
- Automatic data marshaling
- Not part of Java standard
- No straightforward manipulation of Java objects from Native code

# JDirect vs. JNI



99 Worldwide Developers Conference

# Putting It to Work

Jens Alfke

AWT Lead Engineer

# New Java-Savvy APIs

- All-JDirect implementation (no C++)
- Drag & Drop
- Extended AWT Controls

# Drag & Drop

- Java 1.1-compatible
- *Not* Sun's API (which is Java2-only)
- Better integration with Drag Manager

# API: Receiving Drags

```
package com.apple.mrj.dnd;

public interface DragListener extends MouseListener {
        boolean dragEntered( DragEvent e );
        boolean dragMoved( DragEvent e );
        boolean dragExited( DragEvent e );
        boolean dragDropped( DragEvent e );
}
```

# Receiving Drags

```
public boolean dragDropped( DragEvent e ) {
    Drag d = e.getDrag();
    Transferable item = d.getItem(0);
    if( item.isDataFlavorSupported(kTypeTEXT) {
        String s = new String(
            Transfer.getTransferDataBytes(item,kTypeTEXT) );
        insertIntoMyContent(s);
        return true;
    } else
        return false;
}
```

# API: Initiating Drags

```
package com.apple.mrj.dnd;

public interface DragInitiatorListener
                                        extends MouseListener {
        void dragGesture( DragInitiatorEvent e );
        void dragCompleted( DragInitiatorEvent e );
        void dragFailed( DragInitiatorEvent e );
}
```

# Initiating Drags

```
public void dragGesture( DragInitiatorEvent e ) {
    String s = getMyContent();
    Rectangle r = getMyContentBounds();

    Transfer item = new Transfer();
    item.addFlavor(kTypeTEXT, s.getBytes());
    e.getDrag().addItem(item);
    e.setDragRect(r.x, r.y, r.width, r.height);
}
```

# Compatibility

- Implemented in MRJ 2.2 EA1
- ….but it's easy to support standard Java2 drag & drop API as well
- Usage doesn't make your app "MRJ-only"

# Extended AWT Controls

- Extra set of AWT Components
- Exposes more Appearance Manager functionality with AWT-like API
- Appearance—and Kaleidoscope-savvy
- Simpler/faster/smaller than Swing

# Here's What You Get:

Controls and Indicators:

BusyIndicator
DateSelector
DefaultButton
Discloser
GroupBox
IconButton
MixedCheckbox
Nudger
ProgressBar
Slider
TabPanel
Well

Static UI Elements:

FocusRing
IconView
Placard
Separator
StaticText

Windows:

Alert

# Sample API: Slider

```
package com.apple.awt;

public class Slider extends java.awt.Scrollbar {
        public Slider( int orientation,
                       boolean showTickMarks,
                       int indicatorDirection );
        public boolean isShowingTicks( );
        public int getIndicatorDirection( );
}
```

# Compatibility

- An add-on to MRJ 2.2 (SDK coming soon)
    - Experimental/unsupported!
- *Not* a standard Java platform API

99 Worldwide Developers Conference

# Demo

Drag & Drop

Extended AWT Controls

99 Worldwide Developers Conference

# JDirect: Down the Rabbit Hole

# Calling into the Generated Interfaces

```
import com.apple.mrj.macos.toolbox.Toolbox;
import com.apple.mrj.macos.generated.SoundFunctions;

public void playABeep( ) {
        SoundFunctions.SysBeep(1);
}
```

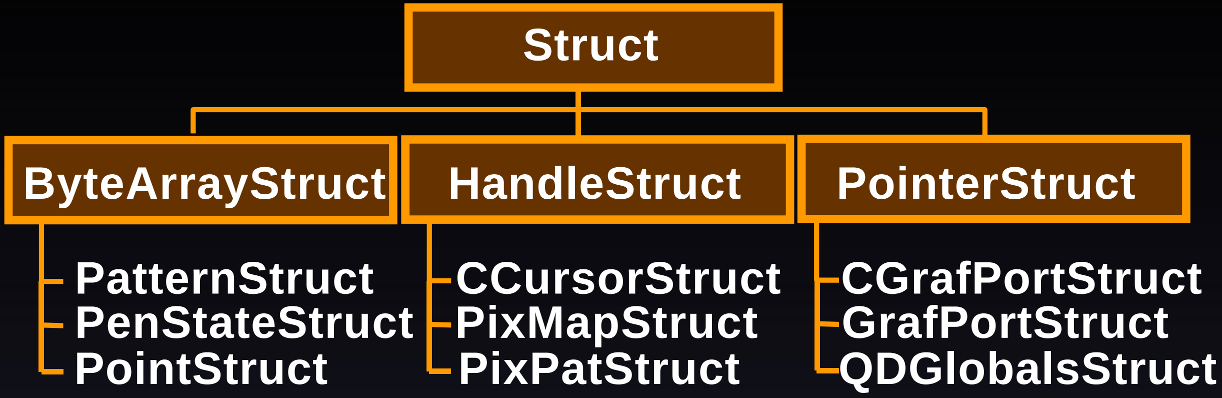
But that’s not quite correct. . .

# Toolbox Synchronization

```
import com.apple.mrj.macos.toolbox.Toolbox;
import com.apple.mrj.macos.generated.SoundFunctions;

public void playABeep( ) {
        synchronized( Toolbox.LOCK ) {
                SoundFunctions.SysBeep(1);
        }
}
```

# JDirect and Data Types



Struct

ByteArrayStruct
- PatternStruct
- PenStateStruct
- PointStruct

HandleStruct
- CCursorStruct
- PixMapStruct
- PixPatStruct

PointerStruct
- CGrafPortStruct
- GrafPortStruct
- QDGlobalsStruct

# Parameters in JDirect

```
public static short FSpGetFInfo(
         FSSpecStruct        spec,
         FInfoStruct         fndrInfo)
{
          return FSpGetFInfo(
                  spec.getByteArray(),
                  fndrInfo.getByteArray());
}
public native static short FSpGetFInfo(
                                     byte[] spec,
                                     byte[] fndrInfo );
```

# Defining Callbacks

```
public class MyAEEventHandler
        implements AEEventHandlerInterface
{
        public short AEEventHandler(
                int theAppleEvent,
                int reply,
                int handlerRefcon )
        {
                ....
        }
}
```

# Defining Callbacks

```
public final class AEEventHandlerClosureUPP
        extends MethodClosureUPP
{
        public AEEventHandlerClosureUPP(
                AEEventHandlerInterface context )
        {
                super(context,
                        "AEEventHandler",
                          "(III)S",
                        0x00000FE0);
        }
}
```

# Registering Callbacks

```
MyAEEventHandler           myHandler;
AEEventHandlerClosureUPP   handlerUPP;

myHandler = new MyAEventHandler();
handlerUPP = new AEEventHandlerClosureUPP(
                    myHandler );

err = AppleEventFunctions.AEInstallEventHandler(
          kTestClass,
          kTestID,
          myAEHandlerUPP,
          0,
          false );
```

99 Worldwide Developers Conference

# MRJ Debugging

# MRJ Dcmd

- Java Stack Crawls
- Java Object Inspection
- Deadlock Detection
- Misc. Utilities

# MRJ Debug Build

- Method Tracing
- Allocation and GC analysis
- Should be run without JITC
- Impacts Performance

# Thinking Toward the Future

99 Worldwide Developers Conference

# What You Should Plan For

- Deprecation of APIs
  - (JRI, JManager1, JDirect1, etc.)
- Mac OS X / Carbon

# What We Have Planned

- What we have planned
  - HTTPs / SSL Support for Applets (applications coming too)
  - Performance improvements
    - Applet caching
    - Faster GC
    - Faster class loading
    - QuickDraw acceleration
  - Java2 under investigation

# We Like Standards Too

- MRJ will always be 100% compatible
- MRJ 2.1 already supports Swing
- Java2 implementation in development
    - AWT drag and drop API
    - Java2D graphics API
    - …both better than in JDK

99 Worldwide Developers Conference

# Java 2D

John Burkey
AWT Graphics Engineer

# Java2D—High Quality Graphics in Java

- Antialiased Graphics
- General Paths (Fills and Strokes)
- Compositing/Transfer Modes
- 3x2 Matrix (Rotate, Scale, Translate)
- Arbitrary Fills (Gradient, Textures, Developer supplied)
- Advanced Text Handling

# Why We All Need It

- Antialiasing
- Rotated Text (Text as first class graphics objects)
- Translucency, Compositing, Transfer Modes
- Arbitrary Fills
- Line widths

99 Worldwide Developers Conference

# Demo

John Burkey
AWT Graphics Engineer

# Feedback Loop

- MRJ Developer Page
  http://developer.apple.com/java/
- MRJ Discussion List
  http://www.lists.apple.com/mrj.html
- MRJ Feedback
  mrj_feedback@apple.com
- Bug Reporting
  http://developer.apple.com/bugreporter/
- MRJ Home Page
  http://www.apple.com/java/

# Other Java Forums

| | |
|---|---|
| **Java Overview** | Hall 2<br>**Tues., 2:30pm** |
| **Quicktime for Java** | Hall A2<br>**Tues., 4:00pm** |
| **Java on Mac OS X** | Hall 2<br>**Thurs., 1:00pm** |
| **AppleScript for Java** | Hall B<br>**Thurs., 2:30pm** |

99 Worldwide Developers Conference

# Q&A

Think different.™

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys

99 | Worldwide
Developers
Conference