# Troubleshooting Java™ Platform, Micro Edition
## Tips from the Pros

**Rodney Aiglstorfer**

CTO and Co-Founder
mFoundry Inc.
www.mfoundry.com

TS-5927

# Goal of This Talk

## Troubleshooting Java ME™

In this session you will get an opportunity to learn the many “secrets” to developing real world Java ME applications

# Agenda

Java ME Realities

What Devices to Support

Test Strategies and Certification

Tools and IDEs

Java ME Secrets

Q&A

# A Quick Introduction

- Rodney Aiglstorfer
- CTO and Co-Founder of mFoundry
- Principal Architect of mWorks


Offering great flexibility, mFoundry's mWorks platform enables companies of all sizes to develop premium and ad supported applications that are distributed via carriers, content partners and retail outlets. mFoundry is a leading mobile application platform provider and publisher that provides a comprehensive solution for developing and distributing mobile applications. For more information, please visit www.mfoundry.com

# Agenda

**Java ME Realities**

What Devices to Support

Test Strategies and Certification

Tools and IDEs

Java ME Secrets

Q&A

# Java ME Realities

## Run Once…Test Everywhere

- Java ME is a collection of Java standards which are liberally interpreted by device and VM manufacturers
- There are many optional aspects of the Java ME standards which don't get supported
- Not all Java ME devices support MIDP
- Not all devices support Java ME
- Detailed specs for devices are difficult and sometimes impossible to find

# Java ME Realities

## Run Once…Test Everywhere

- Applications must be tested on every device you expect to support
- Devices can be grouped into families to minimize the number of devices you need to test on
- There are few reliable resources to help you group devices—Experience, Trial and Error are your best tools
- On-device testing cannot be automated cheaply so plan for a person to do this

# Java ME Realities

## Selling Through Carries

- Carriers are the primary method of selling an application in the United States
- They take between 40% and 60% of the sales price of your application
- They pay you every three months
- Monthly Recurring Charge (MRC), pay-per-use, and one-time payment models are supported by most carriers
- Applications must be approved and certified by the Carrier—not just anyone can sell on the carrier's deck

# Java ME Realities

## Selling “Off-Deck” (e.g. Not via the Carriers)

- Premium SMS
  - Most common way of selling in Europe
  - Requires partner with SMS Aggregator (mQube, Netsize)
  - Charge still appears on the user’s phone bill
- Credit Card payment via browser
  - Will still require a partnership with an SMS aggregator unless you are willing to have your customers type in the URL manually

# Java ME Realities

## Some Interesting Statistics

- $1.4 Billion in Carrier Revenues in 2003 which will grow to $15.5 Billion by 2008 (ARC group)
- Java technology represents 3% of data revenue in 2003 and project to 12.4% in 2008 (ARC group)
- $17 Billion will be generated in 2008 by Java based applications alone (ARC group)
- Current Global Market for mobile applications is $4–$5 Billion conservatively
- Asian Markets lead in number of downloads

# Java ME Realities

## Some Interesting Statistics

- By 2005 60% of handsets sold supported Java technology (Yankee Group)
- Installed base with Java technology exceeds 100 Million devices (Yankee Group)

# Agenda

Java ME Realities

**What Devices to Support**

Test Strategies and Certification

Tools and IDEs

Java ME Secrets

Q&A

# What Devices to Support

## Features That Will Reduce Your Supported Device List

- MIDP 1.0 vs. MIDP 2.0 baseline
- Size of JAR
- Heap
- Network connectivity
- Secure network connectivity
- Processing power
- PIM integration
- PUSH support

# What Devices to Support (Cont.)

## Features That Will Reduce Your Supported Device List

- Device and carrier specific API
- Mobile media API
- Wireless messaging API
- Bluetooth
- Dial support
- Location API
- SVG or 3D graphics support

# What Devices to Support

## Resources That Are Helpful

- Phonescoop.com
- WURFL.org
- forum.nokia.com
- developer.sonyericsson.com
- developer.cingular.com
- sprintdevelopers.com
- javaverified.com
- your competitor's web site

# Agenda

Java ME Realities

What Devices to Support

**Test Strategies and Certification**

Tools and IDEs

Java ME Secrets

Q&A

# Test Strategies and Certification

## When Do You Need to Go Through Certification?

- If you plan to sell through carrier
- If you need to sign your MIDlet
- If you need to access carrier controlled APIs
  - Video (Sprint)
  - LBS
- If you want to benefit from Java Verified™ marketing programs

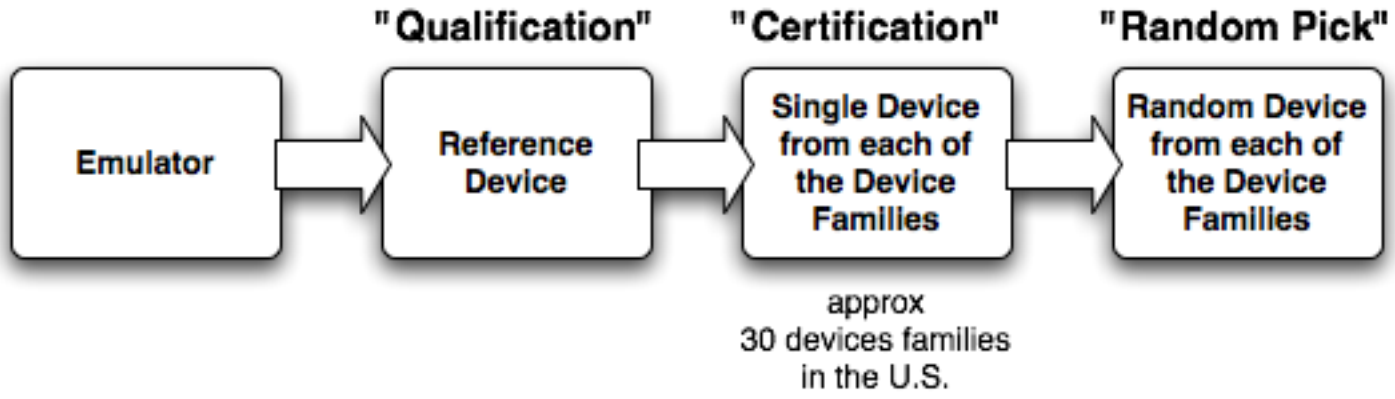
# Test Strategies and Certification

## Certification Is Not Necessary in Most Cases If:

- You plan to distribute your MIDlet free to the end user
- You plan to sell your MIDlet through an SMS aggregator using a PSMS short code

# Test Strategies and Certification

## Most Effective Acceptance Test Strategy



# Agenda

Java ME Realities

What Devices to Support

Test Strategies and Certification

**Tools and IDEs**

Java ME Secrets

Q&A

# Tools and IDEs

## What Wireless Tool Kit Should You Use?

- Sun provides a basic WTK that serves most purposes
- Device manufacturers also provide their own “skinned” WTKs and Emulators that unfortunately do not accurately simulate the real devices
- Some carriers also provide WTKs that contain libraries for their proprietary APIs
- Devices like the Blackberry must be specially compiled using tools only available in their respective WTKs

# Tools and IDEs

## What Are My Options for an IDE?

- Eclipse + Eclipse ME
- Eclipse + ME4SE (works with OS X)
- NetBeans™ Mobility Pack
- Text Editor “X” + ANT + Antenna

# DEMO

Eclipse + ME4SE

Running on OS X

# Agenda

Java ME Realities

What Devices to Support

Test Strategies and Certification

Tools and IDEs

**Java ME Secrets**

Q&A

# Java ME Secrets

**Networking Secrets**

Image Secrets

XML Secrets

RMS Secrets

MIDlet Life-Cycle Secrets

Unit Testing Secrets

Debugging Secrets

# Networking Secrets

## HTTP Is the Only Way to Go

- Anything other than HTTP connectivity should be avoided—not enough devices support UDP or Socket Connectivity
- HTTPS is not a required protocol in the MIDP 1.0 specification—that said some MIDP 1.0 devices do support HTTPS, so CHECK FIRST
- Always execute network requests in a separate thread
- There are multiple ways of downloading from a URL over HTTP(s)

# Code Sample

## Simplest (and Slowest) Way to Download Content

```
StreamConnection c = null;
InputStream s = null;
try {
    String url = "http://www.mobuser.com/atom.xml";
    c = (StreamConnection)Connector.open(url);
    s = c.openInputStream();
    int ch;
    while ((ch = s.read()) != -1) {
        ...
    }
} finally {
    if (s != null) s.close();
    if (c != null) c.close();
}
```

# Code Sample

## A Faster Approach Is to Download in “Chunks”

```
StreamConnection c = null;
InputStream s = null;
try {
    String url = "http://www.mobuser.com/atom.xml";
    c = (StreamConnection)Connector.open(url);
    s = c.openInputStream();
    int rd;
    byte[] buf = new byte[50];
    while ((rd = s.read(buf, 0, buf.length)) != -1) {
        ...
    }
} finally {
    if (s != null) s.close();
    if (c != null) c.close();
}
```

# Code Sample

## The Fastest Approach Is to Download “Fully”

```
StreamConnection c = null;
InputStream s = null;
try {
    String url = "http://www.mobuser.com/atom.xml";
    c = (ContentConnection)Connector.open(url);
    int len = (int)c.getLength();
    s = c.openDataInputStream();
    byte[] data = new byte[len];
    dis.readFully(data);
} finally {
    if (s != null) s.close();
    if (c != null) c.close();
}
```

# Code Sample

## But Beware of HTTP 1.0 and HTTP 1.1 Protocols

```
StreamConnection c = null;
InputStream s = null;
try {
    String url = "http://www.mobuser.com/atom.xml";
    c = (ContentConnection)Connector.open(url);
    int len = (int)c.getLength();
    s = c.openDataInputStream();
    if (len > 0) {
        byte[] data = new byte[len];
        dis.readFully(data);
    } else {
        int rd;
        byte[] buf = new byte[50];
        while ((rd = s.read(buf, 0, buf.length)) != -1) {
                ...
        }
    }
} finally {
    ...
}
```

# Networking Secrets

## Common Pitfalls

- Never close a connection from another thread while a read could be taking place—you will crash most phones
- Carrier Gateways often cache respones
- Carrier Gateways have been known to add, modify and sometimes remove headers from your requests—best rely on your own protocols and not the headers in the HTTP request
- Java VMs on Samsung Devices sold by GSM carriers in the U.S. have their networking layer restricted (even blocked on early devices)
- Blackberry devices have to contend with a BES in addition to the carrier gateway—be prepared for hell when developing networked applications on Blackberry

# Java ME Secrets

Networking Secrets

**Image Secrets**

XML Secrets

RMS Secrets

MIDlet Life-Cycle Secrets

Unit Testing Secrets

Debugging Secrets

# Images Secrets

## The Basics

- PNG is the only officially supported image format
- JPG is sometimes available but its best not to assume that it is and to instead transform on the server (if dynamically downloaded)
- PNG Transparency support can't be assumed in MIDP 1.0
- It is always better to send one big image instead of many little images—so create filmstrips or tile maps for images that share common dimensions such as icons or sprites
- There is no "free" support for sprites in MIDP 1.0 but it is pretty easy to do using clip rects
- Always use double buffer when possible

# Java ME Secrets

Networking Secrets

Image Secrets

**XML Secrets**

RMS Secrets

MIDlet Life-Cycle Secrets

Unit Testing Secrets

Debugging Secrets

# XML Secrets

## Pull Parsing and WBXML

- Pull Parsers are more efficient from a code size and performance perspective (www.xmlpull.org)
- WBXML is a much more efficient transmission encoding for XML instead of as text (www.w3.org/TR/wbxml/)
- The kXML parser by Stefan Haustein is the best pull parser implementation due to its small code size and additional support for WBXML

# Code Sample

## Sample Pull Parsing

```
XmlPullParser p = new XmlPullParser()
p.setInputString(source, "UTF-8");

while(p.getEventType() != XmlPullParser.DOCUMENT_END) {
    switch(p.getEventType()) {
        case XmlPullParser.START_TAG :
                ...
        case XmlPullParser.END_TAG :
                ...
        case XmlPullParser.TEXT :
                ...
    }
    p.nextToken();
}
```

# Java ME Secrets

Networking Secrets

Image Secrets

XML Secrets

**RMS Secrets**

MIDlet Life-Cycle Secrets

Unit Testing Secrets

Debugging Secrets

# RMS Secrets

## Getting the Facts—Next to Impossible

- Just because your phone has X Mb does not mean your MIDlet will be able to access that entire about
- When in doubt try an explicit setting in the JAD using the MIDlet-Data-Size property
- Some devices expect the MIDlet-Data-Size to be in bytes and others in kBytes—so beware, there is no documentation anywhere that will tell you which to use
- Don’t trust the API—find out for yourself

# Code Sample

## Accessing Available RMS Space

```
/*
 * The following is a common way to determine how much
 * space is available in a RecordStore.  But this is not
 * always a reliable.  It could mean how much is left in
 * the individual RecordStore or it could mean the total
 * amount of space left for ANY RecordStore.
 */

RecordStore rs = RecordStore.openRecordStore(...)

int availableSize = rs.getSizeAvailable();
```

# Code Sample

## On Some Devices Individual Records May Be Constrained

```
/*
 * Some older devices have a limit on the size of an
 * individual record in a RecordStore.  There is no
 * way to query what that limit is ... your only
 * options is to try to fill it until it breaks and
 * then "break up" your record into chunks no larger
 * than that limit.
 */

RecordStore rs = RecordStore.openRecordStore(...)

// this line may fail even if
// myData.length < rs.getSizeAvailable()

rs.addRecord(myData, 0, myData.length)
```

# Java ME Secrets

Networking Secrets

Image Secrets

XML Secrets

RMS Secrets

**MIDlet Life-Cycle Secrets**

Unit Testing Secrets

Debugging Secrets

# MIDlet Life-Cycle Secrets

## What Are the Life-Cycle Events in MIDP?

**`startApp()`**

- Signals the MIDlet that it has entered the Active state
- Cannot be used as a “resume” event
- Can be called when started from a destroyed state or a paused state

**`pauseApp()`**

- Signals the MIDlet to enter the Paused state.
- Not reliably implemented across MIDP 2.0 devices
- Only supported with MIDP 2.0

**`destroyApp()`**

- Signals the MIDlet to terminate and enter the Destroyed state
- Many MIDP 1.0 device terminate before this method completes

# Java ME Secrets

Networking Secrets

Image Secrets

XML Secrets

RMS Secrets

MIDlet Life-Cycle Secrets

**Unit Testing Secrets**

Debugging Secrets

# Unit Testing Secrets

## Can JUnit Be Used when Developing MIDP Applications?

- JUnit should Always be used to validate and regression test core aspects of your MIDlet's logic
- Use ME4SE as the MIDP library when running your Test Cases
- A simple MVC pattern will greatly facilitate the use of JUnit when testing a MIDlet
- Only GUI tests should need to be done by a human—everything else should be testable using JUnit + ME4SE

# DEMO

JUnit + ME4SE

MIDP Regression Testing

# Java ME Secrets

Networking Secrets

Image Secrets

XML Secrets

RMS Secrets

MIDlet Life-Cycle Secrets

Unit Testing Secrets

**Debugging Secrets**

# Debugging Secrets

## Where Do You Debug?

- Always try to debug first in the emulator
- Using ME4SE will allows you to use the Eclipse Debugger to step-through your ME code
- Some devices can stream sysouts to a console but these devices are rare in the Java ME world
- Debugging on devices is very crude—be prepared to roll your own log file that captures sysouts

# DEMO

Debugging using Eclipse + ME4SE

# Q&A

# Troubleshooting Java™ Platform, Micro Edition
## Tips from the Pros

**Rodney Aiglstorfer**

CTO and Co-Founder
mFoundry Inc.
www.mfoundry.com

TS-5927