



JavaOne

Develop Commercial Cross-Platform Solutions by Integrating Open Source Third-Party Solutions and Proprietary Code

Dario Laverde
Software Architect
mediatracker software
<http://mediatracker.com>

TS-7688



Goal of This Session

What you will learn in an hour

Learn how to design, develop and package a cross-platform, cross-device, commercial software application by integrating both closed proprietary code with open source code solutions.

Agenda

Packaging and Installation With a Java™
Runtime Environment (JRE™)

Cross-Platform and Device Development

Desktop Application Look and Feel

Server Applications and Obfuscation

Agenda

Packaging and Installation With a Java™ Runtime Environment (JRE™)

Cross-Platform and Device Development

Desktop Application Look and Feel

Server Applications and Obfuscation

Packaging and Installation

With JRE software

- Desktop vs. server (hosted vs. deliverable)
 - Assume it's already installed (a requirement)
 - Include or download the JRE software with your Application
- Licensing Issues—Historical and with new GPL
- Embedded JRE software/Java Virtual Machine (JVM™) options
 - Java Platform, Micro Edition (Java ME platform) vs. Java platform 1.x (“PersonalJava™ platform Edition”)
 - Third-Party JVM software: Kaffe, SableVM
- Why tell customer you use Java technology?
 - Native L&F Installation—Avoiding Java Web Start
 - Use open source installers: e.g., NSIS

The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Packaging and Installation (Cont.)

- Display license(s) for third-party software
 - During installation and/or first run
 - Example open source licenses: GPL LGPL BSD Apache
- Windows packaging options
 - exe, msi, zip
- Linux packaging options
 - bin, tar.gz, ipkg/ipk, rpm
- Mac OS X packaging options
 - app, dmg, zip, sit (no longer recommended)



DEMO

Sample Coding Demo

Creating a Windows Installer, a Mac OS X Application Wrapper, and a Linux ipkg

Agenda

Packaging and Installation With a Java™
Runtime Environment (JRE™)

Cross-Platform and Device Development

Desktop Application Look and Feel

Server Applications and Obfuscation

Cross-Platform and Device Development

Write once, test everywhere

- Creating a shared core code base
 - Least common denominator (as much as possible)
 - For backwards compatibility: use: `-source` `-target`
- Mimic a pre-processor
 - Public static final booleans
 - Ant tasks that clean up classes for specific targets
- Cross compile embedded VMs
 - Not just for target but for testing environment
- Maintain best practices



DEMO

Demo Illustrating a Cross-Device Build
(Targeting Different JVM Software Environments)

Agenda

Packaging and Installation With a Java™
Runtime Environment (JRE™)

Cross-Platform and Device Development

Desktop Application Look and Feel

Server Applications and Obfuscation

Desktop Application Look and Feel

The native user experience

- Install/Uninstall/Upgrade/Launch
 - Compare with all popular native apps for each OS
- Add Icons/artwork required for each platform
- Add ability to run as a background service
 - The Windows NT style service
 - Daemons/scripts for the rest
- Add tray functionality
- Make use of **Runtime.getRuntime().exec(...)**
- Use the browser for help/documentation



DEMO

Demo Wrapping JRE Version (1.x–5.x) in a Windows exe for Both Launching and Tray Functionality

Agenda

Packaging and Installation With a Java™
Runtime Environment (JRE™)

Cross-Platform and Device Development

Desktop Application Look and Feel

Server Applications and Obfuscation

Server Applications and Obfuscation

- You have more options with enterprise apps:
 - Dictate the JRE Version (e.g., 5 or 6) when recommending server box specifications (especially for dedicated servers)
 - If deliverable is for hosted environments—
Recommend one or code for the LCD JRE software and application server available at most hosting providers

Server Applications and Obfuscation (Cont.)

- Deliver your solution with a packaged application server (depending on license)
 - For example, Apache Tomcat packaged with your server application using the Apache 2 license
 - Allows for fully preconfigured and optimized solutions packaged just like a desktop application
- And just like with desktop applications, obfuscate your code
 - Just because it runs in a application server doesn't mean you can't obfuscate non-public code



DEMO

Obfuscating an Enterprise Deliverable Using
an Open Source Obfuscator

Summary

- L&F shouldn't just be in Swing; Java applications should install and run like all other native applications
- Leverage open source as much as possible
 - During development, for installation
 - Include third-party open source code/libraries
 - With deliverable or as a separate download
 - Respect licenses! If you change the source, submit changes back to community
- Consider open sourcing part of your code



DEMO

<code />

For More Information

- BOF-3373 Package, Protect, Promote: Essential Tools for Producing Competitive Commercial Desktop Applications
- TS-3290 Easy Deployment Is Finally Here
- Java Modularity JSRs: 291, 277/294



Q&A

<code />



JavaOne

Develop Commercial Cross-Platform Solutions by Integrating Open Source Third-Party Solutions and Proprietary Code

Dario Laverde
Software Architect
mediatracker software
<http://mediatracker.com>

TS-7688