Tips, Tricks, and Hidden Knowledge for Java Packagers

BOF3472

Chris Bensen
Principal Member of Technical Staff

Danno Ferrin
Principal Member of Technical Staff

Java Client Deployment and Performance
October 27, 2015

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1 ▶ Goals

2 ▶ Launcher

3 ▶ Layout of Bundled Application

4 ▶ Bundler API

5 ▶ Daemons/Services

Goals

Goals

- Improve Build Tool and IDE Integration
- Simplify Packaging Process
- Improve Platform Specific Fidelity
- Easy Distribution of Java Apps
  - App Stores
  - Alternative to Web Start and Applets
- Modular JRE support
- Reduce Size

Launcher

Macros in config file

- $APPDIR
  - Directory to root of package.

- $PACKAGEDIR
  - Directory to root + /app on Windows and Linux or Contents/Java on Mac. This is where all developer files go.

- $LAUNCHERDIR
  - Directory where the launcher lives. Same as $APPDIR on Windows and Linux or Contents/MacOS on Mac.

- $JREHOME
  - Root directory of the JRE being used.

Layout of Bundled Application

JavaOne™
ORACLE

Layout of Bundled OS X Application

```
MyAwesomeApp.app/Contents
  Java/
    MyAwesomeApp.cfg
    MyAwesomeApp.jar
    Packager.jar
  MacOS/
    MyAwesomeApp
    libpackager.dylib
  Plugins/Java.jre
```

Layout of Bundled Windows Application

```
MyAwesomeApp\
  app\
    MyAwesomeApp.cfg
    MyAwesomeApp.jar
    Packager.jar
  MyAwesomeApp.exe
  packager.dll
  msvcr120.dll
  msvcp120.dll
  runtime\jre
```

Layout of Bundled Linux Application

```
MyAwesomeApp/
  app/
    MyAwesomeApp.cfg
    MyAwesomeApp.jar
    Packager.jar
  MyAwesomeApp
  libpackager.so
  runtime/jre
```

Configuration Files

- Internal implementation detail
- .cfg are now INI files

App Data

- Mac

  ```
  ~/Library/Application Support/[app.preferences.id]/
  packager/jvmuserargs.cfg
  ```

- Windows

  ```
  C:\Users\[username]\AppData\Roaming\[app.preferences.id]
  \packager\jvmuserargs.cfg
  ```

- Linux

  ```
  ~/.local/[app.preferences.id]/packager/jvmuserargs.cfg
  ```
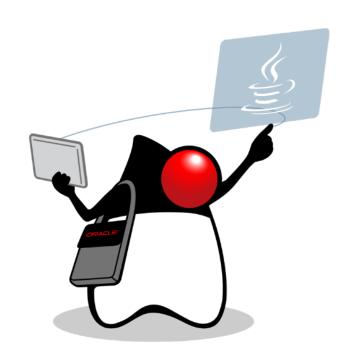
Bundler API

Bundler API Goals

- Allow for future unknown bundlers to be added to the system
- Allow tools (API/IDE) to
  - Provide useful information on success/failures
  - Provide useful information on configuration of the bundlers
- Allow for multiple platforms to share the same configuration

Daemons/Services and Bundlers Tips and Tricks

- Uses same directory layout as an installed application
- Must be "system wide" - no per user install

- Mechanism used:
  - Mac - uses launchctl
  - Windows - regular old Windows Service
  - Linux - integrated into /etc/init.d and/or invoke-rc

Mailing Lists:

openjfx-dev@openjdk.java.net

http://mail.openjdk.java.net/mailman/listinfo/openjfx-dev

Bug Reporting:

https://javafx-jira.kenai.com

Blog

https://blogs.oracle.com/talkingjavadeployment/

Integrated Cloud
Applications & Platform Services