
ORACLE®

JavaOne™

ORACLE®

Packaging Java Applications

CON3467

Chris Bensen
Principal Member of Technical Staff

Danno Ferrin
Principal Member of Technical Staff

Java Client Deployment and Performance
October 27, 2015

JavaOne ORACLE

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.

Packager Agenda

1. Where we were <= 8u40
2. Current State = 8u60
3. Future >= 9

Packager Agenda

1. Where we were <= 8u40
2. Current State = 8u60
3. Future >= 9

Name Change

- JavaFXPackager is now just JavaPackager
  - Still Packages JavaFX Apps
  - Still Packages Java Swing Apps
  - Still Packages AWT Apps
  - Still Packages SWT Apps
  - Still Packages JNLP Apps
- Use JavaPackager

8u20 API

- Configured through Map<String, ? super Object>
- Introduces BundlerParamInfo
- Bundlers can be called independently
- Provide centralized Bundlers object
  - Registers Default Bundlers
  - Registers Bundlers found via META-INF/services
  - Developers can manually register bundlers

Tooling support

- CLI uses new APIs
- JDK Bundled Ant integration uses new APIs
- Maven JavaFX plugin upgraded to use new APIs
  - Use latest 8.1.x version
- Gradle JavaFX plugin upgraded to use new APIs
  - Use latest 8.1.x version

8u40 features

- Native launcher re-write
- Command Line Arguments Defaults
- File Associations
- Multiple Launchers
- UserJVMArguments API
- Mac JRE support
- Packages Daemons and Services

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
  ```

Bundlers

- To list the bundlers that are available:
  `javapackager -help bundlers`
- To list the preferred set of bundler arguments for a particular bundler:
  `javapackager -help <bundler-id>`
- Example 1: `javapackager -help bundlers`

```
dmg - DMG Installer - INSTALLER
mac.appStore - Mac App Store Ready Bundler - INSTALLER
mac.app - Mac Application Image - IMAGE
pkg - PKG Installer - INSTALLER
```

- Example 2: `javapackager -help mac.app`

Bundler Arguments - How to specify in Ant

- Use the new fx:bundleArgument element (child to fx:deploy)

```
<fx:deploy verbose="true" outdir="awesomeApp" nativeBundles="image">

  <fx:application name="AwesomeApp" mainApp="awesome.app.Main"
                  version='1.0.42'/>

  <fx:resources id="appRes">
    <fx:fileset dir="build/jars" includes="awesomeApp.jar Util.jar"/>
  </fx:resources>

  <fx:info vendor="JavaOneDemo" title="Awesome!!1!!" category='SomeCategory'
           email='awesomeApp@example.com'/>

  <fx:bundleArgument arg="mac.CFBundleIdentifier"
                     value="Awesome Developer"/>
</fx:deploy>
```

Mac AppStore Bundler Tip

- We suggest setting mac.CFBundleVersion

```
<fx:deploy verbose="true" outdir="awesomeApp" nativeBundles="image">
…
  <fx:bundleArgument arg="mac.CFBundleVersion"
                     value="Awesome Developer”/>
</fx:deploy>
```

Service / Daemon Bundlers

- Installs your program as a service/daemon
  - Headless operations - no GUI
- Mac - PKG installer only
  - Installed into **launchctl**
- Windows - EXE and MSI Installers
  - Installed as a Service
- Linux - DEB and RPM Installers
  - Installed in **init.d**

Service / Daemon Installers

- CLI
  - new '**-daemon**' flag
- Ant
  - new '**daemon**' boolean attribute on '**fx:application**'
- Gradle and Maven
  - Use Bundler Arguments to setup

Service / Daemon Bundler Arguments

serviceHint	- Boolean	- Install as a service/daemon Default is false
runAtStartup	- Boolean	- Start service on system startup? Default is false, require manual start
startOnInstall	- Boolean	- Start service after package install? Default is false, require manual start
stopOnUninstall	- Boolean	- Stop service when removing package? Default is true, stop running service when removing package

Packager Agenda

1. Where we were <= 8u40
2. Current State = 8u60
3. Future >= 9

Tour of Bundlers

- Disk Image Bundlers
  - Mac .app
  - Windows
  - Linux
- Installer Bundlers
  - Mac - DMG, PKG, and Mac App Store Ready PKG
  - Windows - Wix Toolset .msi and InnoSetup .exe
  - Linux - Debian .deb and RPM .rpm
- JNLP Bundler NEW!

Mac AppStore Submission Issues

- 8u60 WebKit links with Apple's "private" APIs

```
ubrk_getRuleStatus
ubrk_setUText
ucnv_getCanonicalName
ucnv_reset
ucol_strcollIter
```

- APIs are for the text system ICU (www.icu-project.org) that reside in the QtWebKit library
- Fix the problem
  - JIRA: https://bugs.openjdk.java.net/browse/JDK-8138652

Mac AppStore Submission Issues

- Workaround Fix is scheduled for next year
- Automatically strips /lib/libjfxwebkit.dylib for bundled JRE
- JIRA: https://bugs.openjdk.java.net/browse/JDK-8138650
- Webrev: http://cr.openjdk.java.net/~shemnon/8138650/webrev.00/
- Changeset: http://hg.openjdk.java.net/openjfx/8u-dev/rt/rev/761213753af4

Mac AppStore Submission Issues

- Workarounds:
  - Remove /lib/libjfxwebkit.dylib from the JRE you are going to bundle

Packager Agenda

1. Where we were <= 8u40
2. Current State = 8u60
3. Future >= 9

JDK 9

These features are very tentative

- Module support (Project Jigsaw)
- JEP 275: https://bugs.openjdk.java.net/browse/JDK-8080531
- webrev: http://cr.openjdk.java.net/~shemnon/8080531/webrev.04
- Repository: http://hg.openjdk.java.net/openjfx/sandbox-9-jake/rt
- Changeset: http://hg.openjdk.java.net/openjfx/sandbox-9-jake/rt/rev/4dc807c092f6

JDK 9 Modules

- java.activation
- java.annotations.common
- java.base
- java.compact1
- java.compact2
- java.compact3
- java.compiler
- java.corba
- java.datatransfer
- java.desktop
- java.instrument
- java.logging
- java.management
- java.naming
- java.prefs
- java.rmi
- java.scripting
- java.se
- java.security.jgss
- java.security.sasl
- java.smartcardio
- java.sql
- java.sql.rowset
- java.transaction
- java.xml.bind
- java.xml.crypto
- java.xml
- java.xml.ws
- javafx.base
- javafx.controls
- javafx.fxml
- javafx.graphics
- javafx.media
- javafx.swing
- javafx.web
- jdk.accessibility
- jdk.attach
- jdk.charsets
- jdk.compiler
- jdk.crypto.ec
- jdk.crypto.pkcs11
- jdk.deploy
- jdk.deploy.osx
- jdk.hotspot.agent
- jdk.httpserver
- jdk.internal.le
- jdk.internal.opt
- jdk.jartool
- jdk.javadoc
- jdk.javaws
- jdk.jcmd
- jdk.jconsole
- jdk.jdeps
- jdk.jdi
- jdk.jdwp.agent
- jdk.jfr
- jdk.jlink
- jdk.jvmstat
- jdk.localedata
- jdk.management.cmm
- jdk.management
- jdk.management.resource
- jdk.naming.dns
- jdk.naming.rmi
- jdk.pack200
- jdk.packager
- jdk.packager.services
- jdk.plugin.dom
- jdk.plugin
- jdk.policytool
- jdk.rmic
- jdk.scripting.nashorn
- jdk.scripting.nashorn.shell
- jdk.sctp
- jdk.security.auth
- jdk.security.jgss
- jdk.snmp
- jdk.xml.bind
- jdk.xml.dom
- jdk.xml.ws
- jdk.zipfs

JDK 9 Tools

- jlink
- jmod
- jdeps

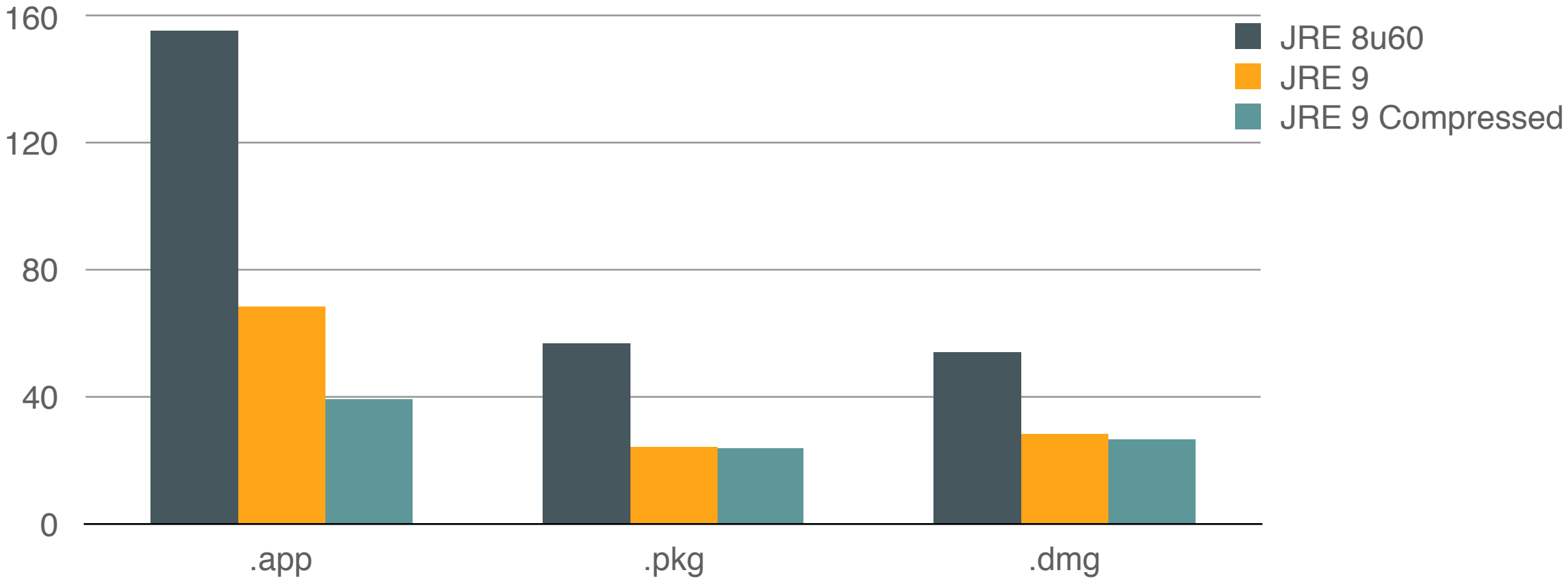
jlink Compression

```
0: Constant strings sharing, 30% size reduction, negligeable startup impact
1: ZIP compression, around 50%+ size reduction, small startup impact
2: ZIP+Constant strings sharing, 55%+ size reduction, small startup impact

By default, all resources are compressed. You can express the set of
Additional Options:  --compress-resources-level <0|1|2> --compress-
```

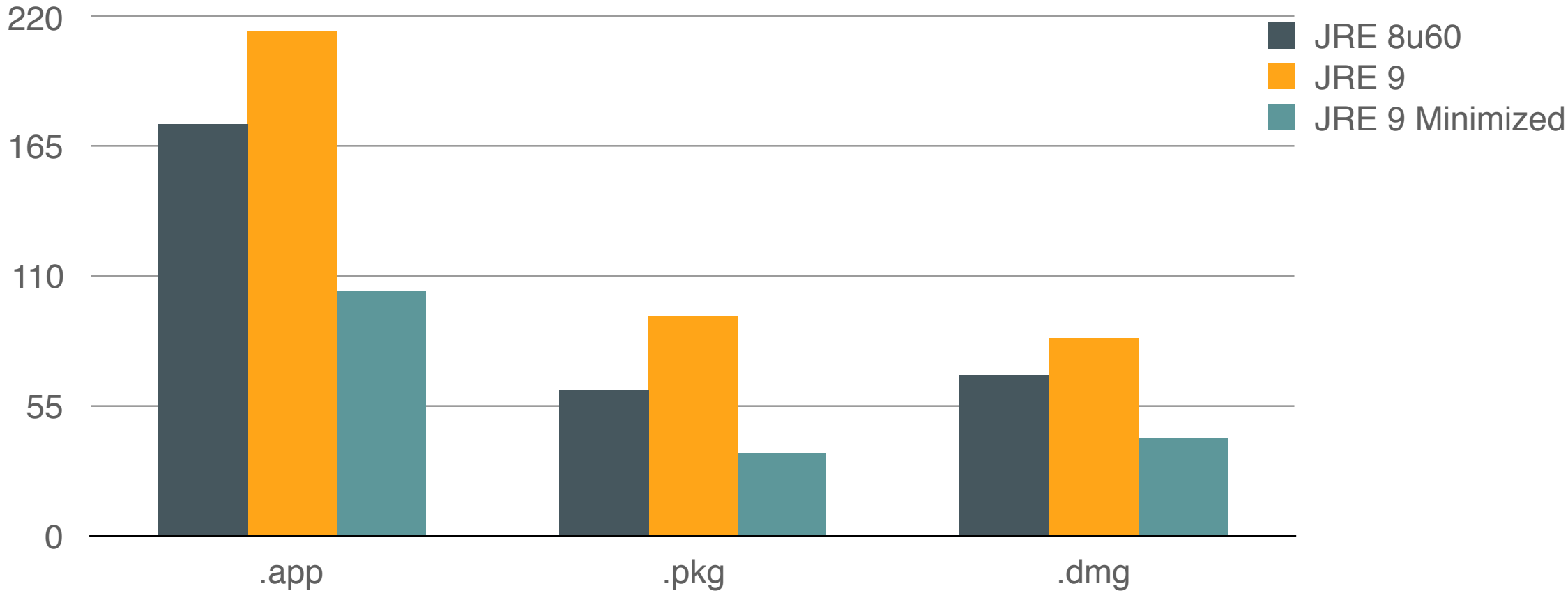
Up to **59%** App Size Reduction!

The Module Advantage



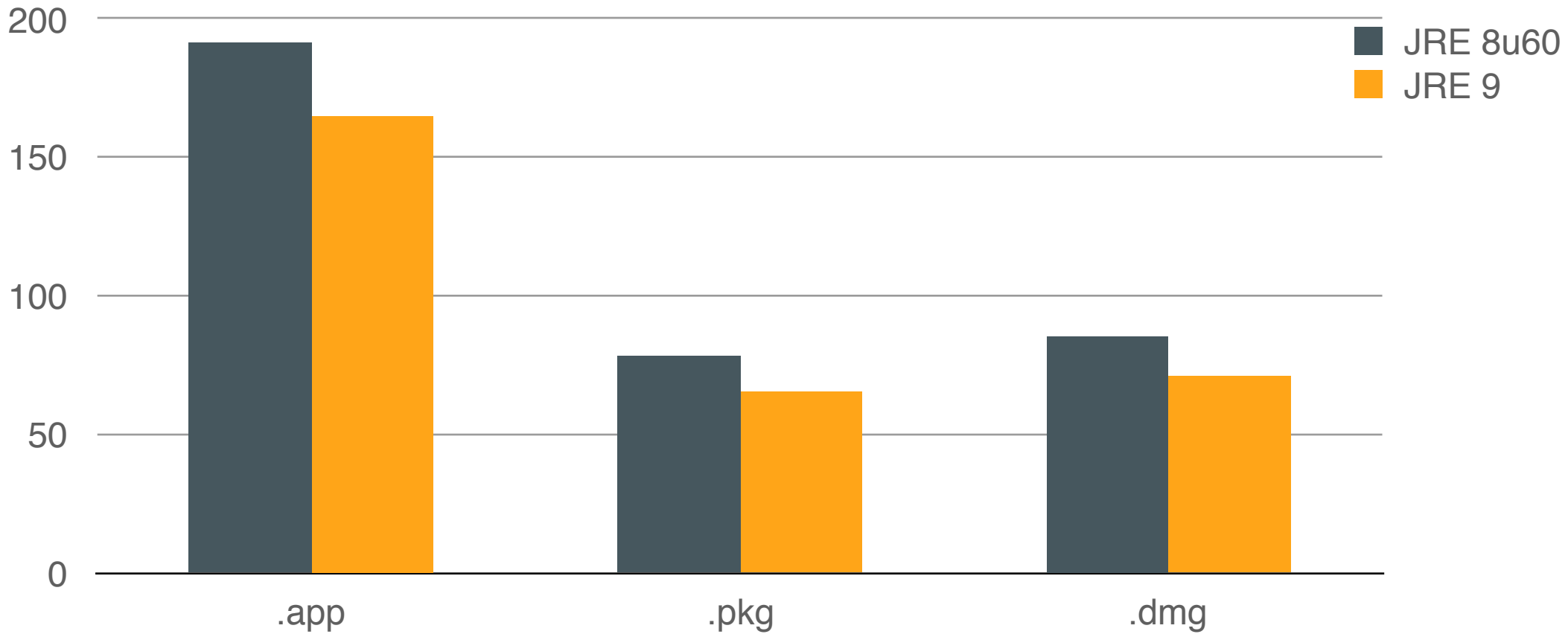
Example is a Hello World Swing App. Your App may vary.

The Module Advantage

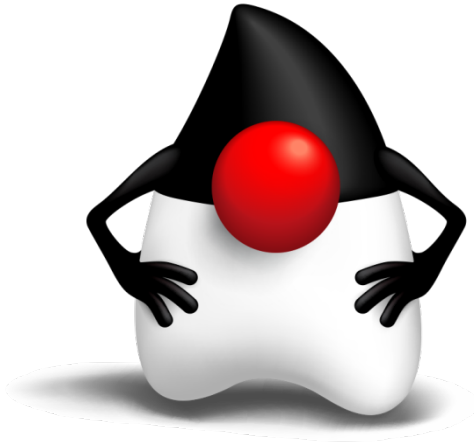


Example is a Hello World JavaFX App (not compressed). Your App may vary.

The Module Advantage



Example is Ensemble JavaFX App. Your App may vary.



Demo

Windows App Store - Major Roadblocks

- The Windows app store only accepts WinRT apps
- Apps that link against many of the traditional Windows APIs are banned (GDI, GDI+, etc)
- http://blogs.msdn.com/b/win8devsupport/archive/2012/12/20/porting-desktop-apps-to-windows-store-apps.aspx

Future Features Under Consideration*

These features are very tentative

- JDK 9 Module support
- ~~Auto Memory Configuration~~
- Windows Authenticode Signing
- Make Packaging Simpler
- Jigsaw Integration (Generate small, modular JRE)
- Auto Update (for when you don't have an App Store)
- Single Instance

* Subject to change

Future Features Under Consideration Continued…

These features are very tentative

- Re-writing Windows Executable MetaData
- Native Java Splash Screen Support
- Java Packager UI
- More Platform Fidelity Support
- Convert high resolution icon to platform icon
- General purpose install time and first run app capabilities
- Any Audience Requests?

Jigsaw Links

- http://openjdk.java.net/projects/jigsaw/
- http://openjdk.java.net/projects/jigsaw/quick-start

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

JavaOne™

ORACLE®


ORACLE®