

ORACLE®



JavaOne™

ORACLE®

Packaging and Deploying Java Apps in Java 8u20

CON2247

CREATE
THE
FUTURE

Danno Ferrin
Principal Member of Technical Staff
Java Client Deployment and Performance
October 2, 2014

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.

What's in a Name

- JavaFXPackager is now just JavaPackager
 - Still Packages JavaFX Apps
 - Still Packages Java Swing Apps
 - Still Packages AWT Apps
 - Still Packages SWT Apps
 - Now Packages Daemons and Services

Program Agenda

- 1 ▶ Peek Behind the Curtain: API Re-write
- 2 ▶ Bundler Arguments
- 3 ▶ Service / Daemon Support
- 4 ▶ Tour of Bundlers
- 5 ▶ 8u40 / JEP-208 features

Program Agenda with Highlight

- 1 Peek Behind the Curtain: API Re-write
- 2 Bundler Arguments
- 3 Service / Daemon Support
- 4 Tour of Bundlers
- 5 8u40 / JEP-208 features

Why re-write the API?

- Tooling
 - Better IDE integration
 - Better build tool integration
- Decoupling bundlers
 - Reduce bundler cross-interference
 - Third party bundlers

API Pre 8u20

- Configured through BundleParams
 - Followed JavaBeans conventions
 - Exposed all config to all bundlers
- Invoked through PackagerLib
 - Also included JavaFX Tasks like css2bss
 - Always Executed JNLP
 - Made fine tuned execution difficult

API post 8u20

- Configured through `Map<String, ? super Object>`
- Introduces `BundlePramInfo`
- Bundlers can be called independently
- Provide centralized `Bundlers` object
 - Registers Default Bundlers
 - Registers Bundlers found via `META-INF/services`
 - User 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

BundlerParamInfo

- Provides User Facing Descriptions of Configuration Items
- Provides tooled shorthand names for params
- Provides typing hints
 - Only hints, generics not reifiable (yet)
e.g. **List<String>** looks like **List**
- Hooks for default values and conversion from **String** values

Program Agenda with Highlight

- 1 Peek Behind the Curtain: API Re-write
- 2 Bundler Arguments**
- 3 Service / Daemon Support
- 4 Tour of Bundlers
- 5 8u40 / JEP-208 features

BundlerArguments

- All configuration has been re-cast to be name/value pairs
- Existing configuration classes have been changed under the covers to use the name/value pair in the `Map<String, ? super Object>` configurations
- This means end users can now add configuration parameters known only to the bundlers
 - Mac - Signing Identities, App Store Category, etc.
 - Win - UUIDs, Start Menu Group, etc.
 - Linux - Maintainer e-mail, OSS License, etc.

Bundler Arguments - How to specify in the CLI

- Use the new `-B<name>=<value>` flag

```
javapackager -deploy -verbose -outdir awesomeAppDir -outfile AwesomeApp \  
  -native image -vendor JavaOneDemo -name AwesomeApp \  
  -description "This App Is Awesome!" -srcdir build/jars \  
  -srcfiles awesomeApp.jar:Util.jar -title "Awesome!!1!!" \  
  -appclass awesome.app.Main \  
  -BapplicationCategory=SomeCategory \  
  -BappVersion=1.0.42 \  
  -Bemail=awesomeApp@example.com \  
  "-Bmac.signing-key-user-name=Awesome Developer"
```

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.signing-key-user-name"  
    value="Awesome Developer" />  
</fx:deploy>
```


Bundler Arguments - How to specify in Maven

- Use the `<bundleArguments>` element

```
<plugin>
  <groupId>com.zenjava</groupId>
  <artifactId>javapackager-maven-plugin</artifactId>
  <version>8.1.3</version>
  <configuration>
    <mainClass>awesome.app.Main</mainClass>
    <appName>Awesome App</appName>
    <bundleType>image</bundleType>
    <bundleArguments>
      <mac.signing-key-user-name>Awesome Developer</mac.signing-key-user-name>
      <!-- other configs go here, omitted for brevity -->
    </bundleArguments>
  </configuration>
</plugin>
```

Bundler Arguments - How to specify in Gradle

- Use the `bundleArguments` block (inside the JavaFX block)

```
javafx {
```

```
    packaging = 'image'  
    vendor = 'JavaOneDemo'  
    name = 'AwesomeApp'  
    description = 'This App Is Awesome!'  
    applicationClass = 'awesome.app.Main'  
    category = 'SomeCategory'  
    version = '1.0.42'
```

```
    bundleArguments {  
        email = 'awesomeApp@example.com'  
        'mac.signing-key-user-name' = 'Awesome Developer'  
    }
```

```
    // you can also use map syntax: bundleArguments = ['a':'b']
```

```
}
```

Program Agenda with Highlight

- 1 Peek Behind the Curtain: API Re-write
- 2 Bundler Arguments
- 3 Service / Daemon Support**
- 4 Tour of Bundlers
- 5 8u40 / JEP-208 features

Service / Daemon Bundlers

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

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 |

Program Agenda with Highlight

- 1 Peek Behind the Curtain: API Re-write
- 2 Bundler Arguments
- 3 Service / Daemon Support
- 4 Tour of Bundlers**
- 5 8u40 / JEP-208 features

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

Common Bundler Arguments - General

- **name** - The name of the application. Default value for stuff like file name, app bundle name, shortcut name, etc.
 - CLI: `-name` argument
 - Ant: `fx:deploy/fx:application/@name` attribute
- **appVersion** - The version of the application.
 - Ant: `fx:deploy/fx:application/@version`

Common Bundler Arguments - Command Line

- **jvmOptions** - The JVM Options to apply to the JVM.
 - Ant: `fx:deploy/fx:application/fx:jvmArg`
- **jvmProperties** - Java Properties to set on the JVM.
 - Ant: `fx:deploy/fx:application/fx:jvmProperty`
- **applicationClass** - The Main Class to execute.
 - Ant: `fx:deploy/fx:applicaiton/@mainClass`

Common Bundler Arguments - Classes, Jars, and Assets

- **appResources** - All of the files that go in the application directory. Jars, media assets, help files, etc.
 - CLI: **-srcdir** and **-srcfiles** arguments (can be added multiple times)
 - Ant: **fx:deploy/fx:resources** elements
- **classpath** - The classpath for the executed application.
 - Can be derived from **appResources**
- **mainJar** - The jar containing the main class. Derived by packager if not set.
 - Can be derived from **appResources** and **applicationClass**

Common Bundler Arguments - Runtime

- **runtime** - The JRE/JDK to bundle with the application
 - Ant: `fx:deploy/fx:platform/@basedir`
 - File path to the Java Virtual Machine to bundle
 - Empty String - Try and use System JVM
 - JVM will be stripped of unneeded file
 - Binary Executables
 - Man Pages
 - Source jars
 - etc. etc.

Mac .app Bundler Specific Bundler Arguments - codesign

- **mac.signing-key-developer-id-app**
 - The full name of the Developer ID application signing key.
 - Pass through value to the **-s** argument for **codesign**.
- **mac.bundle-id-signing-prefix**
 - Pass through value to the **-prefix** argument for **codesign**.

Mac .app Bundler Specific Bundler Arguments

- **icon.icns**
 - File path to the icns version of the application icon.
- **mac.category**
 - Mac App Store encoded category type.
- **mac.CFBundleName**
 - PList value for CFBundleName
- **mac.CFBundleIdentifier**
 - PList value for CFBundleIdentifier

Mac DMG Bundler Specific Bundle Arguments

- **licenseFile**
 - File containing text of a click-through license before DMG is opened up.
- **systemWide**
 - Whether drag-to link is Applications (true) or Desktop (false).

Mac PKG Bundler Specific Bundle Arguments

- **licenseFile**
 - File containing text of a click-through license before DMG is opened up
- **mac.signing-key-developer-id-installer**
 - The full name of the Developer ID installer signing key.
 - Pass through value to the **-s** argument for **codesign**.
- Service/Daemon bundler Arguments
 - Service will be installed so that **launchctl** can control it.

Mac App Store Ready Bundler Specific Bundle Arguments

- **mac.signing-key-app / mac.signing-key-pkg**
 - The full name of the “3rd Party Mac” Application/Installer signing key.
 - Pass through value to the **-s** argument for **codesign**.
- **mac.app-store-entitlements**
 - Build time path to the entitlements file to use during code signing.

Windows Disk Image Bundler Specific Bundle Arguments

- **icon.ico**
 - Application icon in .ico format.

Windows EXE Bundler Specific Bundle Arguments - Metadata

- **copyright**
 - Copyright field in EXE installer.
- **vendor**
 - Publisher as seen in the Uninstaller screen.
- **title**
 - Comments in uninstall screen selection

Windows EXE Bundler Specific Bundle Arguments

- **systemWide**
 - Whether to install in Program File (true) or in user home directory (false)
- **win.menuGroup**
 - Name of windows start menu/screen group
- **shortcutHint / menuHint**
 - Whether to install a shortcut on the desktop/start menu.
 - If both are false, menuHint will be forced true.
- Service / Daemon Bundle Arguments

Windows MSI Bundler Specific Bundle Arguments - Metadata

- **description**
 - Description in MSI metadata.
- **vendor**
 - Authors in MSI metadata.
 - Publisher in Uninstall screen.

Windows MSI Bundler Specific Bundle Arguments

- **productVersion**
 - Product Version UUID for product updates.
- **systemWide**
 - Whether to install in Program File (true) or in user home directory (false).
- **win.menuGroup**
 - Name of windows start menu/screen group.
- **shortcutHint / menuHint**
 - Whether to install a shortcut on the desktop/start menu.
 - If both are false, menuHint will be forced true.
- Service / Daemon Bundle Arguments

Linux Disk Image Bundler Specific Bundle Arguments

- Surprisingly, nothing!

Linux DEB / RPM Bundler Specific Bundle Arguments

- **linux.bundleName**
 - Installed bundle name. Derived from name, but subject to restrictions.
- **category**
 - Category for system desktop menu.
- **icon.png**
 - Icon for system desktop menu, as a PNG.
- **licenseFile**
 - File containing the license text. Not click-through.

Linux DEB Bundler Specific Bundle Arguments - MetaData

- **copyright**
 - Added to license and copyright template.
- **description**
 - Description field in Daemon init script.
- **title**
 - Description field in ``dpkg -s``.
- **vendor / email**
 - Combined to form the Maintainer field in ``dpkg -s``.

Linux RPM Bundler Specific Bundle Arguments - MetaData

- **description**
 - Description in ``rpm -qi``
- **licenseType**
 - License field in ``rpm -qi``
- **title**
 - Summary field in ``rpm -qi``
 - Comment in `.desktop` file.
- **vendor**
 - Vendor in ``rpm -qi``

Program Agenda with Highlight

- 1 Peek Behind the Curtain: API Re-write
- 2 Bundler Arguments
- 3 Service / Daemon Support
- 4 Tour of Bundlers
- 5 8u40 / JEP-208 features**

New BundlerArgs in 8u40

- **mac.CFBundleVersion**
 - Machine readable version.
 - Mac App Store requires incrementing app versions.
- **mac.dmg.simple**
 - Create a DMG w/o applescript integration.
 - Uglier, but applescript tends to hang build servers.

Planned 8u40 features

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

Planned 8u60 Features

These features are very tentative

- Re-writing Windows Executable MetaData
- Native Java Splash Screen Support

- Any Audience Requests?



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/>

Hardware and Software Engineered to Work Together



JavaOne™

ORACLE®

ORACLE®