

EquoIDE and SolsticeOSGi

A new minimalist runtime for Eclipse applications

<https://github.com/equodev/equo-ide/issues/60>

Eclipse Webinar 2023-Jan-25 #60

Open

nedtwigg opened this issue now · 0 comments



nedtwigg commented now

Collaborator



We will be presenting more info on this project as part of [a webinar](#) at 6am PST on Jan 25 2023. We'll post a recording and the slides to this issue after the talk.

Equo Tech, Inc. Jan 25 2023.

Quick demo (Gradle)

```
./gradlew equoIde
```

- Run it on your machine at <https://github.com/diffplug/spotless/pull/1523>

```
plugins {  
→ // https://github.com/equodev/equo-ide/blob/main/plugin-gradle/CHANGELOG.md  
→ id 'dev.equo.ide' ..... version '0.12.1'  
}  
equoIde {  
→ branding.title('Spotless').icon(file('_images/spotless_logo.png'))  
  
→ // waiting on https://github.com/equodev/equo-ide/pull/65  
→ //welcome\(\).openUrlOnStartup\('https://github.com/diffplug/spotless/blob/main/CONTRIBUTING.md'\)  
  
→ // everything below becomes the default after https://github.com/equodev/equo-ide/pull/66  
→ p2repo 'https://download.eclipse.org/eclipse/updates/4.26/'  
→ install 'org.eclipse.releng.java.languages.categoryIU'  
→ install 'org.eclipse.platform.ide.categoryIU'  
  
→ p2repo 'https://download.eclipse.org/buildship/updates/e423/releases/3.x/3.1.6.v20220511-1359/'  
→ install 'org.eclipse.buildship.feature.group'  
}
```

Quick demo (Maven)

```
./mvnw equo-ide:launch
```

```
<build><plugins><plugin>
  ·· <groupId>dev.equodev</groupId>
  ·· <artifactId>equo-ide-maven-plugin</artifactId>
  ·· <version>0.11.0</version>
  ·· <configuration>
  ·· ·· <branding>
  ·· ·· ·· <title>My IDE</title>
  ·· ·· ·· <icon>${project.basedir}/imgs/my_icon.png</icon>
  ·· ·· ·· <splash>${project.basedir}/imgs/my_splash.png</splash>
  ·· ·· </branding>
  ·· ·· <p2repos>
  ·· ·· ·· <p2repo>https://download.eclipse.org/eclipse/updates/4.26/</p2repo>
  ·· ·· ·· <p2repo>https://download.eclipse.org/technology/m2e/releases/2.1.2/</p2repo>
  ·· ·· </p2repos>
  ·· ·· <installs>
  ·· ·· ·· <install>org.eclipse.platform.ide.categoryUI</install>
  ·· ·· ·· <install>org.eclipse.reलग.java.languages.categoryIU</install>
  ·· ·· ·· <install>org.eclipse.m2e.TODO</install>
  ·· ·· </installs>
  ·· </configuration>
```

<https://github.com/equodev/equo-ide/issues/18>

Use m2e to import "this" project #18

 nedtwigg opened this issue on Dec 6, 2022 · 0 comments



nedtwigg commented on Dec 6, 2022

When using the Maven plugin, the default install will include [m2e](#) and a fresh import of the current project. This is required for Equo IDE to hit `1.0` levels of usability.

The power of OSGi

- Load and unload plugins dynamically
- Lazy initialization of plugins -> fast startup
- Multiple versions of a class can coexist
 - e.g. Guava 15 and Guava 19 at the same time, thanks to `Bundle.loadClass` instead of `Class.forName`

The cost of OSGi

- You have to do Import-Package and Export-Package right.
- You can't do Class.forName
- It's not enough to have the classes right. You also have to have the metadata right.
- Developers are not learning it anymore.

<https://trends.google.com/trends/explore?date=all&q=OSGi>



The power of SolsticeOSGi

- ```
BundleContext context = new dev.equ.solstice.AtomosFrontend(installDir).getBundleContext();
```
- Find every OSGi bundle on the classpath.
- At runtime, modify the OSGi headers
  - Remove all metadata related to versions.
  - If a bundle requires something which is missing, remove that requirement.
- Use Atomos to start a single-class loader OSGi
  - <https://github.com/apache/felix-atomos> (huge thanks to Thomas Watson!)
- **If the classes are there, it will run.**
- Spend your time learning about your customers and their problems, don't spend it learning OSGi and p2.

# The costs of SolsticeOSGi

- ~~Load and unload plugins dynamically~~
  - Workaround: restart the IDE when a plugin changes
- ~~Lazy initialization of plugins → fast startup~~
  - Workaround: our “slow” eager approach still feels fast.
- ~~Multiple versions of a class can coexist~~
  - If your project can't update its transitive dependencies to their latest available version, your project has died.
  - Bring it back to life! Update your deps, raise your minimum JRE!

# The power of Eclipse p2

- It can manage licenses.
- It can install and manage java runtimes.
- It can install binary artifacts set their permission bits to executable on unix machines.
- It can use SAT4J boolean reasoning library to determine when multiple versions of a transitive library are required.
- **It can download jars.**
- To do all of this, p2 needs a lot of code and a lot of transitive dependencies.



# The power of Solstice p2

- It only does jars.
- ~2400 LOC and two transitive deps (org.tukaani:xz and okhttp)
- [https://github.com/equodev/equo-ide/blob/main/P2\\_MULTITOOL.md](https://github.com/equodev/equo-ide/blob/main/P2_MULTITOOL.md)

- `equoList --all=categories`
  - `equoList --all=features`
  - `equoList --all=jars`
- `equoList --installed`
- `equoList --problems`
- `equoList --optional`
- `equoList --detail=any.unit.id`
- `equoList --raw=any.unit.id`

```
0 unmet requirement(s), 0 ambiguous requirement(s). For more info: `gradlew equoList
60 optional requirement(s) were not installed. For more info: `gradlew equoList --o
+-----+-----+
| maven coordinate / p2 id | repo |
+-----+-----+
com.ibm.icu:icu4j:72.1	maven central
commons-io:commons-io:2.11.0	maven central
jakarta.servlet:jakarta.servlet-api:4.0.4	maven central
```

# Want to build an Eclipse plugin?

- Learn OSGi



- 
- Learn p2
- Now start to solve your problem.

# Want to build an EquoIDE Hook?

- Don't learn OSGi or p2. Just put the dependencies on your classpath.
- Now write classes that solve your problem.
  - Don't write metadata.
  - Don't even *have* metadata.
- Simplest possible IDE Hook
  - <https://github.com/equodev/equo-ide/pull/65>
- Gradle Buildship import
  - <https://github.com/equodev/equo-ide/pull/66>

# Want to use Eclipse dependencies?

- Until recently, Eclipse did not publish to maven central.
- Now some are on maven central, but most Eclipse projects are still only available via p2.
- Use Solstice to mix dependencies from maven and p2.
- Example from Spotless
  - Calculate p2 dependencies
  - Download from maven where possible
  - Download from p2 where not
  - <https://github.com/diffplug/spotless/pull/1524>

# Summary

- Eclipse and its community are amazing.
- Very difficult to get started as an Eclipse contributor.
  - OSGi and p2 are complex.
  - They can be optional!
- You can develop Eclipse IDEs, applications, and plugins without learning OSGi or p2.