

Maven vs Gradle

This is a working document on the topic of converting from gradle to maven.

If you add items / questions / etc, please tag and maybe color it (e.g., <dl>...) to make it easier to track things. Open questions / issues are in red.

At a high level the motivations are:

- improve consumability of Edgent by users
 - add publishing of signed Edgent binary release artifacts (JAR, WAR) in maven repos
 - ultimately cease creating signed binary release bundles (tgz)
- simplify the build tooling
 - the gradle scripting has some non-standard / complicated processing either out of ignorance or to achieve certain things
 - TODO add the "why the complications" note here. (manifest classpath, binary bundle tgz with transitive deps, ...)
 - Suspect that to simplify maven based tooling we'd need to eliminate some of those things (which would also simplify gradle stuff)

Chris Dutz created [PR-309](#) for working on a maven based system.

PR-309 now includes a maven wrapper (mvnw). Set JAVA_HOME and then use "mvnw" instead of "mvn" below and you don't have to manually install maven.

Or, to manually install maven

<https://maven.apache.org/download.cgi>

<https://maven.apache.org/install.html>

Maven concepts: lifecycles, phases, goals, plugins

High level building and testing

- mvn clean install # builds and tests add -DskipTests to omit testing, --fail-at-end for "continue"-like
- mvn clean package # a bit fewer phases than install
- mvn clean compile # even fewer/faster
- mvn site:site site:staging # see target/stagingindex.html
- mvn dependency:copy-dependencies # copies all dep jars into various target/dependencies
- see below for building for multiple target platforms (j7, android)

Things I need to educate myself on / all need to understand:

- how do we generate an official signed source release bundle (tgz)
 - Today
 - gradlew release # also builds binary tgz bundle
 - gradlew signAll # generates .asc, .md5, .sha
 - we have scripts to stage / publish to ASF svn repos
 - Mvn
 - mvn install -Papache-release
 - see item #9 below
- how do we generate a rat report for source release artifacts
 - Today
 - gradlew rat # we have customizations that exclude various things
 - Mvn
 - Automatically generated as part of the normal maven build - build fails if anything isn't OK
- how do we generate junit and jacoco HTML reports
 - Today
 - gradlew reports
 - reports/tests/overview-summary.html # JUnit
 - reports/coverage/index.html # Code coverage
 - Mvn
 - Normal Text and XML reports are generated in the test-phase
 - HTML reports are created by the site plugin: mvn clean install site:site (The site is generated in target/site directory)
- how do we just compile (no test build / execution)
 - Today
 - gradlew assemble
 - Mvn
 - Compile tests but don't execute them: mvn install -DskipTests
 - Don't compile tests: mvn install -Dmaven.test.skip=true
- how do we run all/some java8 tests (and build if needed)
 - Today
 - gradlew test --continue # all non-Manual tests
 - gradlew <project>:test --tests *.SomeTest
 - gradlew <project>:test --tests *.SomeTest.someMethod
 - gradlew <project>:cleanTest <project>:test # force rerun
 - gradlew connectors:kafka:test --tests ".*Manual" # after setup&start kafka server

- gradlew connectors:mqtt:test --tests '.*Manual' # after setup&start mqtt server
 - gradlew connectors:jdbc:test # after setup DERBY_HOME
 - Mvn
 - mvn test --fail-at-end
 - mvn -pl <project> test -Dtest=**/SomeTest test
 - mvn -pl <project> test -Dtest=**/SomeTest#someMethod test
 - There is no such concept as continuing a run ...
 - mvn -pl connectors/kafka test '-Dtest=**/*Manual' test
 - mvn -pl connectors/mqtt test '-Dtest=**/*Manual' test

how do Edgent runtime developers develop w/Eclipse

- Today
 - gradlew setupExternalJars # populates ../externalJars/... with ext deps
 - have .classpath files in repo # these reference ../externalJars/...
 - then just usual java project auto-build
 - can run test with <project>Test > main > Run as > JUnit test
 - Mvn
 - In IntelliJ all just works
 - have Eclipse mvn plugin (bundled in Eclipse IDE for Java Developers - Neon/4.6.1)
 - pom.xml > Run As > Maven > Existing Maven projects
 - The IDE is automatically configured by the settings in pom.xml and the normal java auto build should continue to work.
 - <dlaboss> seems to work fine now since the Eclipse .project/.classpath files have been removed in the PR

how do we build and access javadoc

- Today
 - gradlew assemble or gradlew release or gradlew aggregateJavadoc
 - builds javadoc into build/distributions/docs/javadoc
 - Mvn
 - Automatically built by the maven site plugin: mvn clean install site:site (The site is generated in target/site directory)

how do we build and test for java8 tgt

- Today
 - gradlew test # build all deps, run all non-Manual tests
 - Mvn
 - mvn test (Would compile and run the tests)

how do we build and test for java7 tgt

- Today
 - gradlew assemble # generates java8, java7, android tgt class/jar files
 - gradlew test7Compile # build appropriate tests
 - # muck with EVs to setup to use java7 JRE
 - # ...
 - gradlew test7Run # run w/java7
 - gradlew tests7Reports # JUnit and coverage reports
 - Mvn
 - ~~Generate a binary assembly for Java8 or Java7~~
 - ~~Create a maven profile for java7 (unfortunately I haven't quite understood the differences of the Java7 and the Android build. There is quite a lot of code that doesn't compile for Java7)~~
 - <dlaboss> Edgent is written to a restricted Java8 - basically only Java7 + Lambdas. The gradle/ant machinery for building for java7 is to build with java8 and then generate java7 jars using "retrolambda". See platform/java7/android/build.gradle (and leveraged build.xml).
 - <dlaboss> the current state of the PR (10-July) is to use mvn toolchains and Profiles to include building/testing alternate platforms
 - create a toolchains-local.xml file by cloning the toolchains-travis.xml file in the repo and setting the appropriate j8-jdk-home and j7-jdk-home locations on your system
 - e.g., on my osx system:
 - j8 jdkHome: /Library/Java/JavaVirtualMachines/jdk1.8.0_112.jdk/Contents/Home
 - j7 jdkHome: /Library/Java/JavaVirtualMachines/jdk1.7.0_80.jdk/Contents/Home
 - then build/test via: mvn -t toolchains-local.xml -Djava8.home=\$JAVA_HOME clean install -Pplatform-java7,platform-android -DskipTests # omit -DskipTests to run tests

how do we build and test for android tgt

- Today
 - android tgt is basically the java7 artifacts plus one or two more jars (all included with java8 build)
 - we don't have tests / test runs for the android tgt - ~not an issue since it includes the java7 jars, which we do test
 - Mvn
 - ~~Generate a binary assembly for Android~~
 - ~~Create a maven profile for Android (unfortunately I haven't quite understood the differences of the Java7 and the Android build. There is quite a lot of code that doesn't compile for Java7)~~
 - <dlaboss> see java7 above

A not quite 2 hour introduction video to Maven:

<https://vimeo.com/167857327>

Issues / TODOs - things without strikethrough still need attention

27Oct Summary of things still needing attention (RED items below)

- 23Oct animal-sniffer config not detecting J8 API uses (JDBC, MQTT, File connector tests) - emailed Chris
 - 28Oct animal sniffer only checks compile code, not test code. Problems like this will be revealed by running the tests with the toolchain profile.
 - 23Oct changes to Jenkinsfile not used by Jenkins? - emailed Chris
 - 28Oct the Jenkinsfile is only picked up by the pipeline build. There is a manually setup build (<https://builds.apache.org/view/E-G/view/Edgent/job/edgent-dev/>) which the Jenkinsfile has no effect on.
 - 23Oct "mvn test" fails (console-server copy fails: "Artifact has not been packaged yet") - emailed Chris
 - In case of a "mvn test", the jars are not created as the jars are created in the package phase. This is the reason that module fails in this case (Don't know how to avoid this however)
 - item 9 source release bundle - various things
 - 12Sep - do we need some file that documents, in aggregate, the external dependencies Edgent has (with what license/notice info)? Related, see the "samples" item.
 - item 12 website pages - various things
 - item 15 samples - post merge split out into its own repo
 - item 17 (WISH) mvn Edgent Application archetype
-
1. ~~(done) the PR adds html comments (ALv2 headers) to many markdown file (e.g., README.md). As it turns out, that further confuses the Eclipse "wikitext" editor and the preview panel now shows blank for the file. cheesh, as if some of the braindead behaviors of that editor weren't already enough 😊 SOLUTION: install the "Markdown text editor" from the Eclipse marketplace and adjust file associations accordingly. Open the Markdown View to see the preview. Probably need to put something about this in DEVELOPMENT.md for Eclipse.~~
 2. ~~(done) Need to put some doc / warning somewhere as we get lots of these warnings and~~
 - a. ~~mvn clean install~~

```
[INFO] Scanning for projects...  
  
[WARNING]  
  
[WARNING] Some problems were encountered while building the effective model for org.apache.edgent.analytics:edgent-analytics-sensors:jar:1.2.0-SNAPSHOT  
  
[WARNING] 'version' contains an expression but should be a constant. @ org.apache.edgent.analytics:edgent-analytics-sensors:${edgent.version}, /Users/dlaboss/git/incubator-edgent/t-4/analytics/sensors/pom.xml, line 31, column 12
```

 - b. ~~Chris: Unfortunately we have to live with that. Using variables for the versions can cause problems. In our case I made sure they don't. But we need that in order to create the java 8 and java 7 packages. But Maven tends to be over careful~~
 3. ~~(done) the connector-kafka pom isn't preserving the non-transitive dep specs that are noted in connectors-kafka/build.gradle (see commentary there)~~
 - a. ~~the effects are that a binary bundle (assembly) or flat dir of ext dependencies will have extra things in it. Ditto when creating an uber jar for an Edgent app (or sample)~~
 - b. ~~the pom's specs can be made to match the non-transitive gradle decls/behavior by adding type and exclusion decls like the following.~~
 - i.

```
<dependency>  
  <groupId>org.apache.kafka</groupId>  
  <artifactId>kafka_2.10</artifactId>  
  <version>0.8.2.2</version>  
  <type>jar</type>  
  <exclusions>  
    <exclusion>  
      <groupId>*</groupId>  
      <artifactId>*</artifactId>  
    </exclusion>  
  </exclusions>  
</dependency>
```
 - ii. ~~(17.07.2017) Done (not done; was only added to kafka_2.10)~~
 - iii. ~~(01Aug) still need the exclusions on ALL of the non-Edgent dependencies, not just kafka_2.10~~
 - iv. ~~(01Aug) committed changes to add it to ALL~~
 4. ~~(done) same non-transitive spec issue for the analog to android/hardware/build.gradle in android/android/hardware/pom.xml~~
 - a. ~~dlaboss: actually this is a case where "provided" is the correct dependency config. The platform where the Edgent app is deployed should have the dependent "com.google.android:android:4.1.1.4" jar/deps.~~
 5. time observations:
 - a. it is much slower (minutes slower) to build with mvn than gradle. not an issue when developing in an IDE I guess.
 - b. 0m35 time mvn compile # any "install" includes populating the mvn repo
 - c. 0m38 time mvn clean compile # mvn => no build avoidance I think
 - d. 3m46 time mvn package -DskipTests -Papache-release # j8, includes javadoc at least to a certain level, created source bundle, not signed
 - e. 2m47 time mvn clean install -DskipTests # only builds j8
 - f. 2m38 time mvn install -DskipTests # that's right after the prior mvn cmd

- g. 7m9 time mvn -t toolchains-local.xml -Djava8.home=\$JAVA_HOME install -Pplatform-java7,platform-android -DskipTests # ~similar to "gradlew assemble" I think (but gw includes javadoc)
- h. 10m8 time mvn -t toolchains-local.xml -Djava8.home=\$JAVA_HOME install -Pplatform-java7,platform-android,apache_release -DskipTests # ~similar to "gradlew release" (gw doesn't run rat, but does create binary tgz and javadoc)
- i. 0m39 time gradlew cleanAll assemble # includes javadoc gen
- j. 0m7 time gradlew assemble
- k. 0m57 time gradlew release # j8/j7/android, includes source & binary tgz but not signed, no rat
- l. TESTING...
- m. 10m51 time gradlew test # j8
- n. 10m56 time gradlew cleanAll test # j8
- o. 13m16 time mvn clean install # j8
- p. junit and coverage html reports?
- q. ability to run j7 tests without redoing j8 tests?
- 6. (done) Apache Edgent: Samples: Scenarios — spelling
- 7. (done) Apache Edgent: Runtimes — use to be singular "runtime"
 - a. (11.07.2017) Well I did notice, that there were multiple runtimes so I thought better name the artifact that way (I would suggest to rename the directory to "runtimes" as well as the other directories containing multiple things all have plural names)
 - b. Hmm... unlike connectors, providers, samples, "runtime" is the aggregator for various parts of "the" core Edgent runtime (singular) — though some of the pieces of it are optional hence the finer grained jars. The various runtime sub-projects aren't different implementations of "the" runtime. I would feel better if we didn't change this level of naming at this time. Can you live with that? It can always be revisited after the conversion is complete and the dust settles. Note, at least one person suggested consolidating various projects — e.g., maybe a single api jar and single runtime jar.
 - c. it's been renamed back to runtime
- 8. importing maven projects into Eclipse
 - a. (done) 12Sep — 1 warning remains after importing pom.xml and "Overriding managed version 3.6 for maven-site-plugin" if appropriate to retain, annotate it to squelch it
 - b. (done) — 100 pom warnings for imported maven projects in Eclipse
 - i. lots of unnecessary version id specs (same as parent — remove them?)
 - ii. some plugin version overrides (can squelch them)
 - c. (fixed) 7 errors for android&distribution poms — but otherwise all seems fine
 - i. the import (of all projects/poms) generated a "Setup Maven plugin connectors" dialog. 7 Errors: 1 for maven-toolchain-plugin, 3 for retrolambda plugin process:main, 3 for retrolambda plugin process:test.
- 9. source release bundle - names/content
 - a. 12Sep — MUST exclude maven-wrapper.jar from bundle (bundle can't have binary content)
 - i. (fixed) 18Sep Done — (though at the moment still getting the gradle-wrapper.jar — cleanup pending)
 - b. 12Sep — DEVELOPMENT.md — see comments there for addressing Building and Jenkins chapters
 - i. 18Sep Processed the second TODO (Jenkins and SonarQube)
 - ii. 26Sep <dl> I did some work on the Building part and think it's OK now.
 - c. (fixed) 12Sep — it's very easy for swill (e.g., editor backup files like README.md —) to creep into the source bundle. Can some exclusions be added to the pom?
 - i. 18Sep Added an exclusion for "*" files ... would need a test to verify
 - d. 12Sep — eventually exclude samples as they will be in their own release bundle (see "samples" section below). — see commentary in top-level pom.xml
 - i. (fixed) 18Sep Done
 - e. 12Sep - do we need some file that documents, in aggregate, the external dependencies Edgent has (with what license/notice info)? Related, see the "samples" item.
 - f. (done) 12Sep — mvn build warning — duplicate decl of maven-assembly-plugin
 - g. via mvn clean install — Papache-release — DskipTests
 - i. see <http://maven.apache.org/plugins/maven-gpg-plugin/usage.html> for passphrase config info
 - h. mvn generated name is currently: edgent-parent-1.2.0-SNAPSHOT-source-release.zip
 - i. (17.07.2017) Done should now be "apache-edgent-incubating-1.2.0-SNAPSHOT-source-release.zip"
 - i. LICENSE: (need to remove no longer bundled jquery... code), NOTICE: OK
 - i. (18.07.2017) Done
 - j. formats: only zip now?
 - i. (18.07.2017) Now also tar.gz is generated.
 - k. checksums: missing md5 and sha?
 - i. (18.07.2017) These are usually generated by Nexus when uploading artifacts to Nexus
 - l. (done) README updated for new j8/j7/android build steps
 - m. (done) reminder to update JAVA_SUPPORT.md. The mvn mods result in java7 platform now having all of the Edgent components present in java8. Ditto for Android (minus the usual things).
 - n. all tests pass when building from source bundle?
 - i. I seem to recall Chris disabled some SSL related tests? — Others?
 - 1. (18.07.2017) Think I had some troubles in the beginning, but I think all are on again ... at least all that were on when I started.
 - 2. <dl> 01Aug all tests are passing (build/run from rope AND from source bundle) and only skipped tests are correctly: JDBC and Kafka connectors and one in File connector
 - ii. (done) do the manual kafka and jdbc tests pass?
 - 1. <dl> 01Aug — kafka manual tests pass (setup kafka, the "cd connectors-kafka; mvn test -Dtest=**/*Manual")
 - 2. <dl> 01Aug — jdbc tests fail — easily fixed by adding dependency to derby and letting mvn download it. Now JDBC-connector tests always run — no longer need manual run. Added change request to the PR.
 - o. (done) JAVA_SUPPORT.md: needs updating for j7&android
 - p. (done) RELEASE_NOTES (summary of PR309 changes), CONTRIBUTORS
 - q. (post-merge is OK) src/site/*: status of this?
 - i. (18.07.2017) Just a dummy for now, but this is the place where some detailed documentation with graphics charts and so on could go to and could be automatically generated in the site of the project (automatically updating the website)
 - r. (post-merge is OK) Failing Tests.txt: status of this?
 - i. (18.07.2017) All automated tests are successful will have to check the manual ones as they seem to have been dropping below my radar.

- s. (post-merge is OK) **TODO: file JIRA to track cleaning up other cruft from source bundle (and repo): gradle stuff, appropriate source and binary bundle licensing stuff, PROBLEMS.md**
- 10. ~~binary release jars were names/content, ... for j7 and android too~~
 - a. ~~04Oct - java7 and android jars (other than console jar/war) missing LICENSE and NOTICE~~
 - b. ~~12Sep - MANIFEST/LICENSE - missing stuff for the bundled d3/query/gson/metrics-core stuff in the war~~
 - i. (23.09.) Should be done by now
 - ii. 25Sep <dl> not yet... following up on dev@
 - iii. 04Oct war's LICENSE/NOTICE all set now (including j7/android)
 - c. ~~12Sep - MANIFEST/NOTICE - missing the ibm contribs one for most jars and the war. And the war's is missing notices for bundled things noted above?~~
 - i. (25.09.) Should every jar/war/tar/zip contain the ibm contribs suffix?
 - ii. (25.09.) done
 - iii. (25Sep) <dl> not yet... needs correct initial Edgent copyright date. Furthermore, for the WAR, the bundled metrics-core 3.1.2 has a NOTICE (<https://github.com/dropwizard/metrics/blob/3.1-maintenance/NOTICE>) that may need to be addressed (depends on outcome of 10.a.ii and whether or not we need full license copies of the WAR bundled external components)
 - iv. 04Oct jar/war NOTICE all set now
 - d. (ok) 12Sep - the WAR now contains edgent jars. Is that tolerable/OK?
 - e. (done) 12Sep - setup for staging at ASF nexus (any setup for ultimate publish to maven central)
 - i. 18Sep This is already done. If I was to change "clean install" to "clean deploy" on the ASF Jenkins we would start publishing Maven artifacts to Apache Maven repo and when doing a release it would automatically deploy the release artifacts to a staging repo. As soon as the release vote passes, we set the staging repo to "release" and the artifacts get released to Maven Central.
- 11. what needs to be edited to do a non-SNAPSHOT release? (see related info under "build scripts" below)
 - a. (11.07.2017) I wrote all of this down for the Flex project: [Releasing FlexJS with Maven](#)
- 12. Website "documentation" (RM guide, getting started, website downloads) - can't really update it till we merge and make a release
 - a. 12Sep - updated getting started page (need to preserve old too)
 - b. 12Sep - updated website downloads page - including info on how to create your own binary bundle tarball from released jars
 - c. 12Sep - RM Guide (need to preserve old for now) - leaving "undone" until we really have fully doc'd steps / flow comparable to current content's completeness
 - i. (23.09.) If "RM Guide" refers to the documentation for a Release Manager to do a release ... I guess the guide I wrote for FlexJS could be just what we need: [Releasing FlexJS with Maven](#)
- 13. updating any of the release mgmt build scripts (noted in the RM-guide) to deal with switch to maven and/or changes in binary artifacts / bundling
 - a. hmm... the current scripts learn the version-id of the release from the gradle.properties file.
 - b. related, all of the pom.xml files have the version-id in them. Maybe create a new properties file with the version-id info that can also be used to drive a new script that edits the poms?
 - c. (11.07.2017) With maven there should be no need for any scripts like this as Maven provides all means to do releases out of the box.
- 14. ~~binary release BUNDLE~~
 - a. <dl> ~~ASSERTION: no longer release binary bundle (with or without ext deps)~~
 - i. (post-merge is OK) cleanup binary release/(LICENSE, NOTICE, README) in the github repo
 - ii. ~~MUST have a easy way to get/download all edgent binary artifacts and ext deps~~
 - 1. ~~get-edgent-jars.sh (need to fix kafka poms - item #3; how to handle WAR? related item #15)~~
 - b. ~~otherwise, have correct bundle name, content, ...~~
 - c. (11.07.2017) ~~It should be producing a distribution in the distribution module (quite new module) (<dl> it's ok to retain distribution module and we just not use it at this time)~~
- 15. samples
 - a. sent @dev email on 19Jul http://mail-archives.apache.org/mod_mbox/incubator-edgent-dev/201707.mbox/ajax/%3C29D9F1D6-EC63-40CE-890A-341D479D76A6%40apache.org%3E
 - i. see samples/README.md and APPLICATION_DEVELOPMENT.md
 - b. (done) do not include samples binary jar in a release
 - c. (fixed 20Sep) 12Sep Edgent Console not working - browser page comes up but graph portion of it is missing
 - i. e.g., (cd samples/topology; ./run-sample.sh DevelopmentSample) and open reported console URL
 - d. (fixed) 20Sep regression in Edgent Console
 - i. hover on stream has "No value - counter not present" at bottom instead of tuple count
 - e. (ok now) 12Sep ensure get-edgent-jars.sh and generated classpath.sh handles things for Edgent Console (e.g., collects war file)
 - f. 12Sep get-edgent-jars.sh refers to a TODO URL for documentation of Edgent's 3rd party dependencies and their license/note info
 - g. 12Sep add a separate samples source bundle to the release - "apache-edgent-samples-...".
 - i. (done) Exclude samples in the main source release bundle - TODO "samples" issues/comments in top-level pom.xml and platforms/java7/pom.xml
 - 1. (23.09.) Should be done by now
 - ii. **Chris to address by migrating samples to separate repo once everything else is OK**
 - 1. (23.09.) I would like to address this as soon as everything is merged back. Then I will create the appropriate Infra ticket and request a new repo based on the content of the samples sub-dir (This way the history is preserved)
 - iii. FYI currently: to generate a samples source bundle in samples/target: (cd samples; ./mvnw package -DskipTests -Papache-release)
 - h. (done) have non-Edgent parent dependent pom(s), supporting uber jar and j8, j7, android building - see samples/README.md
 - i. (done) include package-app.sh (see samples/README.md)
 - j. (done) include get-edgent-jars.sh
 - k. (done) updated doc/READMEs for all samples; see new samples/README.md for overview, and how to build and run
 - l. (done) samples build with mvn and run via uber jar
 - m. (done) in lieu of, or in addition to, a Edgent application archetype, add a simple/clonable app template sample: samples/template
- 16. javadoc
 - a. 12Sep - javadoc (target/site/apidocs/index.html) problems
 - i. edgent_overview.html Getting Started / Samples and Building Applications sections need rework for new environment
 - 1. 27Sep adjusted (it now just references our website's edgent getting started). Post release we can consider more refactoring.
 - 2. 18Sep if the edgent_overview should be part of the generated site, it should be moved to "src/site/???" I would suggest to migrate it to asciidoc or markdown format.
 - ii. (fixed) is missing overview from edgent_overview.html
 - iii. (fixed) lacks grouping

- iv. (fixed) includes things it shouldn't (e.g., connector "runtime" packages, fvt/swt stuff)
- v. For reference see content at <https://edgent.apache.org/javadoc/latest/index.html>. See gradle based custom config at <https://github.com/apache/incubator-edgent/blob/master/gradle/javadoc.gradle>
- b. 12Sep - all of the project links are broken on the generated target/site/index.html
 - i. 18Sep - The links in the "target/site" directory are usually broken, but if you run "mvn site:stage" the root modules target directory will contain a "target/staging" directory which should contain the aggregated content of all "target/site" directories. Then all links should work.
- 17. (WISH) mvn archetype for Edgent app
 - a. 12Sep (post-merge is OK). how will a user get the code supporting this?
 - i. (23.09.) The user will use the default maven archetype plugin and simply pass in the artifact-id of our edgent archetype and maven will generate an empty edgent application which the user can then start filling with life.
- 18. (done) build requires Java7 JDK?
 - a. Chris mentioned working on eliminating that requirement. Not sure if Java7 is only required for Pplatform-java7 and then only for test execution.
 - b. (25.09.2017) should be done now.
- 19. (done) Jenkins
 - a. (post merge is OK) setup for auto nightly? builds via Jenkins — Chris, what's needed for this?
 - b. (post merge is OK) related: consider having travis PR validation only do j8 (to speed things up)
 - c. (25.09.2017) should be done now. (-)
- 20. (done) websocket connector
 - a. component/package/jar names were a mess before the mvn work (my bad)
 - b. retain package, class names; filed JIRAs for subsequent cleanup ([EDGENT-427](#), [EDGENT-426](#))
 - c. rationalize jar/component names with mvn work — only change the jar names once
 - i. simplify
 - ii. use "websocket" as the high level connector name
 - iii. it is implicitly a websocket **client** connector; make "server" the special case (we don't provide/release a server side connector)
 - iv. other than external jetty jars, goal is to have a single jar for the connector — edgent-connectors-websocket-~~ever~~-jar
 - 1. connector will implicitly be JSR356/javax.websocket based (no javax in the names)
 - 2. connector will implicitly be Jetty javax.websocket based

1.1.0 Jar	30June-PR-309	Want	Notes
edgent.connectors.wsclient-javax.websocket	edgent-connectors-websocket-wsclient-<ver>	edgent-connectors-websocket-<ver>	Contains o.a.e.c.wsclient.javax.websocket.Jsr356WebSocketClient JSR356/javax.websocket based impl of the connector Directly used by Edgent app, needs to add to classpath Wish class was o.a.e.c.websocket.WebSocketStreams
edgent.connectors.wsclient	edgent-connectors-wsclient-<ver>	edgent-connectors-websocket-base<ver> <TBD remove jar>	Contains o.a.e.c.wsclient.WebSocketClient The generic Edgent WebSocket Connector Client API (independent of JSR356/javax.websocket API) Motivation was there are other non-javax.websocket websocket client impls out there. Edgent app needs to add to classpath Wish class was o.a.e.c.websocket.api.IWebSocketStreams / gone TBD move into edgent-connectors-websocket component/jar or completely eliminate
edgent.javax.websocket	edgent-connectors-websocket-<ver>	edgent-connectors-websocket-misc-<ver> <TBD remove jar>	Things a JSR356/javax.websocket based impl needs (invisible to Edgent client app) Defines a JSR356 SSL Hackery interface - o.a.e.javax.websocket.EdgentSslContainerProvider try to remove this jar in the future (e.g., by combining into the other jar)? 1.1.0 manifest adds external JSR356/javax.websocket API definition jar to classpath Wish class was o.a.e.c.websocket.runtime.EdgentSslContainerProvider TBD move into edgent-connectors-websocket component/jar
javax.websocket-client	edgent-connectors-websocket-client-<ver>	edgent-connectors-websocket-jetty-<ver> <TBD remove jar>	Things for a Jetty-based impl of JSR356/javax.websocket (invisible to Edgent client app) Has impl for the JSR356 SSL Hackery - o.a.e.javax.websocket.impl.EdgentSslContainerProviderImpl 1.1.0 manifest adds Jetty jars to classpath; service decl for the SSL hackery impl

			<p>Wish class were o.a.e.c.websocket.runtime.jetty. EdgentSslContainerProviderImpl</p> <p>TBD move into edgent-connectors-websocket component/jar</p>
javax.websocket-server	edgent-connectors-websocket-server-<ver>	edgent-connectors-websocket-server-<ver>	Just for websocket client testing. Not part of binary release.