

Runtime Infrastructure - Bug #6671

Build configurations need "fine tuning"

08/12/2022 10:50 AM - Roger Borrello

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		case_num:	
billable:	No	version:	
vendor_id:	GCD		
Description			
Related issues:			
Related to Conversion Tools - Feature #6669: load all conversion rules from t...		New	

History

#1 - 08/12/2022 04:47 PM - Roger Borrello

- Related to Feature #6669: load all conversion rules from the jar by default and honor the file system if it is there added

#2 - 08/12/2022 04:47 PM - Roger Borrello

- Subject changed from Build configurations are not accurate to Build configurations need "fine tuning"

The 3821c_14154 revision allows the Hotel ChUI server to start using the fwdServer deployment, and the client to start using the fwdClient deployment.

I don't believe this task is going to pose an issue with the progress of #6214, as we plan to use the fwdConvert configuration, which currently is everything (until [#6669](#) is resolved).

In the meantime, any notes related to the configurations can be kept here.

#4 - 09/02/2022 11:11 AM - Roger Borrello

I ran 2 builds of 3821c, the difference between them was build A had janusgraph-core 0.1.0 as part of both the Convert and Server configurations, and build B only included it in the Convert configuration.

The result is the inclusion of these jars in build A's server configuration:

```
commons-configuration-1.10.jar
commons-lang3-3.3.1.jar
gremlin-core-3.2.3.jar
gremlin-groovy-3.2.3.jar
gremlin-shaded-3.2.3.jar
groovy-2.4.7-indy.jar
groovy-2.4.7.jar
groovy-console-2.4.7.jar
groovy-groovysh-2.4.7-indy.jar
groovy-json-2.4.7-indy.jar
groovy-jsr223-2.4.7-indy.jar
groovy-swing-2.4.7.jar
groovy-templates-2.4.7.jar
groovy-xml-2.4.7.jar
high-scale-lib-1.1.4.jar
hppc-0.7.1.jar
```

```
ivy-2.3.0.jar
jBCrypt-jbcrypt-0.4.jar
janusgraph-core-0.1.0.jar
javatuples-1.2.jar
javax.json-1.0.jar
jcabi-log-0.14.jar
jcabi-manifests-1.1.jar
jcl-over-slf4j-1.7.21.jar
jline-2.12.jar
metrics-core-3.0.1.jar
reflections-0.9.9-RC1.jar
snakeyaml-1.15.jar
spatial4j-0.4.1.jar
tinkergraph-gremlin-3.2.3.jar
```

That build also does **not** include:

```
javassist-3.12.1.GA.jar
reflections-0.9.5.jar
```

That's a hefty number of dependencies brought in by janusgraph. As for the reduced jars, it looks like reflections-0.9.9-RC1.jar is brought in which supersedes reflections-0.9.5.jar, which must be a dependency of something else in the server configuration. javassist-3.12.1.GA.jar doesn't appear to have the same circumstance (something superseding it).

When attempting to run the server (of hotel) with the p2j linked to build A, it starts fine. When trying to use build B, which is greatly reduced, the server doesn't start:

```
com.goldencode.p2j.cfg.ConfigurationException: Initialization failure
    at com.goldencode.p2j.main.StandardServer.hookInitialize(StandardServer.java:2167)
    at com.goldencode.p2j.main.StandardServer.bootstrap(StandardServer.java:1050)
    at com.goldencode.p2j.main.ServerDriver.start(ServerDriver.java:499)
    at com.goldencode.p2j.main.CommonDriver.process(CommonDriver.java:518)
    at com.goldencode.p2j.main.ServerDriver.process(ServerDriver.java:217)
    at com.goldencode.p2j.main.ServerDriver.main(ServerDriver.java:874)
Caused by: java.lang.NoClassDefFoundError: org/apache/commons/lang3/tuple/Pair
    at com.goldencode.p2j.persist.orm.RecordMeta.<init>(RecordMeta.java:294)
    at com.goldencode.p2j.persist.orm.DmoClass.assembleImplementation(DmoClass.java:442)
    at com.goldencode.p2j.persist.orm.DmoMetadataManager.registerDmo(DmoMetadataManager.java:241)
    at com.goldencode.p2j.persist.meta.MetadataManager.initialize(MetadataManager.java:448)
    at com.goldencode.p2j.persist.DatabaseManager.initialize(DatabaseManager.java:1091)
    at com.goldencode.p2j.persist.Persistence.initialize(Persistence.java:852)
    at com.goldencode.p2j.main.StandardServer$10.initialize(StandardServer.java:1295)
    at com.goldencode.p2j.main.StandardServer.hookInitialize(StandardServer.java:2163)
    ... 5 more
Caused by: java.lang.ClassNotFoundException: org.apache.commons.lang3.tuple.Pair
    at java.net.URLClassLoader.findClass(URLClassLoader.java:387)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
    ... 13 more
```

#5 - 09/02/2022 11:21 AM - Hynek Cihlar

Roger, you will have to add the required dependencies, previously included by Janus graph, manually back to the Server runtime configurations.

#6 - 09/02/2022 11:28 AM - Roger Borrello

Hynek Cihlar wrote:

Roger, you will have to add the required dependencies, previously included by Janus graph, manually back to the Server runtime configurations.

How can I make that determination of required dependencies? Is there a tool? The gradle dependencies report just reports what is configured in build.gradle.

#7 - 09/02/2022 11:31 AM - Hynek Cihlar

Roger Borrello wrote:

Hynek Cihlar wrote:

Roger, you will have to add the required dependencies, previously included by Janus graph, manually back to the Server runtime configurations.

How can I make that determination of required dependencies? Is there a tool? The gradle dependencies report just reports what is configured in build.gradle.

The tool is the compiler and runtime testing. Apparently there are only runtime issues left. The typical failure to look for will be the missing Java class you found. But there may be more tricky issues, for example the implementations may alter behavior depending on the available classes (dependencies). I'm afraid there is no simple automated way.

#8 - 09/02/2022 12:14 PM - Roger Borrello

I suppose that's how we got to this point. It's difficult if you don't keep the dependencies as light as possible as you are adding functionality.

Right now the build.xml archive task doesn't use the dist/convert directory, opting for collecting build/lib/** because of [#6669](#). But also, the other configs are not very accurate, so the dist/server and dist/client directories are not "trustworthy", as you mentioned. And attempts to modify for vulnerabilities, like [#6643](#), become quite the challenge. :-(

#9 - 01/16/2023 05:23 PM - Roger Borrello

I found that `src/4gl/embedded/**` was missing from the **archive** task for the `"${dist.home}/fwd_deploy-convert_${major}.${minor}.${release}_${repo}_${branch}_${rev}_${today_date}.zip"` file. Let me know if this can be merged to trunk.

The change is:

```
=== modified file 'build.xml'
--- old/build.xml      2022-11-25 21:44:40 +0000
+++ new/build.xml      2023-01-13 20:55:17 +0000
@@ -5,7 +5,7 @@
 ** Module      : build.xml
 ** Abstract    : ant build script for the FWD project
 **
- ** Copyright (c) 2004-2022, Golden Code Development Corporation.
+ ** Copyright (c) 2004-2023, Golden Code Development Corporation.
 **
 **  _#_ _I_ _Date_ _JPRM_ _Description_
 ** 001 GES 20041130  @18843 First version based on samples from ECF.
@@ -240,6 +240,7 @@
 **      TJD 20220502          Fix antlr generated files for Java 11 warnings
 **      TJD 20220804          Make it compatible with both Java 8 and Java 11
 **      SBI 20221121          Included web il8n resources com/goldencode/p2j/ui/client/driver/web/tr/*.json
+ ** 104 RFB 20230113          Updated the archive.convert_destfile to include the src/4gl/embedded/** director
Y.
-->

<!--
@@ -1015,6 +1016,7 @@
      <zipfileset dir="${basedir}"
          includes="version.properties
                  rules/**
+                  src/4gl/embedded/**
+                  src/com/goldencode/**" />
      <!-- Ignore ./convert until #6669 <zipfileset dir="${dist.home}/convert" prefix="build/lib" includes=
***" /> -->
    </zip>
```