

## Runtime Infrastructure - Bug #7589

### org.slf4j.impl.StaticLoggerBinder when deploying p2j.jar in tomcat

07/19/2023 07:14 AM - Constantin Asofiei

|                        |         |                        |           |
|------------------------|---------|------------------------|-----------|
| <b>Status:</b>         | Closed  | <b>Start date:</b>     |           |
| <b>Priority:</b>       | Normal  | <b>Due date:</b>       |           |
| <b>Assignee:</b>       | Galya B | <b>% Done:</b>         | 100%      |
| <b>Category:</b>       |         | <b>Estimated time:</b> | 0.00 hour |
| <b>Target version:</b> |         | <b>case_num:</b>       |           |
| <b>billable:</b>       | No      | <b>version:</b>        |           |
| <b>vendor_id:</b>      | GCD     |                        |           |
| <b>Description</b>     |         |                        |           |

#### History

##### #1 - 07/19/2023 07:17 AM - Constantin Asofiei

- Priority changed from Normal to Urgent

A customer deploys p2j.jar in a tomcat web app. Having org.slf4j.impl.StaticLoggerBinder defined results in problems like this when starting the web app:

```
org.springframework.beans.factory.support.ScopeNotActiveException: Error creating bean with name 'scopedTarget.sessionScopedDataV5': Scope 'session' is not active for the current thread; consider defining a scoped proxy for this bean if you intend to refer to it from a singleton; nested exception is java.lang.IllegalStateException: No thread-bound request found: Are you referring to request attributes outside of an actual web request, or processing a request outside of the originally receiving thread? If you are actually operating within a web request and still receive this message, your code is probably running outside of DispatcherServlet: In this case, use RequestContextListener or RequestContextFilter to expose the current request.
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:383)
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:208)
    at org.springframework.aop.target.SimpleBeanTargetSource.getTarget(SimpleBeanTargetSource.java:35)
    at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:195)
    at com.sun.proxy.$Proxy62.toString(Unknown Source)
    at org.slf4j.helpers.MessageFormatter.safeObjectAppend(MessageFormatter.java:299)
    at org.slf4j.helpers.MessageFormatter.deeplyAppendParameter(MessageFormatter.java:271)
    at org.slf4j.helpers.MessageFormatter.arrayFormat(MessageFormatter.java:233)
    at org.slf4j.helpers.MessageFormatter.arrayFormat(MessageFormatter.java:173)
    at com.goldencode.p2j.util.logging.CentralSlf4jLogger.format(CentralSlf4jLogger.java:994)
    at com.goldencode.p2j.util.logging.CentralSlf4jLogger.debug(CentralSlf4jLogger.java:343)
    at ...
    at ...
    at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
```

and this :

```
at com.sun.jmx.mbeanserver.JmxMBeanServer.invoke(JmxMBeanServer.java:801)
at com.sun.jmx.remote.security.MBeanServerAccessController.invoke(MBeanServerAccessController.java:468)
at javax.management.remote.rmi.RMIConnectionImpl.doOperation(RMIConnectionImpl.java:1468)
at javax.management.remote.rmi.RMIConnectionImpl.access$300(RMIConnectionImpl.java:76)
at javax.management.remote.rmi.RMIConnectionImpl$PrivilegedOperation.run(RMIConnectionImpl.java:1309)
at java.security.AccessController.doPrivileged(Native Method)
at javax.management.remote.rmi.RMIConnectionImpl.doPrivilegedOperation(RMIConnectionImpl.java:1408)
at javax.management.remote.rmi.RMIConnectionImpl.invoke(RMIConnectionImpl.java:829)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at sun.rmi.server.UnicastServerRef.dispatch(UnicastServerRef.java:357)
```

```
at sun.rmi.transport.Transport$1.run(Transport.java:200)
at sun.rmi.transport.Transport$1.run(Transport.java:197)
at java.security.AccessController.doPrivileged(Native Method)
at sun.rmi.transport.Transport.serviceCall(Transport.java:196)
at sun.rmi.transport.tcp.TCPTransport.handleMessages(TCPTransport.java:573)
at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run0(TCPTransport.java:834)
at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.lambda$run$0(TCPTransport.java:688)
at java.security.AccessController.doPrivileged(Native Method)
at sun.rmi.transport.tcp.TCPTransport$ConnectionHandler.run(TCPTransport.java:687)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:750)
```

Caused by: java.lang.IllegalStateException: No thread-bound request found: Are you referring to request attributes outside of an actual web request, or processing a request outside of the originally receiving thread? If you are actually operating within a web request and still receive this message, your code is probably running outside of DispatcherServlet: In this case, use RequestContextListener or RequestContextFilter to expose the current request.

```
at org.springframework.web.context.request.RequestContextHolder.currentRequestAttributes(RequestContextHolder.java:131)
at org.springframework.web.context.request.SessionScope.get(SessionScope.java:55)
at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:371)
... 129 more
```

Greg: we either need to move the org.slf4j.impl.StaticLoggerBinder in a separate jar and not deploy this in tomcat or use some slf4j config to disable this.

## #2 - 07/19/2023 07:35 AM - Galya B

The problem is not Tomcat, but Spring. I'm not sure how to reproduce it to be able to offer a smarter solution instead of losing some logs.

## #3 - 07/19/2023 07:37 AM - Galya B

How is their Spring logging configured, does it produce separate files?

## #4 - 07/19/2023 07:59 AM - Constantin Asofiei

Galya B wrote:

How is their Spring logging configured, does it produce separate files?

I do not know these details.

But, as this is not a FWD JVM (server or client), I don't think CentralLogger should default (via slf4j) or interfere with the web app logging configuration.

**#5 - 07/19/2023 08:16 AM - Galya B**

Usually when you decide to use a framework, logging is provided by the framework. Here we have Spring framework in FWD framework and FWD framework needs to pretend it's invisible. There is something fishy here, but I can't put my finger on it right away. So this spring web app is deployed as .war in deploy/lib and defined in standard/warfile in directory? I'll try to reproduce it.

For the time being easiest would be to remove slf4j logs by separating StaticLoggerBinder, as suggested, but it means 3rd party libs logs will be lost.

**#6 - 07/19/2023 08:28 AM - Constantin Asofiei**

No, the spring web app is running in its own JVM via tomcat - it is not running in the FWD server JVM.

The customer reports that after removing org.slf4j.impl.StaticLoggerBinder from the p2j.jar used in tomcat the issue is gone.

Galya, please create a separate fwd-slf4j.jar with only org.slf4j.impl.StaticLoggerBinder and do exclude this class from p2j.jar. This way we'll have:

- the FWD server will add fwd-slf4j.jar to the classpath (via deploy/lib/) and all will work the same
- when deploying to tomcat, this @fwd-slf4j.jar@ will not be included.

**#7 - 07/19/2023 08:33 AM - Galya B**

- Status changed from New to WIP

- Assignee set to Galya B

The problem is not Tomcat, but something with Spring picking up slf4j impl from the classpath. Can you please share the full stacktrace for the above exception, while I'm working on separating the class?

**#8 - 07/19/2023 08:39 AM - Constantin Asofiei**

Galya B wrote:

The problem is not Tomcat, but something with Spring picking up slf4j impl from the classpath.

They have no control over the classpath, to i.e. move p2j.jar to be the last in the classpath; even so, we should not introduce configuration requirements when p2j.jar is used outside of FWD server or client.

Can you please share the full stacktrace for the above exception, while I'm working on separating the class?

This is all I have from the customer.

**#9 - 07/19/2023 08:47 AM - Galya B**

When using framework as library inside a container there is a very high chance to get classpath problems. The real solution of this task is not separating slf4j and losing logs. Configurations are a way around bad classpath, when developers don't understand what it is or don't care what they put in it.

Both logs are pretty obscure without the cause of the issue (initial call in the stacktrace). This does not look acceptable to me for accepting an Urgent task.

**#10 - 07/19/2023 08:50 AM - Constantin Asofiei**

Galya B wrote:

The real solution of this task is not separating slf4j and losing logs.

What do you mean by 'losing logs'? Logs from FWD? From somewhere else?

Both logs are pretty obscure without the cause of the issue (initial call in the stacktrace). This does not look acceptable to me for accepting an Urgent task.

The customer is down without a fix, and can't work with the webapp. Please go ahead with the separate jar at this time.

**#11 - 07/19/2023 08:51 AM - Galya B**

Constantin Asofiei wrote:

Galya B wrote:

The real solution of this task is not separating slf4j and losing logs.

What do you mean by 'losing logs'? Logs from FWD? From somewhere else?

From FWD's 3rd party libraries.

Both logs are pretty obscure without the cause of the issue (initial call in the stacktrace). This does not look acceptable to me for accepting an Urgent task.

The customer is down without a fix, and can't work with the webapp. Please go ahead with the separate jar at this time.

I am working on it, as I've said, but the task can't be closed with this "fix".

**#12 - 07/19/2023 08:52 AM - Galya B**

Ask the customer to send the full exceptions, when their stress level is lower and they can manage the mouse to copy paste more context.

**#13 - 07/19/2023 08:53 AM - Constantin Asofiei**

Galya B wrote:

From FWD's 3rd party libraries.

FWD can't be in charge of 3rd party libraries when is deployed in a separate JVM; FWD is in charge of logging only when running as FWD server or FWD client.

**#14 - 07/19/2023 08:55 AM - Galya B**

Constantin Asofiei wrote:

Galya B wrote:

From FWD's 3rd party libraries.

FWD can't be in charge of 3rd party libraries when is deployed in a separate JVM; FWD is in charge of logging only when running as FWD server or FWD client.

FWD consists of 3rd party libraries and quite a few I should say.  
Do you mean when FWD is used as library, there will be no FWD logs at all?

**#15 - 07/19/2023 08:57 AM - Constantin Asofiei**

Galya B wrote:

Do you mean when FWD is used as library, there will be no FWD logs at all?

For FWD, the default is CentralLoggerFallback, which fallback to STDERR - so log message are still emitted. What I mean is that there will be no log files managed by FWD.

**#16 - 07/19/2023 09:28 AM - Galya B**

I'm not sure if the default slf4j impl that logs to stderr slf4j-simple-2.0.7.jar will work with their setup. Anyways, I'll have to work on reproducing it for a long-term solution.

7589a r14664 ready for review.

**#17 - 07/19/2023 09:36 AM - Constantin Asofiei**

Roger, please take a look at 7589a.

Galya, I think the jar needs to be added to this, too:

```
<zip destfile="${archive.convert_destfile}" >
```

**#18 - 07/19/2023 09:41 AM - Galya B**

I checked it, it's fine. This covers it:

```
<zipfileset dir="${basedir}/build/lib"
            prefix="build/lib"
            includes="*" />
```

I've also tested the deployment with the separate jar and slf4j impl is picked up by 3rd party libs.

**#19 - 07/19/2023 09:43 AM - Constantin Asofiei**

I've done a ./gradlew all and it fails with this:

```
FAILURE: Build failed with an exception.
```

```
* What went wrong:
```

```
Execution failed for task ':ant-jar'.
```

```
> Manifest file: ~/branches/7589a/build/manifest/fwdsSlf4j.mf does not exist.
```

```
* Try:
```

```
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output.
```

```
Run with --scan to get full insights.
```

\* Get more help at <https://help.gradle.org>

Deprecated Gradle features were used in this build, making it incompatible with Gradle 6.0.  
Use '--warning-mode all' to show the individual deprecation warnings.  
See [https://docs.gradle.org/5.6.4/userguide/command\\_line\\_interface.html#sec:command\\_line\\_warnings](https://docs.gradle.org/5.6.4/userguide/command_line_interface.html#sec:command_line_warnings)

BUILD FAILED in 1m 3s  
36 actionable tasks: 36 executed

**#20 - 07/19/2023 09:45 AM - Galya B**

Ops, I forgot b2r add again, up in r14665.

**#21 - 07/19/2023 10:08 AM - Roger Borrello**

Constantin Asofiei wrote:

Roger, please take a look at 7589a.

What should I take a look at? You mean with my customer's app?

**#22 - 07/19/2023 10:09 AM - Constantin Asofiei**

Roger Borrello wrote:

Constantin Asofiei wrote:

Roger, please take a look at 7589a.

What should I take a look at? You mean with my customer's app?

I know you've made edits to build.xml and build.gradle lately, if the change makes sense in this regard.

**#23 - 07/19/2023 10:44 AM - Greg Shah**

When used in this "separate Tomcat JVM" scenario, we should to implement a fwd-appserver-client.jar which only includes enough of FWD to replace the 4GL "Open Java Client". Customers should not have to load the full p2j.jar for this use case. This work will be done in a different task.

**#24 - 07/19/2023 10:47 AM - Roger Borrello**

Constantin Asofiei wrote:

I know you've made edits to build.xml and build.gradle lately, if the change makes sense in this regard.

Looks fine.

The new jar will be included in the archive, which is unzipped in its entirety by the deploy\_fwd.sh script. That script is used in my docker build scripts. All good!

**#25 - 07/19/2023 10:53 AM - Constantin Asofiei**

- Priority changed from Urgent to Normal

Roger, thanks for checking.

Galya: I've committed this diff from 7589a to 7156b for the customer. I'll let you know when/if they can give us the entire tomcat log.

**#26 - 07/20/2023 07:08 AM - Constantin Asofiei**

The customer's log has this:

```
SLF4J: Class path contains multiple SLF4J bindings.  
SLF4J: Found binding in [jar:file:/home/app/projects/webapp/app-war/target/app/WEB-INF/lib/fwd-p2j-jar-4.0.0.1  
4663.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: Found binding in [jar:file:/home/app/projects/webapp/app-war/target/app/WEB-INF/lib/logback-classic-1.2  
.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]  
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.  
SLF4J: Actual binding is of type [com.goldencode.p2j.util.logging.CentralSlf4jLoggerFactory]
```

Further messages indicate that the webapp uses logback-classic.

**#27 - 07/20/2023 07:14 AM - Galya B**

According to the logs customer's app uses the first implementation it finds, i.e. FWD's, that's why we see Spring trying to use CentralSlf4jLogger in the stacktrace, while logback-classic comes from Spring. Supposedly now that our StaticLoggerBinder is removed, FWD's 3rd party libs will log using



logback-classic.

**#28 - 08/25/2023 07:45 AM - Galya B**

- Status changed from WIP to Review

- % Done changed from 0 to 100

This should be considered resolved. Allowing the customer to include in the classpath their slf4j implementation and removing our own is a good solution.

**#29 - 08/25/2023 08:33 AM - Greg Shah**

Don't we need to do something with 7589a?

**#30 - 08/25/2023 08:37 AM - Galya B**

Right. It has to be merged. It separates slf4j implementation in its own jar, but builds in the same way. Can be removed on deploy.

**#31 - 08/25/2023 08:48 AM - Greg Shah**

Hynek/Constantin: Any objection to the merge?

Galya: What documentation do we need to explain this?

**#32 - 08/25/2023 08:50 AM - Galya B**

There is no change for customers who don't experience problems with their own Java logging. If they have their own Java logging (like Spring does) and complain about it, they will have to just remove the jar from the lib folder. It's very customer specific. Not sure how it can be documented.

**#33 - 08/25/2023 08:56 AM - Greg Shah**

There is no change for customers who don't experience problems with their own Java logging. If they have their own Java logging (like Spring does) and complain about it, they will have to just remove the jar from the lib folder. It's very customer specific. Not sure how it can be documented.

I think in our logging-related pages we need to highlight an example of the problem that can occur and explain the resolution.

**#34 - 08/25/2023 08:59 AM - Greg Shah**

Since we add our own slf4j override implementation, it can affect many customers. This would be unexpected. Adding this to the documentation will at least allow customers to help themselves instead of calling Galya. ;)

**#35 - 08/25/2023 08:59 AM - Galya B**

Yep, yep, I forgot we have relevant documentation... Working on it.

**#36 - 08/25/2023 09:24 AM - Galya B**

Check the new Logging section [Interference with customer's Java app logging](#).

**#37 - 08/25/2023 09:36 AM - Greg Shah**

Check the new Logging section [Interference with customer's Java app logging](#).

It is a good result. I've made some small edits. Please confirm they are OK.

**#38 - 08/25/2023 09:38 AM - Galya B**

Thanks! I was struggling badly with that possessive 's :) It's good.

**#39 - 08/25/2023 12:49 PM - Hynek Cihlar**

Greg Shah wrote:

Hynek/Constantin: Any objection to the merge?

No objection here.

**#40 - 08/25/2023 01:20 PM - Constantin Asofiei**

Hynek Cihlar wrote:

Greg Shah wrote:

Hynek/Constantin: Any objection to the merge?

No objection here.

I'm good too.

**#41 - 08/26/2023 06:25 AM - Greg Shah**

You can merge now.

**#42 - 08/28/2023 01:58 AM - Galya B**

- Status changed from Review to Test

Task branch 7589a was merged to trunk as rev 14710 and archived.

**#43 - 11/29/2023 03:57 AM - Galya B**

- File *fix-spawner-classpath.diff* added

The JVM spawned from `/opt/spawner/spawn` now doesn't have our slf4j implementation and doesn't produce logs for `CentralLogger.logStream`. I think it should be fine if it's included in the classpath for the spawner only, because there is no Spring and dependency injection involved. Patch attached.

**#44 - 11/30/2023 01:53 PM - Greg Shah**

Code Review *fix-spawner-classpath.diff*

I'm fine with the changes.

How have they been tested?

**#45 - 12/13/2023 05:37 AM - Galya B**

7589b r14868: adds FWD's slf4j implementation to the classpath of both the temp JVM created by the spawner and the spawned client.

`execvp` previously was showing a warning that slf4j implementation is not found. Now this is fixed.

`fwd-slf4j.jar`  is added to the client's classpath in `ClientBuilderOptions`, being the default for directory's config `classpath-extra-jars` and thus can be replaced easily with another slf4j implementation.

If the slf4j library file is missing from the path, the client still spawns without issues. When it's available, the logs will have the record `INFO | org.eclipse.jetty.util.log | PID:136019, ThreadName:main | Logging initialized @800ms to org.eclipse.jetty.util.log.Slf4jLog`.

**#46 - 12/14/2023 02:34 AM - Galya B**

This is the type of warnings displayed when no slf4j imp present for the client:

```
23/12/14 09:28:47.867+0200 | WARNING | Spawner-2 | ThreadName:Spawner-2-Logger | SLF4J: Failed to load class "
org.slf4j.impl.StaticLoggerBinder".
23/12/14 09:28:47.867+0200 | WARNING | Spawner-2 | ThreadName:Spawner-2-Logger | SLF4J: Defaulting to no-opera
tion (NOP) logger implementation
23/12/14 09:28:47.867+0200 | WARNING | Spawner-2 | ThreadName:Spawner-2-Logger | SLF4J: See http://www.slf4j.o
rg/codes.html#StaticLoggerBinder for further details.
```

**#47 - 12/22/2023 12:49 PM - Greg Shah**

- Status changed from Test to Internal Test

Code Review Task Branch 7589b Revision 14868

The changes look good. If suitable testing has been completed, we can get this merged.

**#48 - 01/01/2024 02:55 AM - Galya B**

Greg Shah wrote:

Code Review Task Branch 7589b Revision 14868

The changes look good. If suitable testing has been completed, we can get this merged.

I've tested both clients (temp and regular) with and without the .jar in the classpath and I think it can be merged.

**#49 - 01/02/2024 07:55 AM - Greg Shah**

You can merge now.

**#50 - 01/02/2024 08:12 AM - Galya B**

- Status changed from Internal Test to Test

7589b was merged to trunk as rev. 14908 and archived.

**#51 - 01/02/2024 01:54 PM - Roger Borrello**

Happy New Year!

I just wanted to post a "public service announcement" as this update to postbuild.sh creates a minor issue with Docker projects. These are projects that use postbuild.sh to position the spawner in the Docker images, as well as the FWD Docker images, which include as much of the FWD aspects of the /opt/spawner as possible. We don't want to assume a build environment is available, so that is the background for why the postbuild.sh was updated in the first place.

To illustrate, the FWD Docker file has:

```
# Step 9/11
RUN chown -R fwd:fwd /opt/fwd-deploy && \
    mkdir -p /opt/spawner && \
    cp /opt/fwd-deploy/spawner/spawn /opt/fwd-deploy/spawner/p2j.jar /opt/spawner/ && \
    chmod -R gou+rx /opt/spawner/ && \
    chmod u+s /opt/spawner/spawn && \
    chmod 0550 /opt/spawner/p2j.jar
```

which I will be changing to:

```
# Step 9/11
RUN chown -R fwd:fwd /opt/fwd-deploy && \
    mkdir -p /opt/spawner && \
    cp /opt/fwd-deploy/spawner/spawn /opt/fwd-deploy/spawner/*.jar /opt/spawner/ && \
    chmod -R gou+rx /opt/spawner/ && \
    chmod u+s /opt/spawner/spawn && \
    chmod 0550 /opt/spawner/*.jar
```

Then, project Docker files like  `fwd_4.0_ubuntu_22.04_hotel_gui_server_Dockerfile.template`  in the **hotel\_gui** project will have:

```
/opt/fwd-deploy/spawner/postbuild.sh /opt/spawner /opt/fwd-deploy/spawner/spawn /tmp/${PROJECT}/deploy/server/
srv-certs.store /opt/fwd/build/lib/p2j.jar
```

which I will be changing to:

```
/opt/fwd-deploy/spawner/postbuild.sh /opt/spawner /opt/fwd-deploy/spawner/spawn /tmp/${PROJECT}/deploy/server/
srv-certs.store /opt/fwd/build/lib/p2j.jar /opt/fwd/build/lib/slf4j-impl.jar
```

The default for `SLF4J_JAR` (`../lib/fwd-slf4j.jar`) won't be found unless the current directory is `./p2j/src/native`, which is the whole reason to have updated `postbuild.sh` in the first place, so that relative pathing that assumes a build environment can be overridden. I'll be updating **hotel\_gui** and the customer projects I am maintaining.

The often used `install_spawner.sh` scripts will mostly be OK, since typically it is used with a build environment in place. The updated `install_spawner.sh` is a little more complicated in that it is aware of, and uses the `FWD_LIB` environment variable to find the FWD jars if it is set. I'll be updating it as well, to pass in the `SLF4J_JAR` explicitly to `postbuild.sh`. I am also correcting some of the default behavior of `install_spawner.sh` so it still works as it had in the past. But we really do want to avoid forcing a build environment to be available when creating the spawner directory. But it should behave properly if there is a build environment, which I will be correcting in **hotel\_gui**.

There were also a couple of oversights with this branch that was merged to trunk. The first is that there isn't a failure check if `SLF4J_IMPL` doesn't exist:

```
=== modified file 'src/native/postbuild.sh'
--- old/src/native/postbuild.sh      2023-12-13 10:14:46 +0000
+++ new/src/native/postbuild.sh      2024-01-02 17:51:38 +0000
@@ -133,6 +133,11 @@
     exit 1
 fi

+if [ ! -f $SLF4J_IMPL ]; then
+  echo "$SLF4J_IMPL does not exist!"
+  exit 1
+fi
+
+if [ $# -gt 2 ]; then
+  if [ ! -f "$cert_file" ]; then
+    echo "$cert_file does not exist!"
```

The other is really something that was missing before, but should be addressed, and that is the inclusion of the necessary jar files in the **native** archive:

```
=== modified file 'build.xml'
--- old/build.xml      2024-01-02 13:01:11 +0000
+++ new/build.xml      2024-01-02 17:49:45 +0000
@@ -1134,9 +1134,13 @@
     <property name="archive.native_destfile"
               value="${dist.home}/fwd_deploy-native_${major}.${minor}.${release}_${repo}_${branch}_${rev}_${today_date}.zip" />
     <zip destfile="${archive.native_destfile}" >
+      <zipfileset dir="${basedir}/build/lib"
+        prefix="lib"
+        includes="p2j.jar
+          fwd-slf4j.jar
+          ${libname}" />
     <zipfileset dir="${basedir}/build/native"
       includes="spawn*
         ${libname}
         ${launcher}" />
   </zip>
```

```
<chmod file="${archive.native_destfile}" perm="755"/>
```

The inclusion of all the FWD jars in the native archive does make the use of `install_spawner.sh` mostly moot, as at that point only the `srv-certs.store` file is missing from the spawner directory. I'm not sure how/where to address that, but right now the only place this affects is in the creation of Docker images. Some projects (mostly ones I have touched) do take advantage of the deployment of FWD for devops, and I honestly don't know if all the projects are implemented in the same manner.

#### #52 - 01/03/2024 02:25 AM - Galya B

Roger Borrello wrote:

Happy New Year!

Happy New Year!

There were also a couple of oversights with this branch that was merged to trunk. The first is that there isn't a failure check if `SLF4J_IMPL` doesn't exist:

`SLF4J_IMPL` is not critical, but recommended for the execution of the clients. Adding a failure is not necessary. The customer's case presented in this task is an example, where the `slf4j-impl` won't be provided in a jar.

The other is really something that was missing before, but should be addressed, and that is the inclusion of the necessary jar files in the **native** archive:  
[...]

The inclusion of all the FWD jars in the native archive does make the use of `install_spawner.sh` mostly moot, as at that point only the `srv-certs.store` file is missing from the spawner directory. I'm not sure how/where to address that, but right now the only place this affects is in the creation of Docker images.

If you want to introduce a related change, please take into consideration that with 7589b `postbuild.sh` renames `fwd-slf4j.jar` to `slf4j-impl.jar`, when the lib is placed in the spawner dir and this is the name `spawn.c` and `winspawn.c` are looking for. The reason is that a different (non-FWD) `slf4j` implementation can be passed in to the script.

**#53 - 01/03/2024 08:17 AM - Roger Borrello**

Galya B wrote:

If you want to introduce a related change, please take into consideration that with 7589b postbuild.sh renames fwd-slf4j.jar to slf4j-impl.jar, when the lib is placed in the spawner dir and this is the name spawn.c and winspawn.c are looking for. The reason is that a different (non-FWD) slf4j implementation can be passed in to the script.

Ah, thank you for the explanation. So if there is **not** a slf4j-impl.jar in place, things will still work fine, but the logging will be sub-optimal?

The reason I am pressing this is that use of relative pathing for the default (../lib/) is problematic in most of my use cases.

**#54 - 01/03/2024 08:24 AM - Galya B**

Roger Borrello wrote:

Galya B wrote:

If you want to introduce a related change, please take into consideration that with 7589b postbuild.sh renames fwd-slf4j.jar to slf4j-impl.jar, when the lib is placed in the spawner dir and this is the name spawn.c and winspawn.c are looking for. The reason is that a different (non-FWD) slf4j implementation can be passed in to the script.

Ah, thank you for the explanation. So if there is **not** a slf4j-impl.jar in place, things will still work fine, but the logging will be sub-optimal?

If there is no slf4j implementation lib in the classpath (which can also be set by the web framework in the case of [#7589](#)), some 3rd party libraries (most notably Jetty) won't output logs. It's not a major issue, but it's preferable to have it configured. Because of [#7589](#) we had move the slf4j impl as optional / separate jar.

The reason I am pressing this is that use of relative pathing for the default (../lib/) is problematic in most of my use cases.

I see. I'm also not aware of all deployment configs and I'm not sure what else needs to be done.

## #55 - 01/03/2024 01:29 PM - Roger Borrello

Thanks for the explanation. In order to achieve a level of flexibility in the Docker versions of projects, the Dockerfiles will need to have an ENV SLF4J\_IMPL /opt/spawner/fwd-slf4j.jar in it as the default. This requires a pre-requisite of /opt/spawner/fwd-slf4j.jar being in place, which I can achieve with an update to build.xml as noted in [#7589-51](#). I would also propose this update to postbuild.sh just to prevent any unnecessary error messages:

```
@@ -146,7 +146,7 @@
```

```
cp -f $spawn_prog ${spawn_dir}/spawn
cp -f $P2J_JAR ${spawn_dir}/p2j.jar
-cp -f $SLF4J_IMPL ${spawn_dir}/slf4j-impl.jar
+[ -f "$SLF4J_IMPL" ] && cp -f $SLF4J_IMPL ${spawn_dir}/slf4j-impl.jar

chown root:root ${spawn_dir}/spawn ${spawn_dir}/p2j.jar
chown root:root ${spawn_dir}/spawn ${spawn_dir}/slf4j-impl.jar
```

In the Docker file:

```
ENV SLF4J_IMPL /opt/spawner/fwd-slf4j.jar
...
# Step 10/17
RUN unzip /tmp/deploy.zip -d /tmp/${PROJECT} && \
    /opt/fwd-deploy/spawner/postbuild.sh /opt/spawner /opt/fwd-deploy/spawner/spawn /tmp/${PROJECT}/deploy/ser
ver/srv-certs.store /opt/fwd/build/lib/p2j.jar ${SLF4J_IMPL}
```

And the command that builds the image would contain --build-arg SLF4J\_IMPL=<location of slf4j-impl.jar in image>.

Unfortunately the only example we have is `hotel_gui`, which I don't believe uses anything other than `fwd-slf4j.jar`. If there's an example of something other than `fwd-slf4j.jar`, we'd have the Dockerfile doing something like copying the implementation from the context directory to the image, and that location would be the SLF4J\_IMPL value. I can place this as a comment in the `hotel_gui` dockerfile for now.

I want to make sure we have the `fwd-slf4j.jar` available in all cases, and that's why I want to update the `build.xml` to make sure it is in the native archive.

Should I create a 7589b branch to work on this?



**#56 - 01/04/2024 03:36 AM - Galya B**

Roger Borrello wrote:

In order to achieve a level of flexibility in the Docker versions of projects, the Dockerfiles will need to have an ENV SLF4J\_IMPL /opt/spawner/fwd-slf4j.jar in it as the default. This requires a pre-requisite of /opt/spawner/fwd-slf4j.jar being in place, which I can achieve with an update to build.xml as noted in [#7589-51](#).

The Dockerfile accesses /opt/fwd/build/lib/p2j.jar, so SLF4J\_IMPL can default to /opt/fwd/build/lib/fwd-slf4j.jar in the Dockerfile. This way it won't have to be copied to the native archive and the spawn dir won't end up with two .jar files for slf4j.

I would also propose this update to postbuild.sh just to prevent any unnecessary error messages

This is a good addition. You can create a new branch for it or add it to another working branch that will be merged soon. Up to you.

**#57 - 01/04/2024 12:26 PM - Roger Borrello**

Galya B wrote:

Roger Borrello wrote:

In order to achieve a level of flexibility in the Docker versions of projects, the Dockerfiles will need to have an ENV SLF4J\_IMPL /opt/spawner/fwd-slf4j.jar in it as the default. This requires a pre-requisite of /opt/spawner/fwd-slf4j.jar being in place, which I can achieve with an update to build.xml as noted in [#7589-51](#).

The Dockerfile accesses /opt/fwd/build/lib/p2j.jar, so SLF4J\_IMPL can default to /opt/fwd/build/lib/fwd-slf4j.jar in the Dockerfile. This way it won't have to be copied to the native archive and the spawn dir won't end up with two .jar files for slf4j.

The purpose of having the separate **spawner** deployment archive was so that it would contain all files necessary to build a working /opt/spawner (or whatever path is desired) directory, so I'm inclined to keep the FWD implementation in it. I have been placing the deployments under /opt/fwd-deploy/... and then using that location to link up the /opt/fwd to /opt/fwd-deploy/convert and use the /opt/fwd-deploy/spawner as the place from which the postbuild.sh is run. If a customer has an implementation, they can specify it when they build the Docker image, and it won't move the fwd-slf4j.jar in the /opt/spawner directory.

I would also propose this update to postbuild.sh just to prevent any unnecessary error messages

This is a good addition. You can create a new branch for it or add it to another working branch that will be merged soon. Up to you.

I have created **7589b** and updated the build.xml and postbuild.sh in revision **14912**.

**#58 - 01/05/2024 10:41 AM - Roger Borrello**

I tested postbuild.sh with a bogus file, such that I would be able to test my dockerfile's ability to handle a build-arg and create images containing a specified SLF4J jar file.

The dockerfile has:

```
ARG SLF4J_IMPL=/opt/fwd-deploy/spawner/fwd-slf4j.jar
ENV SLF4J_IMPL ${SLF4J_IMPL}
```

This allows --build-arg SLF4J\_IMPL=/path-to/implementation.jar to be specified. I haven't updated the hotel\_gui sample yet, but another project I have added this parameter to the build: --slf4j\_impl=<slf4j-impl> Jarfile for the SLF4J implementation (def=fwd-slf4j.jar) so that it can be passed through to the Docker image creation.

So the default works, utilizing the fwd-slf4j.jar that this branch now includes in the spawner archive, but when I specified something bogus, I did get errors in postbuild.sh due to the "bogus" file not being available. While this wouldn't be normal, because the only time you'd really specify something is if it existed, I wanted to clean up the error message.

The revision 14913 I just checked in shows a more controlled warning message:

```
#11 0.629 Installing the spawn tool /opt/fwd-deploy/spawner/spawn into /opt/spawner/spawn using /tmp/intermoda
l/deploy/server/srv-certs.store as srv-certs.store
#11 0.629 file with p2j=/opt/fwd/build/lib/p2j.jar and slf4j=bogus...
#11 0.789 WARNING! The /opt/spawner/slf4j-impl.jar file was not found.
#11 DONE 0.9s
```

Please review, it would be great to merge this to trunk soon.

**#59 - 01/05/2024 10:50 AM - Greg Shah**

- Status changed from Test to Review

Galya: Please review.

**#60 - 01/05/2024 12:08 PM - Roger Borrello**

Branch 7589b was rebased to trunk\_14912 and is at revision 14914.

**#61 - 01/08/2024 01:34 AM - Galya B**

Roger, did you create 7589b? 7589b is merged. It should have been 7589c.

**#62 - 01/08/2024 11:17 PM - Roger Borrello**

Galya B wrote:

Roger, did you create 7589b? 7589b is merged. It should have been 7589c.

Oops... I've correct it. Branch 7589c is at revision 14914.

**#63 - 01/09/2024 05:56 PM - Roger Borrello**

Branch **7589c** was rebased to trunk\_14915 and is at revision 14916.

**#64 - 01/10/2024 07:20 AM - Galya B**

Two conditions are introduced, that can be merged together: [ -f "\$SLF4J\_IMPL" ] && cp -f \$SLF4J\_IMPL \${spawn\_dir}/slf4j-impl.jar and if [ -f "\${spawn\_dir}/slf4j-impl.jar" ]; then can be simply -f "\$SLF4J\_IMPL" and then the copy can be moved next to the other file operations.

Otherwise it's good.

**#65 - 01/10/2024 08:37 AM - Roger Borrello**

Galya B wrote:

Two conditions are introduced, that can be merged together: [ -f "\$SLF4J\_IMPL" ] && cp -f \$SLF4J\_IMPL \${spawn\_dir}/slf4j-impl.jar and if [ -f "\${spawn\_dir}/slf4j-impl.jar" ]; then can be simply -f "\$SLF4J\_IMPL" and then the copy can be moved next to the other file operations.

Otherwise it's good.

Great feedback... committed revision 14917.

**#66 - 01/10/2024 08:40 AM - Galya B**

Roger Borrello wrote:

Great feedback... committed revision 14917.

Greg, it's good for merge.

**#67 - 01/10/2024 11:02 AM - Greg Shah**

You can merge after 7667b.

**#68 - 01/10/2024 11:07 AM - Roger Borrello**

Greg Shah wrote:

You can merge after 7667b.

I'll rebase and merge as soon as I see it hits.

**#69 - 01/11/2024 09:09 AM - Roger Borrello**

Branch 7589c was rebased to trunk\_14916 and is at revision 14918.

Branch 7589c was merged to trunk as revision 14917 and archived.

**#70 - 01/11/2024 09:12 AM - Greg Shah**

- *Status changed from Review to Test*

**#71 - 05/02/2024 07:52 AM - Galya B**

This can be closed. We've delivered a way to disable the FWD's slf4j impl what was the issue.

**#72 - 05/02/2024 07:55 AM - Greg Shah**

- *Status changed from Test to Closed*

**Files**

---

|                            |         |            |         |
|----------------------------|---------|------------|---------|
| fix-spawner-classpath.diff | 2.48 KB | 11/29/2023 | Galya B |
|----------------------------|---------|------------|---------|