Patching SHRINE 1.22.6 to 1.22.8

These are instructions for updating SHRINE to version 1.22.8, assuming a running version of 1.22.6.

Shut Down SHRINE

Before starting the upgrade process, make sure SHRINE's Tomcat is not running. Leaving it running during this process can cause problems, especially with unpacking new .war files. Simply run the following command:

```
$ /opt/shrine/tomcat/bin/shutdown.sh
```

Create Backups

Now that SHRINE is stopped, it is a good idea to back up the current versions of the components we will be upgrading. The exact method for making this backups may vary, but these instructions will place the backups in a folder called /opt/shrine/upgrade-backups.

Start by creating a folder to contain these backups:

```
$ mkdir /opt/shrine/upgrade-backups
```

Make especially sure that the **shrine-webclient/** folder is backed up. Later on, we will be restoring important webclient configuration files from this backup. If you choose not to make any backups, make sure to at least keep a copy of **i2b2_config_data.js and js-i2b2/cells/SHRINE/cell_config_data.js**!

```
$ mv /opt/shrine/tomcat/webapps/shrine-webclient /opt/shrine/upgrade-backups/shrine-webclient
```

Make especially sure that the shrine.keystore is backed up. If you lose the private side of a cert you may not be able to recover it.

```
$ cp /opt/shrine/shrine.keystore /opt/shrine/upgrade-backups/shrine.keystore
```

Next, move the current SHRINE webapp folder to the backup location:

```
$ mv /opt/shrine/tomcat/webapps/shrine /opt/shrine/upgrade-backups/shrine
```

Make sure to also back up the other existing SHRINE components (shrine-proxy and steward), just in case:

```
$ mv /opt/shrine/tomcat/webapps/shrine-proxy /opt/shrine/upgrade-backups/shrine-proxy
$ mv /opt/shrine/tomcat/webapps/steward /opt/shrine/upgrade-backups/steward
```

Next, you want to backup shrine.xml and steward.xml to the backup folder (remember to restore these files after the upgrade):

```
$ cp /opt/shrine/tomcat/conf/Catalina/localhost/shrine.xml /opt/shrine/upgrade-backups/shrine.xml
$ cp /opt/shrine/tomcat/conf/Catalina/localhost/steward.xml /opt/shrine/upgrade-backups/steward.xml
```

Finally remove the old .war files with this command:

```
$ rm /opt/shrine/tomcat/webapps/*.war
```

Deploy New .war Files

For the 1.22.8 patch, you will need to deploy all new .war files for each application:

shrine.war

Next, we will retrieve the new SHRINE webapp from the HMS Sonatype Nexus server at: https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-war/1.22.8/. From there, download **shrine-war-1.22.8.war** to the **webapps/** directory on the SHRINE server and rename it to **shrine.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-war/1.22.8/shrine-war-1.22.8.war -O shrine.war
```

steward.war

Much like shrine.war, the SHRINE Data Steward can be found on the HMS Sonatype Nexus server at https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/steward/1.22.8/. From there, download **steward-1.22.8.war** to the **webapps/** directory on the SHRINE server and rename it to **steward.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/steward/1.22.8/steward-1.22.8.war -O steward.war
```

shrine-proxy.war

Like other SHRINE artifacts, the SHRINE proxy can be found on the HMS Sonatype Nexus server at https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-proxy/1.22.8/. From there, download **shrine-proxy-1.22.8.war** to the **webapps/** directory on the SHRINE server and rename it to **shrine-proxy.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-proxy/1.22.8/shrine-proxy-1.22.8.war -O shrine-proxy.war
```

SHRINE Webclient

The SHRINE webclient can be found on the HMS Sonatype Nexus server at https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-webclient/1.22.8/. From there, download **shrine-webclient-1.22.8-dist.zip** file to the **webapps/** directory on the SHRINE server and rename it to **shrine-webclient.zip**. Then, unzip the shrine-webclient.zip file.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-webclient/1.22.8
/shrine-webclient-1.22.8-dist.zip -O shrine-webclient.zip
$ unzip shrine-webclient.zip
```

Restore Webclient Backups

After this, restore the previous **i2b2_config_data.js** and **cell_config_data.js** files from your backup and place them in the new shrine-webclient folder:

```
$ cp /opt/shrine/upgrade-backups/shrine-webclient/i2b2_config_data.js /opt/shrine/tomcat/webapps/shrine-
webclient/i2b2_config_data.js
$ cp /opt/shrine/upgrade-backups/shrine-webclient/js-i2b2/cells/SHRINE/cell_config_data.js /opt/shrine/tomcat
/webapps/shrine-webclient/js-i2b2/cells/SHRINE/cell_config_data.js
```

SHRINE Dashboard

Like other SHRINE artifacts, the SHRINE Dashboard can be found on the HMS Sonatype Nexus server at https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/dashboard-war/1.22.8/. From there, download **dashboard-war-1.22.8.war** to the **webapps/** directory on the SHRINE server and rename it to **shrine-dashboard.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/dashboard-war/1.22.8
/dashboard-war-1.22.8.war -O shrine-dashboard.war
```

SHRINE Node Metadata Service

Like other SHRINE artifacts, the SHRINE Node Metadata Service can be found on the HMS Sonatype Nexus server at https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/meta-war/1.22.8/. From there, download **meta-war-1.22.8.war** to the **webapps/** directory on the SHRINE server and rename it to **shrine-meta.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/meta-war/1.22.8/meta-war-
1.22.8.war -O shrine-meta.war
```

Consolidate Shrine.xml and Steward.xml into Context.xml

With SHRINE 1.22.8, you can consolidate the contents of shrine.xml and steward.xml into context.xml, located in /opt/shrine/tomcat/conf/context.xml.

The final context.xml should look like this:

context.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!-- The contents of this file will be loaded for each web application -->

<Context>
        <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <Resource name="jdbc/problemDB" auth="Container" type="javax.sql.DataSource"
              maxActive="100" maxIdle="30" maxWait="10000"
              username="shrine" password="demouser" driverClassName="com.mysql.jdbc.Driver"
              url="jdbc:mysql://localhost:3306/shrine_query_history"
              testOnBorrow="true" validationQuery="SELECT 1"/>

    <Resource name="jdbc/shrineDB" auth="Container" type="javax.sql.DataSource"
               maxActive="100" maxIdle="30" maxWait="10000"
               username="shrine" password="demouser" driverClassName="com.mysql.jdbc.Driver"
               url="jdbc:mysql://localhost:3306/shrine_query_history"
               testOnBorrow="true" validationQuery="SELECT 1"/>

    <Resource name="jdbc/adapterAuditDB" auth="Container" type="javax.sql.DataSource"
              maxActive="100" maxIdle="30" maxWait="10000"
              username="shrine" password="demouser" driverClassName="com.mysql.jdbc.Driver"
              url="jdbc:mysql://localhost:3306/adapterAuditDB"
              testOnBorrow="true" validationQuery="SELECT 1"/>

    <Resource name="jdbc/qepAuditDB" auth="Container" type="javax.sql.DataSource"
              maxActive="100" maxIdle="30" maxWait="10000"
              username="shrine" password="demouser" driverClassName="com.mysql.jdbc.Driver"
              url="jdbc:mysql://localhost:3306/qepAuditDB"
              testOnBorrow="true" validationQuery="SELECT 1"/>

    <Resource name="jdbc/stewardDB" auth="Container" type="javax.sql.DataSource"
              maxActive="100" maxIdle="30" maxWait="10000"
              username="shrine" password="demouser" driverClassName="com.mysql.jdbc.Driver"
              url="jdbc:mysql://localhost:3306/stewardDB"
              testOnBorrow="true" validationQuery="SELECT 1"/>
</Context>
```

Delete Catalina.out / Rotate Out Log

The traditional way to do this is to:

```
$ rm catalina.out
```

This will clear the log file, and will not disrupt the processes that currently hold open file handles.

Another way is to not lose your logging information, by rotating out the log file. To do this, create or edit the file **`/etc/logrotate.d/tomcat`** and have its contents read:

```
$ /var/log/tomcat/catalina.out { copytruncate daily rotate 7 compress missingok size 5M }
```

Then restart logrotate with the command (as root):

```
$ /usr/sbin/logrotate /etc/logrotate.conf
```

And you should have the log file rotated out daily, or if the size exceeds 5M, with the last seven logs kept for debugging purposes.

Change Passwords for all PM user accounts

As a recommendation due to the security changes in the 1.22.8 patch, please change all the passwords for users in the PM cell database.

You will need to sign into the i2b2 instance as an administrator and manually change each user's password.

Start SHRINE

The only thing left to do at this point is start SHRINE back up. Simply do the following:

```
$ /opt/shrine/tomcat/bin/startup.sh
```

Verify SHRINE Upgrade

After starting SHRINE up, verify that the upgrade was properly deployed by checking the SHRINE Dashboard. The exact address you will need to go to depends on your configuration, but the general format looks like the following:

```
https://your.shrine.host:6443/shrine-dashboard
```