Network-Specific: HMS to 1.21.2

In most cases, upgrading an existing instance of SHRINE is a relatively quick process. Exceptions to this rule include older versions of SHRINE that contained substantial changes to configuration files and other portions of the file structure. The instructions here specifically describe an upgrade path from SHRINE 1.20.1 to SHRINE 1.21.2. For instructions on upgrading to SHRINE 1.20.1, check the child pages underneath the "Archived Upgrade Instructions" section.

This guide makes the following assumptions of a current 1.20.1 user. Make sure all of these conditions are satisfied before proceeding:

- i2b2 1.7.05 or newer is installed and operational.
- SHRINE 1.20.1 is installed and operational. (along with Tomcat 7)

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. Simply run the following command:

```
$ shrine_shutdown
```

If the above command is not found, try the following instead:

```
$ /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
```

Remove the old .war files with this command:

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

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 other existing SHRINE components (shrine-proxy), just in case:

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

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
```

Deploy New shrine.war

Next, we will retrieve the new SHRINE webapp from the HMS Sonatype Nexus server at http://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-war/. Navigate to the folder for 1.21.2. From there, download **shrine-war-1.21.2.war** to the **webapps/** directory of the Tomcat installation that SHRINE uses and rename it to **shrine.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget --no-check-certificate https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-war
/1.21.2/shrine-war-1.21.2.war -O shrine.war
```

Deploy New 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/. Navigate to the folder for 1.21.2. From there, download **shrine-proxy-1.21.2.war** to the **webapps/** directory of the Tomcat installation that SHRINE uses and rename it to **shrine-proxy.war.**

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget --no-check-certificate https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-
proxy/1.21.2/shrine-proxy-1.21.2.war -O shrine-proxy.war
```

Deploy New SHRINE Webclient

NOTE: Installation procedures for this component have changed after 1.19.x! Please be careful. Starting with 1.20.x, the SHRINE webclient is packaged as a zip file, and no longer requires a checkout from the old SHRINE SVN repository.

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/. Navigate to the folder for 1.21.2. From there, download **shrine-webclient-1.21.2-dist.zip** to the **webapps/** directory of the Tomcat installation that SHRINE uses and rename it to **shrine-webclient.zip**. Then, unzip the **shrine-webclient.zip** file.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget --no-check-certificate https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/shrine-
webclient/1.21.2/shrine-webclient-1.21.2-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
```

Deploy New SHRINE Dashboard

NOTE: This component is a working prototype, and simply mirrors information from the existing SHRINE Happy module in its current state. Additional functionality is coming in a future release.

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/. Navigate to the folder for 1.21.2. From there, download **dashboard-war-1.20.1.war** to the **webapps/** directory of the Tomcat installation that SHRINE uses and rename it to **shrine-dashboard.war**.

For example:

```
$ cd /opt/shrine/tomcat/webapps
$ wget --no-check-certificate https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/dashboard-
war/1.21.2/dashboard-war-1.21.2.war -O shrine-dashboard.war
```

(Optional) For Users of Oracle or SQL Server

If you experience issues in getting the audit logging functionality working with Oracle or SQL Server, please try the instructions in this article written by a SHRINE developer: Using SHRINE Data Steward with Oracle or SQL Server (the instructions within also apply to the audit logging functionality).

This article contains links to alternative database drivers, as well as information on the necessary changes to make to **shrine.xml** and **shrine.conf**.

Due to the new query audit database functionality added in SHRINE 1.20.1, you will also need to modify **shrine.conf**. There are **audit{}** blocks that must be added to both the **adapter{}** and **queryEntryPoint{}** sections of **shrine.conf**. By default, these are configured to use MySQL, so users of MySQL do not need to add anything to their shrine.conf. However, users of Oracle or SQL Server for SHRINE must add them. See the below for an example:

```
shrine {
[...]
  queryEntryPoint {
    audit {
      database {
        slickProfileClassName="freeslick.driverNameHere$"
      }
    }
    [...]
  }
[...]

  adapter {
    audit {
      database {
        slickProfileClassName="freeslick.driverNameHere$"
      }
    }
    [...]
  }
[...]
}
```

Replace "**freeslick.driverNameHere$**" with the name of the driver you are using ("**freeslick.OracleProfile$**" for Oracle, and "**freeslick.MSSQLServerProfile$**" for SQL Server).

Update PRIVILEGED_USER Table Structure

The **PRIVILEGED_USER** table now allows a user's query threshold to be set to NULL to allow it to fall back to the current value in **shrine.conf**. Run the following queries on your SHRINE query history database:

mysql -u shrine -p -D shrine_query_history

```
alter table PRIVILEGED_USER modify column threshold int null;
```

Create Additional QEP Audit Database Tables

Additional tables have been added to the **qepAuditDB** database to support storing query results at the QEP level. Run the following queries on your QEP audit database:

mysql -u shrine -p -D qepAuditDB

```
create table `previousQueries` (`networkId` BIGINT NOT NULL,`userName` TEXT NOT NULL,`domain` TEXT NOT NULL,
`queryName` TEXT NOT NULL,`expression` TEXT,`dateCreated` BIGINT NOT NULL,`deleted` BOOLEAN NOT NULL,`queryXml`
TEXT NOT NULL,`changeDate` BIGINT NOT NULL);
create table `queryFlags` (`networkId` BIGINT NOT NULL,`flagged` BOOLEAN NOT NULL,`flagMessage` TEXT NOT NULL,
`changeDate` BIGINT NOT NULL);
create table `queryResults` (`resultId` BIGINT NOT NULL,`networkQueryId` BIGINT NOT NULL,`instanceId` BIGINT
NOT NULL,`adapterNode` TEXT NOT NULL,`resultType` TEXT,`size` BIGINT NOT NULL,`startDate` BIGINT,`endDate`
BIGINT,`status` TEXT NOT NULL,`statusMessage` TEXT,`changeDate` BIGINT NOT NULL);
create table `queryBreakdownResults` (`networkQueryId` BIGINT NOT NULL,`adapterNode` TEXT NOT NULL,`resultId`
BIGINT NOT NULL,`resultType` TEXT NOT NULL,`dataKey` TEXT NOT NULL,`value` BIGINT NOT NULL,`changeDate` BIGINT
NOT NULL);
create table `queryResultProblemDigests` (`networkQueryId` BIGINT NOT NULL,`adapterNode` TEXT NOT NULL,`codec`
TEXT NOT NULL,`stamp` TEXT NOT NULL,`summary` TEXT NOT NULL,`description` TEXT NOT NULL,`details` TEXT NOT NULL,
`changeDate` BIGINT NOT NULL);
```

These SQL statements are for MySQL. If you are using Oracle or SQL Server, use either **oracle.ddl** or **mssql.ddl** instead: https://open.med.harvard.edu/stash/projects/SHRINE/repos/shrine/browse/qep/service/src/main/sql?at=refs%2Ftags%2F1.21.2

Modify shrine.conf

The default configuration has changed in SHRINE 1.21 to no longer create a hub unless explicitly asked to. This mirrors the most common installation scenario: a member site connecting to an already established hub. However, on the HMS network, every site is its own hub that broadcasts out to all other sites on its own.

Add the following to the **hub{}** block in **shrine.conf**:

```
hub {
  create = true
  [...]
}
```

Migrate Query History to QEP

This step is crucial if you wish to preserve your previous queries!

SHRINE 1.21 no longer stores query results at the Adapter, but instead stores them at the local QEP. This has the additional benefit of speeding up webclient performance, since the webclient will no longer have to wait for responses from every network member before displaying a list of previous queries. Starting from SHRINE 1.21 and onward, any queries executed will be stored in the QEP Audit Database.

A migration tool has been created for users that wish to migrate their previous query history to this new location, called `adapter-queries-to-qep`.

This migration tool can be found on the HMS Sonatype Nexus server at https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/adapter-queries-to-qep/. Navigate to the folder for 1.21.2. From there, download **adapter-queries-to-qep-1.21.2.zip** to the **/opt/shrine/** directory of your SHRINE server and rename it to **adapter-queries-to-qep.zip**. Then, unzip the **adapter-queries-to-qep.zip** file and navigate to the directory it creates.

For example:

```
$ cd /opt/shrine/
$ wget --no-check-certificate https://repo.open.med.harvard.edu/nexus/content/groups/public/net/shrine/adapter-queries-to-qep/1.21.2/adapter-queries-to-qep-1.21.2.zip -O adapter-queries-to-qep.zip
$ unzip adapter-queries-to-qep.zip
$ cd adapter-queries-to-qep-1.21.2/
```

After this, edit **conf/adapter-queries-to-qep.conf**. There are two major blocks to this conf file: **adapter{}** and **queryEntryPoint{}**. **adapter{}** refers to the old **shrine_query_history** data source, which is where old previous query history will be pulled from. **queryEntryPoint{}** refers to the new **qepAuditDB** data source, which will be the migration destination. Both sections have the following configuration values, which mirror attributes in **shrine.xml** for their corresponding **<Resource />** element:

- **driverClassName**: JDBC driver used to connect to the database. Corresponds to the **driverClassName** attribute in **shrine.xml**.
- **url**: JDBC URL used to connect to the database. Corresponds to the **url** attribute in **shrine.xml**.
- **username**: Username used to connect to the database. Corresponds to the **username** attribute in **shrine.xml**.
- **password**: Password used to connect to the database. Corresponds to the **password** attribute in **shrine.xml**.

With the exception of the **password** variable, the default values should all work with a standard MySQL-backed SHRINE installation.

After confirming that the configuration values are correct, make the **adapter-queries-to-qep** tool executable. Then, run the tool, giving the following arguments:

- i2b2 domain used by SHRINE. i2b2 demo installations use **i2b2demo** by default.
- path to config file for the adapter-queries-to-qep tool. This is almost always **conf/adapter-queries-to-qep.conf**
- path to SHRINE config file. This is typically **/opt/shrine/tomcat/lib/shrine.conf**

For example:

```
$ chmod +x adapter-queries-to-qep
$ ./adapter-queries-to-qep i2b2demo conf/adapter-queries-to-qep.conf /opt/shrine/tomcat/lib/shrine.conf
```

Depending on the size of your database, this process may take a few minutes.

Start SHRINE

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

```
$ shrine_startup
```

If the above command is not found, try the following instead:

```
$ /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/client/index.html#/dashboard/diagnostics
```

It may take a few seconds for the page to load, but after it does load, log in with your SHRINE credentials (any user will do, regardless of role) and verify that the value for "SHRINE Version" is 1.21.2. If it is still displaying an old version, repeat the instructions in the "**Deploy New shrine.war**" section, restart SHRINE, and try again.