

XNAT Setup Options - Custom Configuration Settings

In XNAT 1.7, we are trying to make it easier for administrators to configure XNAT via the UI rather than having to modify properties files. However, there are some cases in which configuration must be done via properties file. In order to even access the database in which we store the properties for an XNAT instance, the XNAT instance needs to know how to access that database. These database access settings are stored in the `xnat-conf.properties` file. If there are other properties that you need (or want) to have set before you start configuring XNAT through the UI, these can also be set in properties files.

Setting properties in xnat-conf.properties

In your XNAT webapp directory, there should be a file located at `WEB-INF/conf/xnat-conf.properties`. In the [XNAT source code](#), this file is located at `src/main/webapp/WEB-INF/conf/xnat-conf.properties`. This file should look something like this (with some comments at the top):

```
datasource.driver=org.postgresql.Driver
datasource.url=jdbc:postgresql://localhost/xnat
datasource.username=xnat
datasource.password=xnat

hibernate.dialect=org.hibernate.dialect.PostgreSQL9Dialect
hibernate.hbm2ddl.auto=update
hibernate.show_sql=false
hibernate.cache.use_second_level_cache=true
hibernate.cache.use_query_cache=true
```

The `datasource` properties here should be set to whatever you set them to when installing XNAT. If you used the defaults suggested in the installation instructions, you should not need to modify these. If you used a different database name than XNAT, you should change the `datasource.url` line like so:

```
datasource.url=jdbc:postgresql://localhost/YOUR_DATABASE_NAME
```

You should also change the username and password lines to match the database username and password you used when setting up XNAT. If you set these properties to the values used when creating your empty database, XNAT should correctly populate the database with all the tables you will need when you start Tomcat.

You should not need to modify any of the hibernate properties, but can if you wish. the dialect should be left set to `org.hibernate.dialect.PostgreSQL9Dialect` because XNAT queries are written for PostgreSQL version 9. You will probably want to leave the `hibernate.hbm2ddl.auto` property set to `update` so that the your schemas will update so that your database stays in sync with any code changes. This is especially important when upgrading from an earlier version of XNAT so that the database tables will match what XNAT expects (e.g. adding a column that is new in 1.7), while preserving your existing data. You can also change `hibernate.show_sql` to `true` if you want all SQL statements to be logged to the console. Finally, you can set the cache settings to `false` if you want to disable the second-level cache and query caching. Your application's performance will likely be worse if you set these caching properties to `false`.

Setting Other Properties Files

In addition to the database and hibernate properties, there are a large number of preferences that can be set by going to Administer->Site Administration when logged in to XNAT as an administrator. However, you may want to have your XNAT initialized with some of these preferences already set. You can do this by creating a properties file containing the preferences you want to have set. For most preferences, they would need to go in `site-config.properties`, but preferences regarding email should go in `notifications.properties`.

Setting Site Config Preferences

If you open `org/nrg/xdat/preferences/SiteConfigPreferences.java`, you can see all the different site config preferences that you can set. At the top of that file is where XNAT will look for the properties file for site config preferences:

```
properties = "META-INF/xnat/preferences/site-config.properties"
```

You can also look through `SiteConfigPreferences.java` to see if there are any properties that you would like XNAT to be initialized with. For example, let's say you want to set what the XNAT archive path is initialized to. In `SiteConfigPreferences.java`, you can see that there is a `getArchivePath` method which gets the archive path preference value by its name, "archivePath":

```
@NrgPreference(defaultValue = "/data/xnat/archive")
public String getArchivePath() {
    return getValue("archivePath");
}
```

By default XNAT will be initialized with the `defaultValue` for this preference, `"/data/xnat/archive"`, but you can have XNAT be initialized with a different archive path by creating a `site-config.properties` file which contains a line like this:

```
archivePath=/opt/customarchive
```

This sets the value for the preference `archivePath` to `"/opt/customarchive"`.

Setting Notifications Preferences

If you open `org/nrg/xdat/preferences/NotificationsPreferences.java`, you can see all the different notifications preferences that you can set. At the top of that file is where XNAT will look for the properties file for notifications preferences:

```
properties = "META-INF/xnat/preferences/notifications.properties"
```

You can also look through `NotificationsPreferences.java` to see if there are any properties that you would like XNAT to be initialized with. For example, let's say you want to set whether admins should be notified on new user registration. In `NotificationsPreferences.java`, you can see that there is a `getNotifyAdminUserRegistration` method which gets the notify admin preference value by its name, `"notifyAdminUserRegistration"`:

```
@NrgPreference(defaultValue = "false")
public boolean getNotifyAdminUserRegistration() {
    return getBooleanValue("notifyAdminUserRegistration");
}
```

By default XNAT will be initialized with the `defaultValue` for this preference, `false`, but you can have XNAT be initialized to always notify admins on user registration by creating a `site-config.properties` file which contains a line like this:

```
notifyAdminUserRegistration=true
```

This sets the value for the preference named `"notifyAdminUserRegistration"` to `true`.

Deploying Your Preference Files

Once you have created preference files containing the values you want, you will need to get them into XNAT. In XNAT 1.7, we have changed the build process and now encourage XNAT site managers to create plugins for all their custom files. Once you have created properties files for site config and/or notifications, you will need to either add them to an existing plugin or create a new one. These files should be located at `META-INF/xnat/preferences/` within your plugin. Plugins are simply jars, so if you are creating a new plugin, simply jar the directory which contains your `META-INF/xnat/preferences` files. Once you have these properties files in jars, simply shut down Tomcat, move the plugin jar file into the **plugins** folder (by default this is under the folder configured as `xnat.home`), and restart Tomcat. For more information in how to set up and deploy plugins, check out our page on [developing XNAT plugins](#).