

Database Update Conventions

So you want to make a data model change?

When creating new features in OpenMRS, it is often necessary to change some database columns, add a table, etc. Database changes are done incrementally so that anyone can update their database from any version to any version.

As of version 1.5, we use the [Liquibase](#) library to define and run each incremental changeset. The changeset file is stored in `/api/src/main/resources/liquibase-update-to-latest.xml`.

As each changeset is executed, it is logged in the user's liquibasechangelog table. Future runs of the changelog file will check the liquibasechangelog table first to see if the changeset has been executed. If it has, then the change is skipped.

This changeset file is shipped with the OpenMRS API jar file and so also the OpenMRS war file. When OpenMRS is started and database changes are needed, openmrs enters "maintenance mode" until an administrator logs in and runs the necessary changes.

Workflow for Adding a Change

Step One: Be sure you are running the latest version

- Make sure you are running the most recent version of your branch (aka, trunk, 1.5.x, etc). Someone might have committed other database changes to tables you are changing.
- Deploy this war file to get your (version specific hopefully) database up-to-date and your changes will be going against the latest version.

Step Two: Update the database script file

1. Open the `/api/src/main/resources/liquibase-update-to-latest.xml` file.
2. Add your update to the **end** using liquibase's format for updates. See the [liquibase manual](#) for help.
 - The "id" must be a globally unique string across all changesets. By convention, we use the current date+time along with an optional incrementing suffix if needed – e.g: `id="200804221449-1"`.
 - Use your [OpenMRS ID](#) for the "author" attribute
 - Be sure to add a "comment" element in your changeSet, it is displayed in the webapp for end users to see
 - We recommend including the ticket number in the comment e.g "Add date_changed column to encounter_type table - TRUNK-2345"

Step Three: Deploy OpenMRS to Run the Changes

- When you compile and deploy the war file, openmrs will prompt you to run the database updates. Run them now to make sure they work correctly.

Step Four : Make related code changes

Obviously, all database related Hibernate mapping files must be updated to work with your new latest datamodel change.

Commit both the liquibase xml file and code changes simultaneously to the git hub repository.

Warnings

Never ever change a changeset! (Well, almost never)

Once you commit something to a branch, you should assume that another developer has *already* checked out your code and applied the database changes! Therefore, you should not just go and modify a changeset! Instead you should add another changeset below the previous one that does the necessary modifications. That way no developers have to go and mess with their database to get the changesets to apply.

Repair the changeset by adding a later changeset (with a different "id") is the preferred approach. But if a changeset contains an error that you must correct, follow the instructions on [How to Modify a Bad Changeset](#).

Preconditions are important

You should always use a precondition in your changeset and not solely rely on liquibase comparing the id to the liquibasechangelog table. This is helpful for users that are upgrading from a 1.5.3 to a 1.6.0 because there have been changes (.1, .2, and .3) that happened to both the 1.5.x branch and to trunk.

Small Changesets are good

Each changeset should only be one logical unit. This allows the changeset to have a simple precondition and makes error recovery simple if the server crashes while running an update.

About the liquibase-data.zip file

When a new minor version (1.4-->1.5) is released, the liquibase-data.zip file should be updated. This is the **only time** the files in the /metadata/model /liquibase-data.zip will be updated. **Developers should only worry** about adding their change to the liquibase-update-to-latest.xml file.