Create upgrade rules for: Amster

One requirement after making a schema change is to write Amster upgrade rules. Amster upgrade is manual via the use of the openam-config-upgrader and its rule files. The upgrader's associated rules perform that transformation. This page is here to help you get started with writing Amster upgrade rules for any schema change.

tl;dr

After a schema change add Amster upgrade rules. Use the existing rules as examples.

Amster files and their purpose

Description

AM has the option of storing its configuration in

- LDAP, **or**
- File base

Amster can be used to export the contents of this configuration to a set of files. The exported Amster files require upgrading before import to a newer version of AM.

Rules to upgrade Amster files

Examples

The the released rules and associated test cases are a good set of examples to follow.

Backporting Amster Upgrade Rules

When creating an AM backport that requires Amster rules files it is important to place the new rule in all the relevant upgrade files. Updates to the following files are required:

- The previous major version to current major version file (e.g. `6.5.x-to-7.x.x.groovy`).
- The current minor version to next minor version file (e.g. `7.0.0-to-7.0.1.groovy`, `7.0.1-to-7.0.2.groovy` etc.).

If you are not yet at the current major version (e.g. you are at `6.5.3` and the current version is `7.0.2`) you will only need to run one file (e.g. `6.5.x-to-7.0.x.groovy`) for your upgrade.

If you are at a previous patch of the current major version (e.g. you are at `7.0.0` and the current version is `7.0.2`) then you will need to run each patch upgrade sequentially (e.g. you will need to run `7.0.0-to-7.0.1.groovy`, followed by `7.0.1-to-7.0.2.groovy`) until you reach the current patch version.