Liquibase Database change tracking

CONCEPTS

Liquibase runs changesets (id + author + file) from a changelog once, and records them in the DATABASECHANGELOG table with a checksum.

Never edit a changeset that has already been deployed: its checksum changes and the next run fails. Add a new changeset instead.

PROJECT AND CONNECTION

`liquibase --version`
Show the installed version

`liquibase` *`<command>`* **`--help`**
Show every flag of a command

`--url=`*`<jdbc-url>`*
JDBC connection URL

`--username=`*`<user>`* **`--password=`***`<pw>`*
Database credentials

`--changelog-file=`*`<path>`*
Root changelog (xml, yaml, json or sql)

`--defaults-file=`*`<file>`*
Read settings from another properties file

CHANGELOG FORMATS

`-- liquibase formatted sql`
First line of a SQL changelog

`-- changeset` *`<author>`***`:`***`<id>`*
Start a changeset in a SQL changelog

`-- rollback` *`<sql>`*
Rollback statement for the changeset above

`<include file="<path>"/>`
Pull another changelog into the root file

UPDATE

`liquibase update`
Deploy all pending changesets

UPDATE (cont.)

`liquibase update-sql`
Print the SQL update would run

`liquibase update-count --count=`*`<n>`*
Deploy only the next n changesets

ROLLBACK

`liquibase rollback --tag=`*`<name>`*
Roll back to a tag

`liquibase rollback-count --count=`*`<n>`*
Undo the last n changesets

Rollback does not bring back data lost to DROP, DELETE or TRUNCATE. Preview with rollback-sql first.

INSPECT AND TRACK

`liquibase status`
List changesets not yet deployed

`liquibase history`
List changesets already deployed

`liquibase tag --tag=`*`<name>`*
Mark the current database state

GENERATE AND SYNC

`liquibase generate-changelog --changelog-file=`*`<file>`*
Reverse-engineer an existing database

MAINTENANCE

`liquibase validate`
Find changelog errors before deploying

`liquibase release-locks`
Clear a lock left by a crashed run

CONFIGURATION

`url:` *`<jdbc-url>`*
Connection URL in liquibase.properties

`changelogFile:` *`<path>`*
Root changelog in liquibase.properties