Mike Willbanks

Sr. Software Engineer, CaringBridge Inc.

LiquiBase

Database Change Management

http://www.liquibase.org

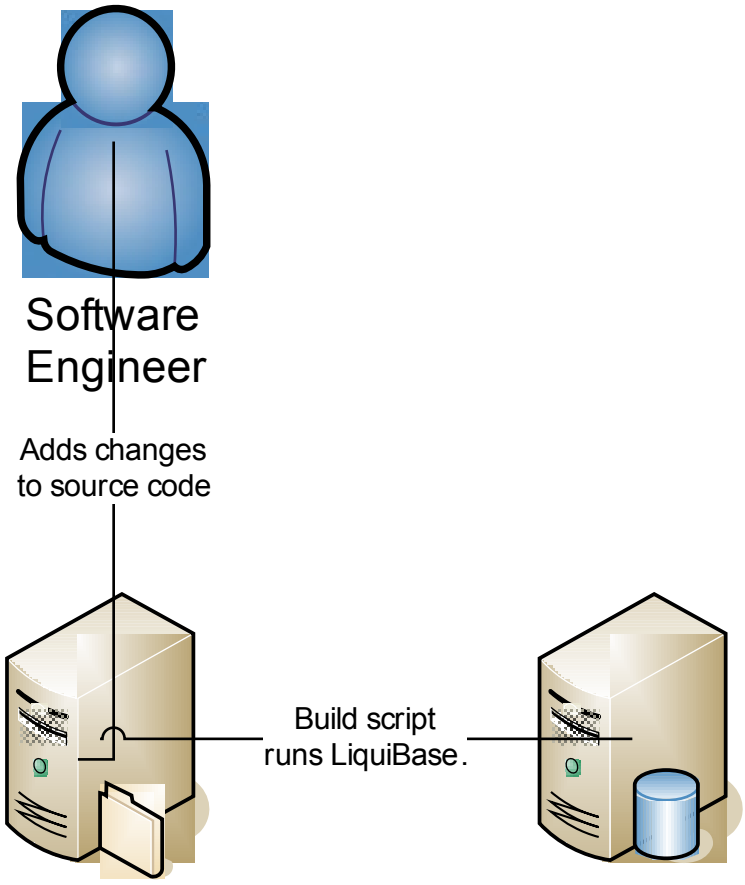
March 30th, 2009

What is LiquiBase

- LiquiBase is an open source, database-independent library for tracking, managing and applying database changes.
- Database changes are stored in an XML file and (optionally) checked into source control.
- LiquiBase executes changes based on this XML file to handle different revisions of database structures and data.

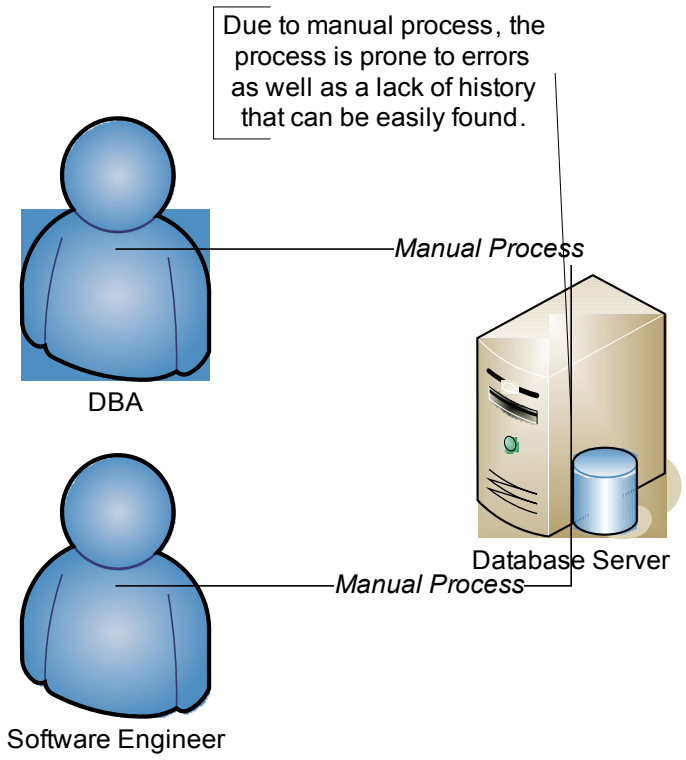
Why LiquiBase?

- Consistent database changes.
- Manage databases at different states.
- Keep a history of changes.
- Automatic rollback support.
- Ability of automation.
- Effectively manage variable change.
- Less human resources / errors.



Problems of Manual Changes

- Inconsistent application of changes.
- Ineffective mechanisms for managing changes.
- Database changes may or may not have been communicated to the team.
- Databases may become out of sync between environments.



The ChangeLog

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.9"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.9
    http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.9.xsd">
    <changeSet id="1" author="mike">
        <createTable tableName="userGroup">
            <column name="id" type="int" autoIncrement="true">
                <constraints primaryKey="true" nullable="false" />
            </column>
            <column name="name" type="varchar(100)" />
        </createTable>
    </changeSet>
</databaseChangeLog>
```

- The changelog is an xml file where all database changes are listed.
- The changelog contains a changeset that lists each individual change.
- The above is an example of creating a table and adding columns.

Generating a ChangeLog

- Getting started, you may want to generate a list of your current database.
- LiquiBase makes this easy by allowing you to run a simple command from the command line client to generate a full changelog.
- Limitations do exist such that it will not export triggers, stored procedures, functions and packages.

```
liquibase --driver=com.mysql.jdbc.Driver \
          --classpath=/path/to/classes \
          --changeLogFile=/path/to/db.changelog.xml \
          --url="jdbc:mysql://hostname/database" \
          --username=dbusername \
          --password=dbpassword \
          generateChangeLog
```

Running a ChangeLog

- Running a changelog is easy, we will focus on the command line client.
- When you first run a changelog, LiquiBase manages those changelogs by adding two tables into your database.
  - databasechangelog: maintains the database changes that were run.
  - databasechangeloglock: ensures that two machines don't attempt to modify the database at one time.

```
liquibase --driver=com.mysql.jdbc.Driver \
          --classpath=/path/to/classes \
          --changeLogFile=/path/to/db.changelog.xml \
          --url="jdbc:mysql://hostname/database" \
          --username=dbusername \
          --password=dbpassword \
          migrate
```

LiquiBase Functionality

- LiquiBase refactoring functionality:
  - Structural
    - Columns, Tables, Views, Stored Procedures
  - Data Quality
    - Lookup Tables, Constraints, Sequences, Defaults.
  - Referential Integrity
    - Foreign Keys, Primary Keys
  - Transformations
    - Insert, Update, Delete, Tag, Stop
  - Architectual
    - Indexes
  - Custom
    - Custom SQL, Modify SQL, Execution

An Example

- We have a database "meetup", we want to have two tables:
  - meetup: contains an id, name, description and the date it was created and last updated.
  - event: contains an id, a meetup id, event name, event description and event date.

Why LiquiBase over Alternatives

- XML file makes it easier to read changes and see a history of changes.
- Support for multiple ChangeLogs
  - Example: Pre-Deployment, Post-Deployment.
- Run through automated systems.
- Heavily documented.
- Project activity.

Questions?