

Concept Dataset Generation Maven Plugin

Primary mentor	Rowan Seymour
Backup mentor	TBD
Assigned to	TBD

Background

It is a challenge for developers, especially those working on large distributions, to maintain test data which is consistent with their production environments. For example, developers may want to write unit tests for an indicator report that includes most of the metadata and concepts used by the distribution.

- [Metadata Deploy](#) module addresses this challenge for a lot of metadata types - but doesn't include concepts, as these don't lend themselves to being described in code. Also distributions commonly deploy concepts via a SQL dump (e.g. CIEL)
- PIH team have written code to generate a MariaDB copy of a production database but there's no easy way to specify what content should be included.

This project would seek to provide a mechanism by which developers can easily generate a test dataset file for use in unit tests, which includes a configurable subset of their entire concept dictionary. It's important that developers can configure what is included in the final dataset file so that they can keep it as small as possible to keep unit test times as fast as possible - it's obviously not feasible to load an entire dictionary like CIEL into memory for each unit test.

Project Champions

- TBD

Skills Needed

- Java coding proficiency
- Basic SQL knowledge

Objectives

- Produce a Maven plugin which can generate concept test data
- Plugin should be configurable in terms of which locales to include, which reference term sources.
- Plugin should be modular and support different sources of concept details (JDBC, OpenConceptLab API)

Extra Credit

- [Possibly] work with OpenConceptLab to provide API support on their end

Contact

- Leave a comment below

Design Proposal

It's common for an OpenMRS distribution to list the concepts it uses in classes of static constants, e.g.

Example dictionary class

```
public class Dictionary {
    public static final String ABACAVIR = "70056AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";
    public static final String ADULT_INPATIENT_SERVICE = "160536AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";
    public static final String ANTENATAL_CASE_NUMBER = "161655AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";
    ...
}
```

Classes like this would be the inputs for the plugin. The plugin when invoked would:

1. Parse out all of the constant fields of such classes
2. Include dependent concepts (e.g. set members and answers)
3. Connect to database and fetch data for all of those concepts

4. Serialize to an XML dataset file

The output file would use the XML dataset format currently used to provide test data in uni tests, e.g.

Example output dataset file

```
<dataset>
  <concept concept_id="70056" retired="0" datatype_id="4" class_id="3" is_set="0" creator="1" date_created="
2006-12-17 00:00:00.0" version="" changed_by="1" date_changed="2012-11-28 20:05:26.0" retired_by="1" uuid="
70056AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" />
  <concept_name concept_id="70056" name="Ziagen" locale="en" creator="1" date_created="2007-10-18 09:35:54.0"
concept_name_id="100397" voided="0" voided_by="1" void_reason="" uuid="100397BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
locale_preferred="false" />
  <concept_name concept_id="70056" name="ABACAVIR" locale="en" creator="1" date_created="2006-12-17 00:00:00.0"
concept_name_id="3277" voided="0" voided_by="1" uuid="3277BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB" concept_name_type="
FULLY_SPECIFIED" locale_preferred="true" />
  <concept_name concept_id="70056" name="ABC" locale="en" creator="1" date_created="2012-11-28 20:05:26.0"
concept_name_id="110802" voided="0" voided_by="1" uuid="110802BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
concept_name_type="SHORT" locale_preferred="false" />
  <concept_name concept_id="160536" name="Adult inpatient service" locale="en" creator="1" date_created="2012-
06-12 02:19:56.0" concept_name_id="108856" voided="0" voided_by="1" uuid="108856BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
concept_name_type="FULLY_SPECIFIED" locale_preferred="true" />
  <concept concept_id="161655" retired="0" datatype_id="1" class_id="7" is_set="0" creator="1" date_created="
2013-04-26 14:32:16.0" version="" changed_by="1" retired_by="1" uuid="161655AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA" />
  <concept_numeric concept_id="161655" units="" precise="0" />
  <concept_name concept_id="161655" name="ANC ID" locale="en" creator="1" date_created="2013-04-26 14:32:16.0"
concept_name_id="123687" voided="0" voided_by="1" uuid="123687BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
concept_name_type="SHORT" locale_preferred="false" />
  ...
</dataset>
```

The plugin itself could be configured in a project's POM something like this example:

Example plugin configuration

```
<plugin>
  <groupId>org.openmrs.contrib.maven</groupId>
  <artifactId>conceptdataset-maven-plugin</artifactId>
  <version>1.0</version>
  <configuration>
    <includeClasses>
      <includeClass>org.openmrs.module.kenyaemr.Dictionary</includeClass>
    </includeClasses>
    <connectionUrl>jdbc:mysql://localhost/openmrs</connectionUrl>
    <outputFile>${basedir}/src/test/resources/test-concepts.xml</outputDirectory>
  </configuration>
</plugin>
```

And when run would request database credentials, e.g.

```
mvn conceptdataset:generate
Enter database username: root
Enter database password: *****
```