Controlled Vocabulary Bulk Data Loading

Automation Exercise

# Background

FamilySearch has a taxonomy data system called the Controlled Vocabulary (CV). Data elements in CV are organized by their definition and are called concepts. Concepts have metadata attached to them as key/valued paired data known as attributes. Concepts also have concrete representations called terms which can be translated into multiple languages. Terms are typed by their purposes (e.g. “OFFICIAL” terms are used for the canonical representation of a concept, “OFFICIAL\_PLURAL” for the pluralized form of the canonical representation, and “ALTERNATE” terms are used for other, non-canonical representations). Terms can be organized into groups of related terms using a List.

To add data to the system, a concept is created to house the definition for the data element. Any attributes are added to the concept. Terms are then created for the concept to be consumable by outside systems. Optionally, terms can be added to a list to group them together.

Bulk Data Loading into the CV is accomplished using a Java application which takes a text file containing commands and communicates with the CV web service endpoints.

# Problem Description

FamilySearch Record Operations needs to automate the generation of CV content. Given a sample data set (Excel Spreadsheet) and list information, your task is to create a program that will convert the data set into the commands necessary to load the data through the Bulk Data Loading program.

## Requirements

Each entry in the spreadsheet must be converted into the required script commands to

1. create the concept with the appropriate TYPE attribute
2. create the term(s) for the concept with the appropriate term type
3. add the new term(s) to the pre-defined list (id: 126890)

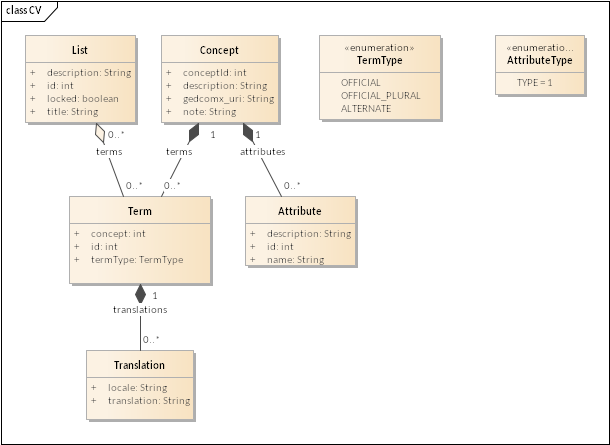
In addition, the list (126890) must be unlocked before you can add terms to it and should be locked again when the terms have been added.

You’re welcome to use any language with which you’re familiar however we’ll want to “run” your solution on the input file during your interview so we’ll need to know what you’re using in advance. Using an online code repository for your solution is highly encouraged (e.g. BitBucket or Github).

# Reference

## Data Model

Here is the Class Diagram for the CV System as described previously in the [Background](#_Background) section.



## Bulk Script Command Reference

The CV Bulk Tool understands commands to accomplish the following tasks:

* [Create a Concept](#_Create_Concept)
* [Update a Concept](#_Update_Concept)
  + You’ll use this to add the TYPE attribute to the concept
* [Create a Term](#_Create_Term)
  + You’ll use this to add terms to a concept
* [Update a List](#_Update_List)
  + You’ll use this to add terms to the list and to lock/unlock the list

All bulk command arguments are tab delimited. Several commands set variables for use by later commands. Variables are referenced by preceding the variable name with a dollar sign ($).

### Create Concept

Creates a new concept and sets the variable “conceptId” with its identifier.

#### Syntax

create\_concept <description>

#### Example

create\_concept A shiny new concept description.

### Update Concept

Updates an existing concept with the desired metadata.

#### Syntax

update\_concept <conceptId> attrs=<id>:<value>

#### Example

update\_concept $conceptId attrs=1:DESCRIPTOR

### Create Term

Creates a new term and sets the “termId” variable to the new identifier.

#### Syntax

create\_term <translation> <locale> <type>

#### Example

create\_term Shiny Concept en $conceptId

### Update List

Updates an existing list.

#### Syntax

update\_list <listId> lock={true|false}

update\_list <listId> terms=<termId>[,<termId>…]

#### Example

update\_list 126890 lock=false

update\_list 126890 terms=$termId