**Data Dictionary Creator**

# Introduction

This document describes the process to configure and run a custom tool, written in Java, to introspect and document the data dictionary for a Salesforce instance.

The tool leverages the Salesforce REST API and an OAuth Password flow login process.

The steps required to configure and run this tool are documented in detail in the following sections, however at a high level they are:

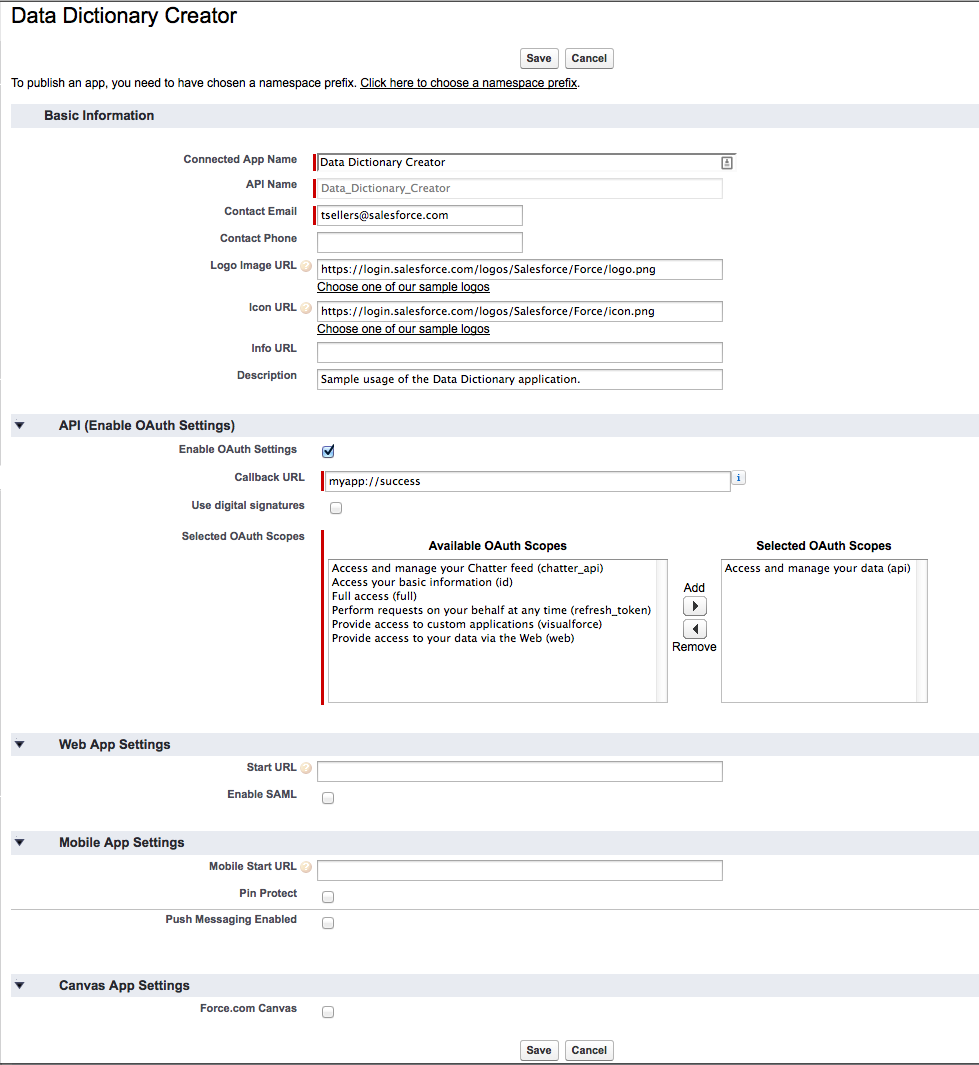
1. Configure the Salesforce instance to allow the tool connection by setting up a new Connected App in Salesforce.
2. Configure the tool using the app.properties file that is in the package.
3. Run the tool from the command line.

This tool requires Java 1.7 to run.

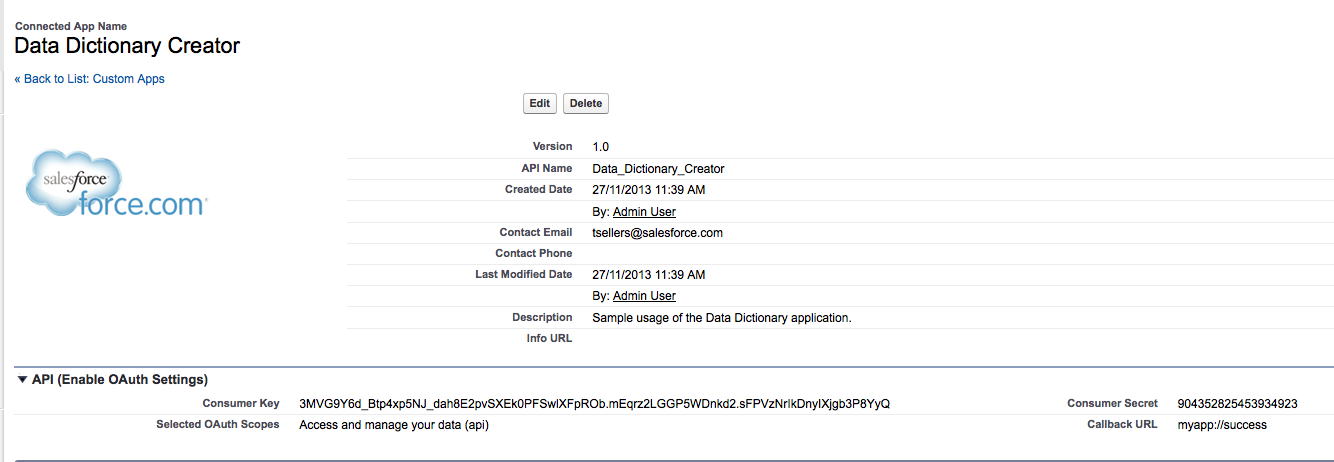
# 

# Create Connected App

The first step required is to create a new Connected App in the sandbox environment that allows the tool to connect via OAuth. Once tool has been executed, it is recommended that this Connected App privilege be removed.

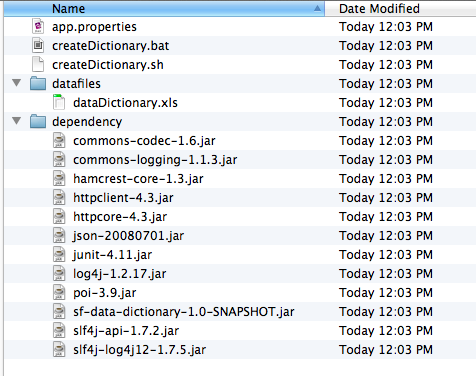


Once created, take note of the Consumer Key and Consumer Secret fields created for this application

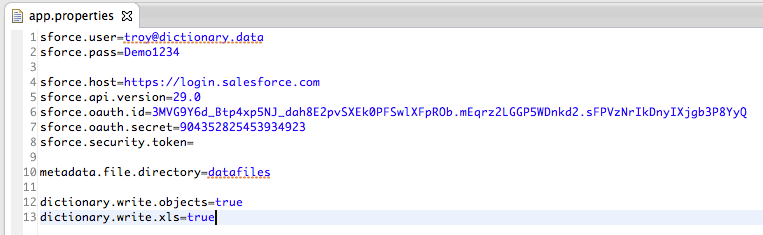


# Configure the Package

The following artifacts would have been provided, if you wish to build from source see Appendix A in this document.



Open the file app.properties and configure your username, password and connected application properties from the previous step. An example of how this file should look is shown below.



This file is configuring the tool to point to a Developer Environment production instance.

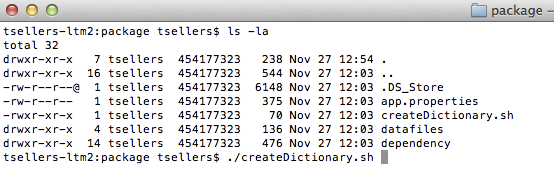
The Connected App Consumer Key maps to sforce.oauth.id property and the Consumer Secret maps to the sforce.oauth.secret property.

dictionary.write.objects=true will cause the tool to write a single .csv file for each object

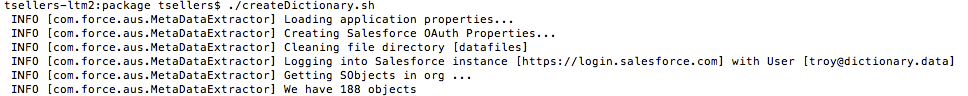
dictionary.write.xls=true will cause the tool to write a single XLS file, each object in its own sheet.

Run the tool using the following command

./createDictionary.sh



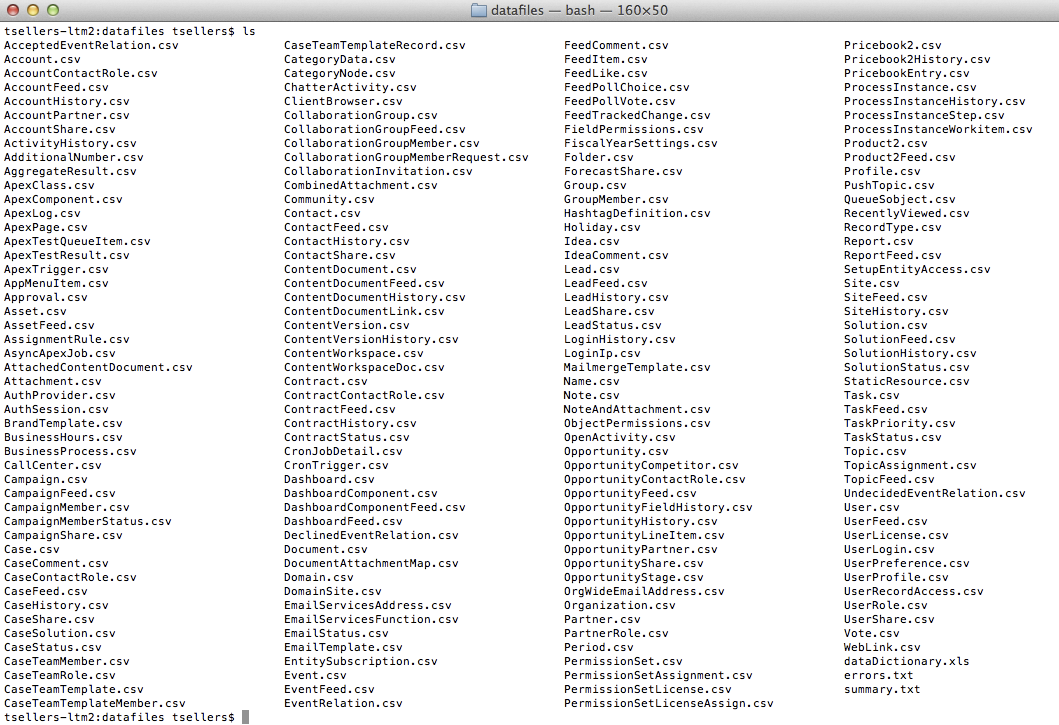
When executing, you should see the output begin as shown below.



On completion, the datafiles directory will be where all the output is found.



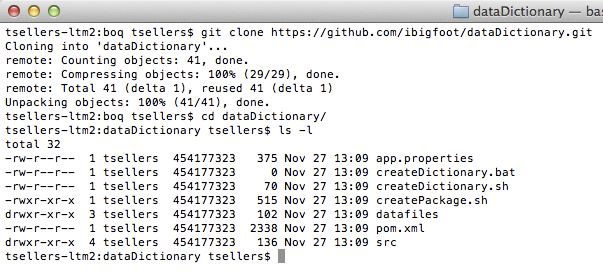
Two files that will be output regardless of configuration will be summary.txt and errors.txt.



# Appendix A – Build from Source

The source code for this tool can be found at

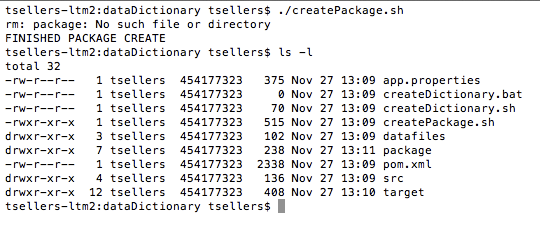
<https://github.com/ibigfoot/dataDictionary>



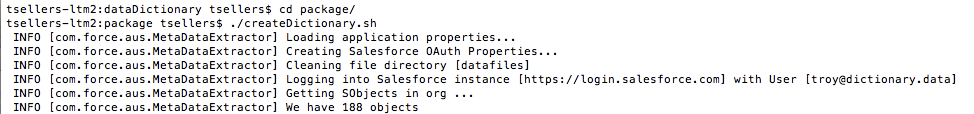
Build with Maven (requires Maven 3 and Java 1.7)



Once build is successful, create the package itself using createPackage.sh



Navigate to /package and run createDictionary.sh



This should populate the datafiles directory.