

SETTING UP DATA LOADER 9.0 FOR AUTO PROCESSES

Requirements and Assumptions:

The purpose of this document is to document findings on the setup of Data Loader 9.0 for automated processes. I will be updating this document as new information comes along.

This document assumes that the reader has an understanding on the different processes run by Data Loader 9.0 (insert, upsert, delete, extract) and will not be discussing the details of the processes. This document will focus on setting up the auto upload processes.

I. Requirements for the batch process:

- Data Loader 9.0 must be installed into a directory on the server machine that will be running the batch processes. For the purposes of this document, the default directory has been used:

 C:\Program Files\salesforce.com\Apex Data Loader 9.0

- process-conf.xml file. A sample of this file can be found at:

 C:\Program Files\salesforce.com\Apex Data Loader 9.0\samples\conf

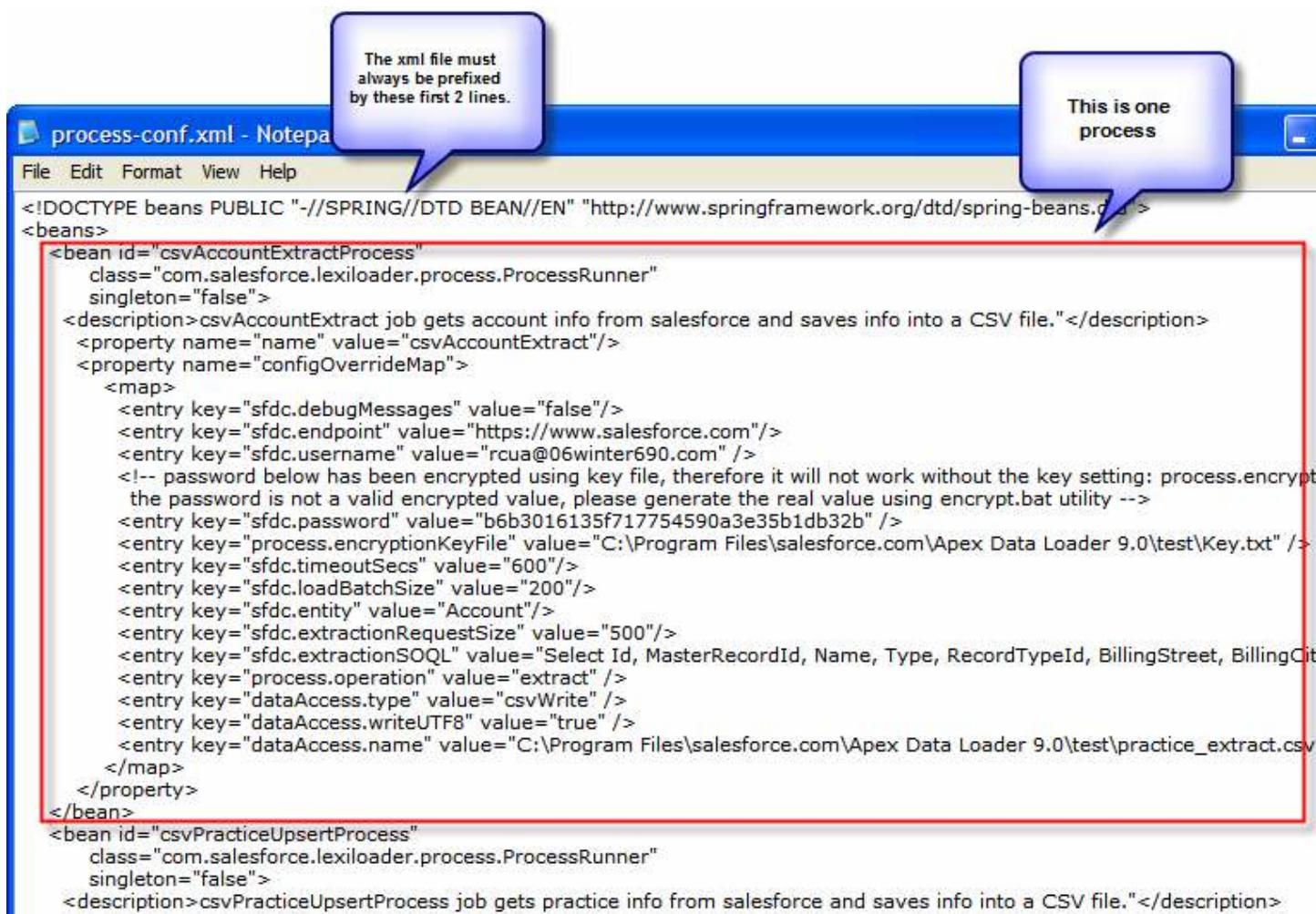
[Java Runtime Environment \("JRE"\) v1.4.2_03](#) This is required for the batch processes to run. You may find that even if you are able to run Data Loader through the Data Loaders GUI, this is not yet available.

II. process-conf.xml

The process-conf.xml file contains the setup for each action that needs to take place. Each Extract, Insert, Upsert, or Delete function needs to have a different section in the process-conf.xml file. This file must be placed within a folder where Data Loader is installed. For example:

 C:\Program Files\salesforce.com\Apex Data Loader 9.0\test

Opening the XML file will show that the different processes are segregated by `<bean>` tags as shown below, there are no limitations to the number of processes (beans) in each XML file:



NOTE: Be careful in using different XML editors in editing the process-conf.xml file. Some XML editors will pad XML tags to the beginning and the end of the file, causing the process to fail without explaining why. Make sure that the first two lines of your XML file is the same as above and should never change.

III. process-conf.xml Parameters:

Parameter	Description
<pre data-bbox="236 1607 953 1672"><bean id="PROCESSNAME" class="com.salesforce.lexiloader.process.ProcessRunner" singleton="false"></pre>	This first line uniquely defines the process in the XML file. This has to be unique and will be used to call the process later on. Only change the PROCESSNAME and leave all other parts the same.
<pre data-bbox="236 1672 953 1784"><description>ENTER YOUR DESCRIPTION HERE</description></pre>	A description of the process. I have not seen a use for this other than to describe the process within the XML file.
<pre data-bbox="236 1784 953 1824"><property name="name" value="PROCESSNAME" /></pre>	I normally copy the PROCESSNAME and place it in this parameter. This is

Parameter	Description
	used in the log files when debug messages are showing. This is the name that is shown on the debug logs.
<property name="configOverrideMap"> <map>	These two lines initiate the parameters for the process. They should not be changed and should always be the same.
<entry key="sfdc.debugMessages" value="false"/>	Set this to 'True' if you want detailed debug messages to appear in the command line.
<entry key="sfdc.debugMessagesFile" value="c:\dataloader\samples\status\accountMasterSoapTrace.log"/>	Set the value to a directory and a file to keep the detailed log file in. WARNING: This can create an extremely large file. Make sure that you have space your hard drive before this is set.
<entry key="sfdc.endpoint" value="https://www.salesforce.com"/>	Keep default, unless loading into Sandbox.
<entry key="sfdc.username" value="rcua@06winter690.com" />	Set this as the username of the user you are importing as.
<entry key="sfdc.password" value="b6b3016135f717754590a3e35b1db32b" />	See the section about Encrypting Passwords in this document.
<entry key="process.encryptionKeyFile" value="C:\Program Files\salesforce.com\Apex Data Loader 9.0\test\Key.txt" />	
<entry key="sfdc.timeoutSecs" value="600"/>	System time out value.
<entry key="sfdc.loadBatchSize" value="200"/>	System batch load size. Set this to a higher value only if internet speed is faster.
<entry key="process.operation" value="extract" />	Signifies the process being run. Possible Values: <ul style="list-style-type: none">- Extract- Insert- Delete- Upsert
<entry key="sfdc.entity" value="Account"/>	Signifies the object that you are running this process for. This should be the API name of the object when you are running processes for custom objects.
<entry key="sfdc.extractionRequestSize" value="500"/>	Batch size for extract processes only.
<entry key="sfdc.extractionSOQL" value="Select Id, MasterRecordId, Name, Type FROM Account " />	This is the SOQL query run for extract processes. You can use the Data Loader GUI to generate the SOQL statement and conditions and simply cut and paste into this entry key.
<entry key="dataAccess.type" value="csvWrite" />	Signifies the action when doing the

Parameter	Description
	<p>process for Extract, Insert, Delete, Upsert. This key should be used in conjunction with the “process.operation” key.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> - csvRead - csvWrite - databaseRead - databaseWrite <p>NOTE: I have not been able to perform a databaseRead or databaseWrite before, however documentation suggests that this will allow to write or read directly into an ODBC connection.</p> <p>When using Extract processes, use csvWrite.</p> <p>When using Delete, Insert, and Upsert processes, user csvRead.</p>
<entry key="dataAccess.writeUTF8" value="true" />	When using a “csvWrite” dataaccess type, set this to True when you want files to be extracted as UTF-8.
<entry key="dataAccess.name" value="C:\Program Files\salesforce.com\Apex Data Loader 9.0\test\ extract.csv" />	<p>Signifies the location and file name of the dataaccess type process.</p> <p>If running a csvRead, the process will look for the file in this location.</p> <p>If running a csvWrite, the process will replace the file in this location.</p>
<entry key="process.mappingFile" value="C:\Program Files\salesforce.com\Apex Data Loader 9.0\test\ upsert_mapping.sdl"/>	Signifies the location of a data loader mapping file. This is the same mapping file used when saving a mapping from the data loader GUI. This is required for insert, delete and upsert processes.
<entry key="sfdc.externalIdField" value="Customer_ID_c"/>	Signifies the External ID field used for the upsert process. This is required for upserts.
<entry key="process.statusOutputDirectory" value="C:\Program Files\salesforce.com\Apex Data Loader 9.0\test\logs" />	Signifies the directory where the data loader success and error files will be created in.
<entry key="process.outputSuccess" value="C:\Program Files\salesforce.com\Apex Data Loader 9.0\test\Logs\csvUpsertProcess_success.csv" />	Signifies the directory and filename where the success and error files will be created. If this key is not specified, the “process.statusOutputDirectory” key will contain similar filenames as ones generated by the data loader
<entry key="process.outputError" value="C:\Program Files\salesforce.com\Apex Data Loader	

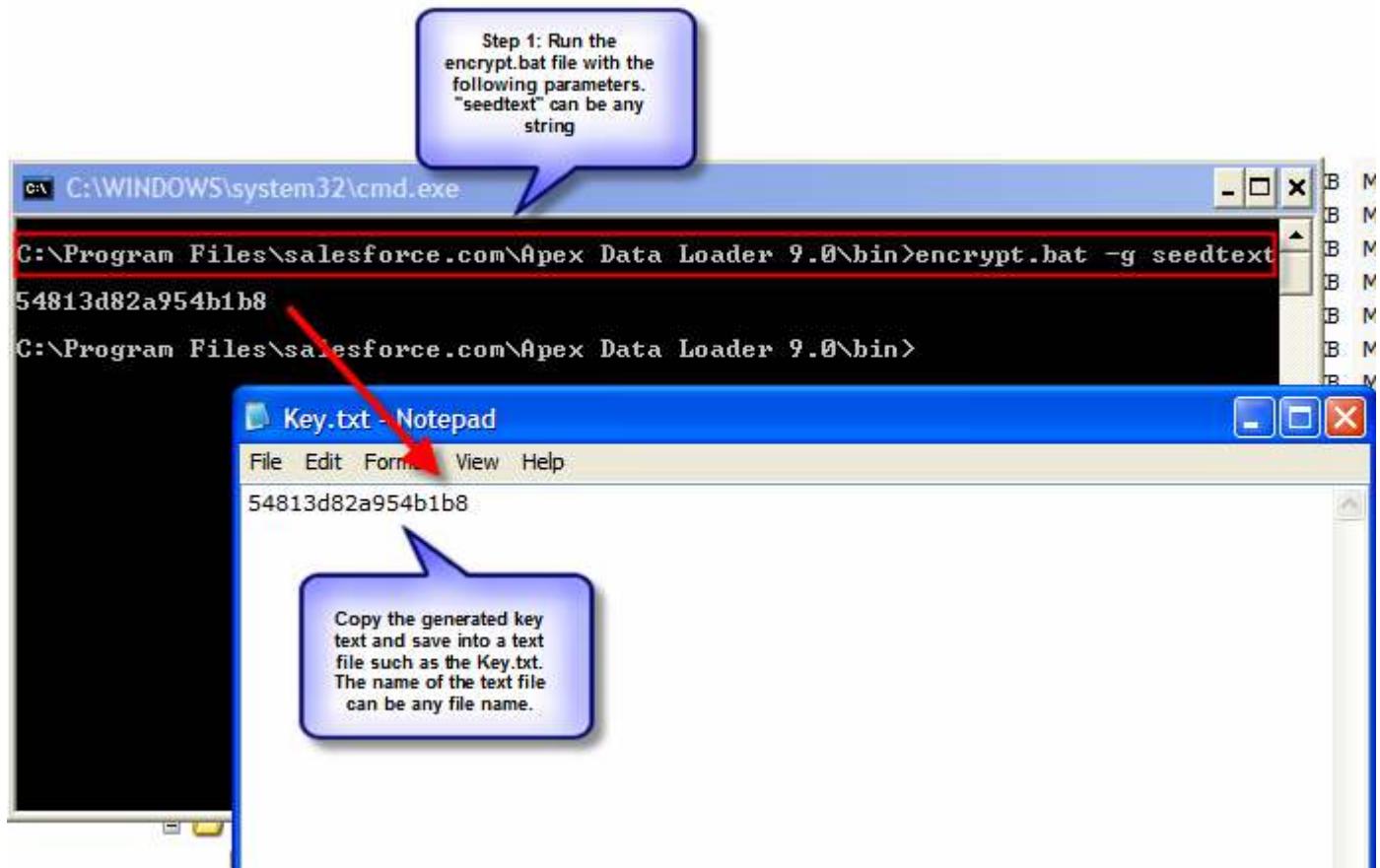
Parameter	Description
9.0\test\Logs\csvUpsertProcess_error.csv" />	GUI.

IV. Encrypting Passwords:

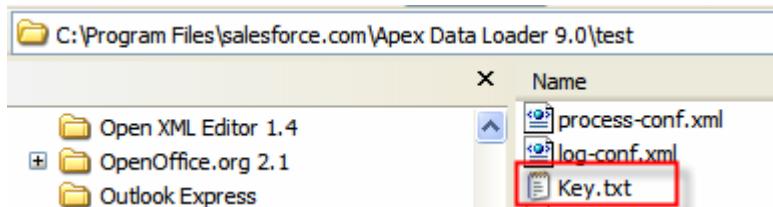
Encrypting passwords for use in the Data Loader command line involves a 2 step process. The following section details these steps:

Step 1: Create the Key File:

Create the Key.txt file. (File name can be different)



Save the Key.txt file into the directory where the process-conf.xml file is located.



Step 2: Create the password hash:

```
C:\> C:\Windows\system32\cmd.exe
C:\> C:\Program Files\salesforce.com\Apex Data Loader 9.0\bin>encrypt.bat -e password
"5244471995e1185a02ad07524d739e7f"

C:\>
```

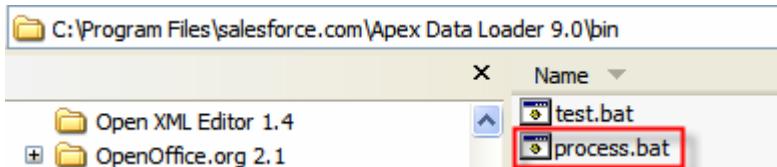
Now you need to update the process-conf.xml file's process beans to include these 2 details into the following entry keys:

```
<entry key="sfdc.password" value="5244471995e1185a02ad07524d739e7f" />

<entry key="process.encryptionKeyFile" value="C:\Program Files\salesforce.com\Apex Data Loader 9.0\test\key.txt" />
```

V. Running Processes

After the process.conf.xml file is created and the beans in the file are properly created, you can run the process by calling the process.bat file from the bin directory in the Data Loader installation directory:



The syntax is as follows:

```
process.bat <directory that contains the process-conf.xml> <process name>
```

Example:

```
process.bat "c:\program files\salesforce.com\apex data loader 9.0\test" csvAccountExtractProcess
```

Note: To call different processes one after the other, create a batch file in the bin directory and prefixing each call with the "call" function. The image below shows how to call the csvAccountExtractProcess 3 times and clearing the screen after each run.

```
test.bat - Notepad
File Edit Format View Help

cls
call process "c:\program files\salesforce.com\apex data loader 9.0\test" csvAccountExtractProcess
cls
call process "c:\program files\salesforce.com\apex data loader 9.0\test" csvAccountExtractProcess
cls
call process "c:\program files\salesforce.com\apex data loader 9.0\test" csvAccountExtractProcess
```

VI. Removing Debug Messages

To remove debug messages from in the command line, make sure you have the log-conf.xml file in the directory where your process-conf.xml file is located. The log-conf.xml file can be found in the “conf” directory of your data loader.

