

SalesForce Dataloader

The SalesForce Dataloader can be integrated with backend processes in the jtel System, which are run, for example, nightly, to synchronise data between the jtel system and SalesForce.

Exactly what processes and operations are performed, is project specific, and may vary greatly from installation to installation.

This section deals with the way in which the dataloader accesses SalesForce, and the creation of a new authorisation token.

Create a User in SalesForce with Access to the Dataloader

Use the normal Salesforce administration procedures to achieve this. You will need the following from the user account to proceed from here:

- SalesForce User Login Name
- Password
- SalesForce Security Token

Update the jtel Dataloader to use the new credentials

This is a complicated process, requiring access to the command line of the system.

1. Login to the telephony / application server of your system, as the local administrator.

2. Map a drive to the location of the dataloader, and change to the directory.

Change (CustomerName) to the name of the customer directory.

Map drive and change to directory

```
net use z: \\acd-lb\shared
z:
cd \{CustomerName}\SalesForce\jteldataloader-0.3
```

3. Create a secret key

Issue the following command. **SecretKeyString** can be changed to anything you desire.

Create secret key

```
java -cp dataloader-40.0.0-uber.jar com.salesforce.dataloader.security.EncryptionUtil -g SecretKeyString
```

4. Note the output

The output from the command above is saved to a file "key.txt" saved in the current directory.

5. Create a token

Copy the following command, and replace **<password>** with the password of the account, which should be used to access Salesforce, and **<token>** with the security token from the user account (obtained from Salesforce before this procedure was started).

Note: there is NO space between the password and the token.

Create token

```
java -cp dataloader-40.0.0-uber.jar com.salesforce.dataloader.security.EncryptionUtil -e <password><token> key.txt
```

Example output:

```
2017-10-18 13:12:16,747 INFO [main] security.EncryptionUtil main (EncryptionUtil.java:365) - 4ff73dc5ea5350cb92be28ad021c633dc62fa0cf735d00c06c4ec0a63c1eecd28c059b80df1541
```

Note: this line will inevitably be broken in the console output to the next line in the window. It is imperative to copy the whole output if this is the case.

The portion of the output required, is the token at the end of the line:

```
4ff73dc5ea5350cb92be28ad021c633dc62fa0cf735d00c06c4ec0a63c1eecd28c059b80df1541
```

6. Modify process-conf.xml

The token from the step above, and the user name, are now changed in the file **config\process-conf.xml**.

Note - on the target system, there may be several files named for example:

- process-conf.production.xml
- process-conf.sandbox.xml
- process-conf.xml

The active file is always **process-conf.xml**, the other files are configuraton backups for testing, for example, against the sandbox. Note - it may be sensible to modify these files accordingly too.

Open process-conf.xml with an editor, for example using the following command:

Edit process-conf.xml

```
start notepad++ config\process-conf.xml
```

It is simplest, to search for the following XML Keys:

- sfdc.username
- sfdc.password

and replace the value in each case.

See the following XML excerpt for an example:

- SalesforceLoginName - the login name of the Salesforce account.

- CreatedToken - the token created in step 5.

process-conf.xml

```
...
    <entry key="sfdc.username" value="SalesForceLoginName"/>
    <entry key="sfdc.password" value="CreatedToken"/>
...
```

Note: these settings will be present in **several locations** in the file. All of these must be modified.

Example process-conf.xml:

process-conf.xml

```
...
    <entry key="sfdc.username" value="test.user@example.com"/>
    <entry key="sfdc.password" value="7687a73934af24359a27577534b81c5beed32fa0cf735d00c06c4ec0a63c1eecd28c059b80df1541"/>
...
```

7. Done

That's it. The next time the dataloader process runs, it will use the new account data.

Note: it is advisable, to setup the Salesforce account, so that the password does not expire.