How To Install And Run The Python Code

**Step1: Install Python on your server**

Tested Version: Python 3.11.0

The following Python libraries will need to exist:

1. importlib
2. requests
3. csv
4. salesforce\_bulk
5. os
6. sys
7. datetime

You can use “pip3 install” to load these libraries.

**Step2: Create Folder structure for installation and Unzip SourceCode.zip**

The following structure needs to exist:

/<Root install folder>

/Input

/Output

/logs

/\_\_pycache\_\_

1. Load SourceCode.zip to the <Root install folder> (SourceCode.zip will be provided)

Below highlighted are the required folders/sub folders (plus the /logs folder under the /output folder):

A picture containing application

Description automatically generated

**Step3: SQL Server**

1. Need this driver for connection to SQL Server (Or whatever you will be using) ODBC Driver 17 for SQL Server
2. The following python libraries are needed
   1. pyodbc
   2. os
   3. pandas
   4. datetime
   5. json
   6. simple\_salesforce

**Step4: Load Execution**

Run Manual For All Files

1. Edit the run\_SQLServerExport.bat to point to the <Root install folder>
2. Double click on run\_SQLServerExport.bat

Run Manual For Individual Files

There may be times where you want to re-load files manually.

1. To create your own load, copy your CSV(s) (UTF-8 format) the name of the file(s) are important to the /Input folder, it must follow the naming conventions (See file naming conventions at the bottom)
2. Double click the run\_Load.bat (Please make sure the first line inside the run\_Load.bat is pointing to your <Root Folder>)

Run as an automated job

1. Create a Windows Task to execute the run\_SQLServerExport.bat file on a frequency of your choice.

**Step5: Review Load Results**

After the load process completes, there will be files created in the /Output folder for any tables that had records to process. For each load you will see a Date/Time stamp prefixed for each file. This will keep a history of files that were processed and a way for you to group runs.

Output Files:

RECORDLOCKS\_ - Will have a list of rows that could not be processed because of record locking issues.

ERROR\_ - Will contain any errors for the load

After a load has completed, Job stats will be automatically loaded to a Salesforce object called Data\_Loads\_Stats\_\_c

Go to the tab “Data Load Stats” to view the statistics. From there you will be able to see which loads had errors, and then you can go back to the /Output folder and interrogate the corresponding ERROR\_ file.

NOTE: For the Quote\_Insert loads, you will see “errors” for when we are inserting a record that already exists, this is supposed to happen. Quote\_Inserts will be successful only for NEW Quotes coming from E2. Any changes to Quotes will be picked up in the Quote\_Update file.

You will then need to fix the errors in your E2 system and when the next load runs, those fixes will hopefully result in a successful load.

**File Naming Conventions:**

It is imperative that you follow these naming conventions. Only these file names will be picked up and processed. Also note that for any loads for the Quote data, you must always create TWO files (Duplicates), one for INSERT and one for UPDATE.

A screenshot of a computer

Description automatically generated with medium confidence

1. How to update the Config file if User/password changes

If Josh’s password changes, you will need to update the os.environ['password'] parameter in the Client\_Config file (located in the c:\Alias-SQL-Table-Loads folder).

See below

[Client\_Config]

# Desert PROD

os.environ['oauthurl'] = "https://desertcoatingsolutions.my.salesforce.com/services/oauth2/token"

os.environ['client\_id'] = "3MVG9ux34Ig8G5epuXWEQpQ7Gz\_zuuv2Soyr2ZwaDScXJyqC1EqxbHYqUZfZ7Ftgstaq\_G0gfHorcViPUeX1a"

os.environ['client\_secret'] = "A10B5FBDBB8FF0B968BA8B44C32267F45477F3B23EA738B941D83038759E3476"

os.environ['username'] = "josh@desertpowder.com"

os.environ['password'] = "Z3U=p3tyTUHw9C!"

os.environ['host'] = "https://desertcoatingsolutions.my.salesforce.com/"

os.environ['security\_token'] = ""

os.environ['DesertRTID'] = "012Dn000000F74NIAS"

os.environ['StandardRTID'] = "012Dn000000F74SIAS"

os.environ['DesertContactRTID'] = "012Dn000000F74XIAS"

os.environ['StandardContactRTID'] = "012Dn000000F74TIAS"

os.environ['DesertOppRTID'] = "012Dn000000F74cIAC"

os.environ['StandardOppRTID'] = "012Dn000000F74JIAS"

#Set Owner to Josh

os.environ['RecordOwnerId'] = '005Dn0000044OGHIA2'

1. How to update the code if they ever wanted to do a full reload of an entire E2 SQL Server table

In order to do a full reload of all records in a SQL Server table, you simply need to remove the filter(s) in the WHERE clause.

For example if you wanted to reload the entire Customer data from E2:

1. Edit the LoadAutomation.py (located in the c:\Alias-SQL-Table-Loads folder)

And change the following highlighted in yellow, from this:

# DSC\_Customer Load

myQuery = """Select LastModifiedDate from Account Where E2\_Customer\_Key\_\_c like 'DCS\_%' and Loaded\_From\_Python\_Process\_\_c = 'Y' and LastModifiedDate <> null order by LastModifiedDate desc limit 1"""

AccountLastRunDate = getLatestRunDate(myQuery)

DCSAccountLastRunDate = AccountLastRunDate

sqlQuery = "SELECT replace(replace(APContact,char(10),''),char(13),'') as APContact,replace(replace(replace(replace(BAddr1,char(10),''),char(13),''),'#',''),',','|') as BAddr1,replace(replace(replace(replace(BAddr2,char(10),''),char(13),''),'#',''),',','|') as BAddr2,BCity,BState,BZIPCode,Phone,Website,replace(replace(replace(CustName,char(10),''),char(13),''),',','|') as CustName,replace(replace(replace(CustCode,char(10),''),char(13),''),',','|') as CustCode,'DCS\_' + CONVERT(varchar(100), CustCode\_ID) as CustCode\_ID,CONVERT(nvarchar,LastModDate, 23) as PreviousModDate,row\_number() over(order by(CustCode\_ID)) as RowNum\_Of\_Source\_File,'Y' as LoadedByPython,GetDate() as LoadDate,'DCS\_Customer.csv' as Source\_File,'DESERT' as LoadForCompany FROM CustCode " + " WHERE LastModDate > " + "'"+ DCSAccountLastRunDate + "'" + " or EnterDate > " + "'"+ DCSAccountLastRunDate + "'" + " ORDER BY CustCode\_ID"

df = pd.read\_sql(sql=sqlQuery, con=connDCS)

df.to\_csv('DCS\_Customer.csv')

to this:

# DSC\_Customer Load

myQuery = """Select LastModifiedDate from Account Where E2\_Customer\_Key\_\_c like 'DCS\_%' and Loaded\_From\_Python\_Process\_\_c = 'Y' and LastModifiedDate <> null order by LastModifiedDate desc limit 1"""

AccountLastRunDate = getLatestRunDate(myQuery)

DCSAccountLastRunDate = AccountLastRunDate

sqlQuery = "SELECT replace(replace(APContact,char(10),''),char(13),'') as APContact,replace(replace(replace(replace(BAddr1,char(10),''),char(13),''),'#',''),',','|') as BAddr1,replace(replace(replace(replace(BAddr2,char(10),''),char(13),''),'#',''),',','|') as BAddr2,BCity,BState,BZIPCode,Phone,Website,replace(replace(replace(CustName,char(10),''),char(13),''),',','|') as CustName,replace(replace(replace(CustCode,char(10),''),char(13),''),',','|') as CustCode,'DCS\_' + CONVERT(varchar(100), CustCode\_ID) as CustCode\_ID,CONVERT(nvarchar,LastModDate, 23) as PreviousModDate,row\_number() over(order by(CustCode\_ID)) as RowNum\_Of\_Source\_File,'Y' as LoadedByPython,GetDate() as LoadDate,'DCS\_Customer.csv' as Source\_File,'DESERT' as LoadForCompany FROM CustCode " + " ORDER BY CustCode\_ID"

df = pd.read\_sql(sql=sqlQuery, con=connDCS)

df.to\_csv('DCS\_Customer.csv')

I would suggest you don’t actually change the original code but instead, make a copy of the code, comment out the original code out and then use the new code. Then after the update you can simply uncomment out the original and delete the changed code. In Python the # is used as a comment.

1. The timing of fixing data issues/errors in E2 and being able to load that fix into Salesforce.

The way the load works, is, we base the SQL Server export(s) on the Salesforce object’s LastModifiedDate, so we can pick up the records that have been created/changed since the last load was done. (Currently running every 15 minutes)

So the only way any E2 data fix/change will be picked up and loaded, is if the LastModDate/EnterDate/DateEnt of **the changed records in E2 is OLDER than the LastModifiedDate** of the Salesforce object you are loading.

I would suggest that you do your data fixes after the final data load run for the day and let all those changes go over on the first run the following day.