**ImportToAccess.py**

Lines 1 – 9: Beginning of the program in which it will connect to Access Database and Pentagon Database using PyODBC.

PyODBC is a python extension that will use Microsoft Access Drivers to properly connect to Access and DSN, Data Source Name, information to connect to an SQL server. For more information on PyODBC, visit <https://code.google.com/p/pyodbc/>.

Lines 11 – 22: These lines are strung throughout the program. They are definitions that were used to minimize the repetitiveness of storing information in Access and SQL tables.

Lines 26 – 135: These lines are used to parse through the 00.txt file that is located in the RFQs from Exostar folder. It will take the information that is found and store it within an XML file named RFQtext. If in the future you do not need or want to store information in an XML file, simply comment out lines 134 and 135. This will prevent the program from writing the RFQtext XML file.

Lines 139 – 183: This where the program will start to parse through the XML tree that is created at line 34. The XML tree is stored in memory until the program ends and/or also is saved within RFQtext XML file. It will then proceed to gather the information needed and store the values within the proper table names in both Access and Pentagon.

The beginning lines of this section will look at the root of the XML tree for the RFQ number as well as start creating a word document that will be used by Daniel King or any sales associate to send out to vendors. It will then proceed to look for RFQ number in NNS\_RFQ table and if found, will stop to ask user if they want to continue or not but will not add the RFQ since root number is found. After RFQ number processing, the program will then go through the XML tree elements so that the information can then be processed into Access Database and Pentagon Database. At first parse, the program will find the purchaser contact element. It will compare the information to the Buyer Contact table and if found, will proceed through the program as intended unless the comparison fails and will then proceed to ask the user whether or not the user wants to continue without properly having found the buyer contact information.

At the end of the program, there are .close() statements. These will close the cursor objects and the database connections, preventing anymore access after the programs completes. It will also write the quote document that the sales representative will use to send to vendors. Since I created a Importtxt.bat(batch) file to run the program through Access Vba, it will continue to loop until the user closes the actual prompt, in this case, the command prompt.

If all is well, the program should properly execute and put all the information where it needs to go. If there is an error, unfortunately the user will need to delete specific information so that the program can run again.

**InvImport.py**

This is a relatively simple program, written for Jacob Davis for ease of importing invoice numbers from Pentagon as well as inputting a UPS tracking number.

Lines 1 – 6: Beginning of the program in which it will connect to Access Database and Pentagon Database using PyODBC. For more information on PyODBC, visit <https://code.google.com/p/pyodbc/>.

Lines 8 – 32: The program will ask the user for an invoice number so that it can search through the Invoice table in Pentagon and obtain the correct information that will be stored in the Access Database.

Lines 34 - 66: After the information is found in Pentagon, the program will then proceed to search for the Purchase Order number, PO, in Access and input the information that is saved in the variables. Once he information is stored correctly, the program will close the connections made and properly loop as per the script written in Importinv.bat until the user is done and closes the command prompt.

**Edit**:

All of the Pentagon Quote Insertion code is commented throughout ImportToAccess.py file. Daniel did not need the quotes in Pentagon so when he is ready and in need of use, just simply uncomment and run.