NOVEC EXpansion Identification System

User Manual

May 3, 2015
Revision B

Austin Orchard
Brian Smith
Tygue Ferrier

GEORGE MASON UNIVERSITY

NOVEC
Investing in Our Communities
SM

Table of Contents

Software Requirements

Operating System

NEXIS is assumed to be run on a Windows machine. The python scripts should be portable to other operating systems but there has been no testing done for other operating systems.

Database

NEXIS requires access to a MySQL database. MySQL can be obtained from http://dev.mysql.com/downloads/windows/ for windows machines

Python 2.7

NEXIS is built using Python 2.7. Python 2.7 can be downloaded at https://www.python.org/download/releases/2.7/ .

Python 2.7 can be installed as part of *Anaconda* from Continuum Analytics, a "*(c)ompletely free enterprise-ready Python distribution for large-scale data processing, predictive analytics, and scientific computing*." https://store.continuum.io/cshop/anaconda/ Anaconda includes the design environment Spyder.

NEXIS requires the following packages: datetime, MySQLdb, os, fiona, shapely, pyproj, csv. To install the packages with Anaconda, open an Anaconda Command Prompt and type *conda install <package name>* or *pip install <package name>*. To install using python, go to python27 folder and type *pip install <package name>*.

Sometimes firewalls can prevent installation of packages; you can download the packages directly from http://www.lfd.uci.edu/~gohlke/pythonlibs/ as .whl files. Then type *pip install <whl file name>* from the command prompt in the directory where the whl file is located.

Installation Instructions

Database

To create the NEXIS schema, connect to the database. Run createNexisSchema.sql. The name of the database server must be up to date in nexisDButils.py in the "__init" section.

NEXIS

Create a folder for NEXIS. Place the following source files in this folder: logger.py, lolaUpdate.py, loudounReports.py, nexisDButils.py, nexisInterface.py, records.py. Place the csv record with updates in the same folder. NEXIS will create nexislogs and nexisreports folders in that directory to store logs and reports.

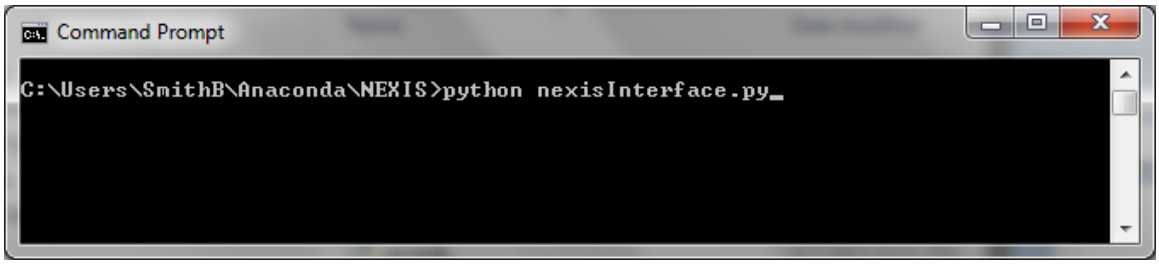
Service Area

The folder containing the shape file showing NOVEC's service area must be updated in loudounReports.py in the "findRecordsInServiceArea" section of loudounReports.py. Currently, it is assumed the shape file is using EPSG 2283 coordinates. If a different coordinate system is used, this must also be updated in the "findRecordsInServiceArea" section of loudounReports.py.

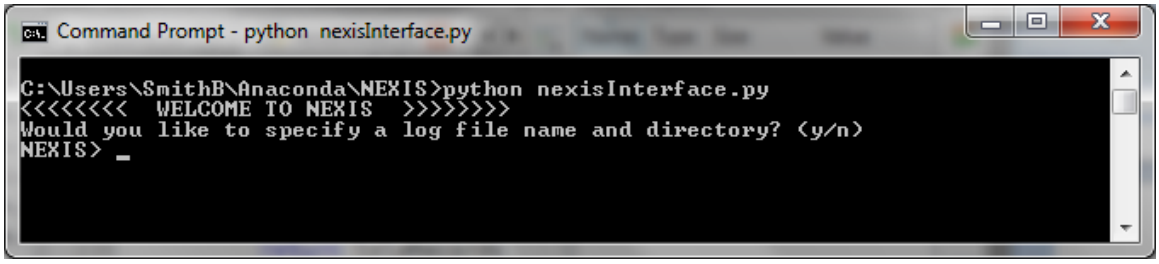
Using NEXIS

To start NEXIS, open a command prompt and move to the NEXIS folder.

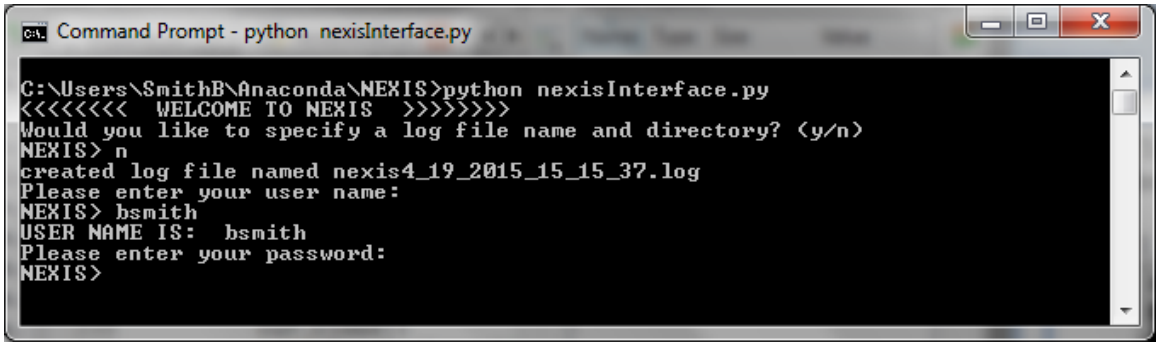
1. At the command line, type *python nexisInterface.py*



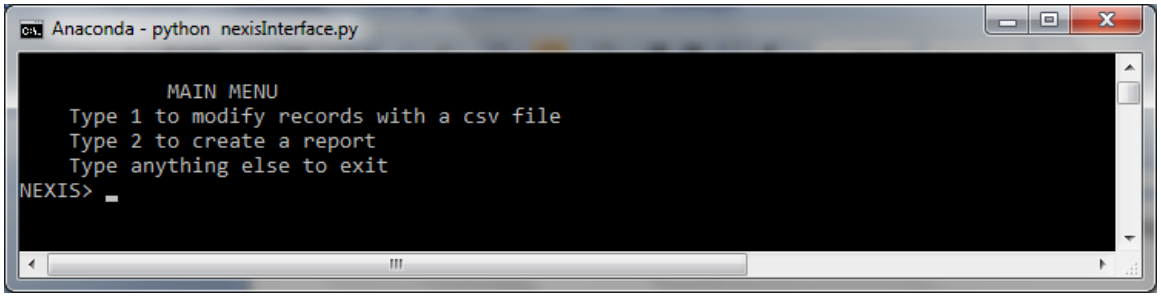
2. The first option is whether to specify the log file name or whether the log file will automatically be generated (recommended). The only acceptable responses are "y" or "n". Other responses will exit NEXIS. If the log file is automatically generated, the log file name will be "nexis" followed by the date and time and ".log"



3. The user will then need to input their database user name and password:

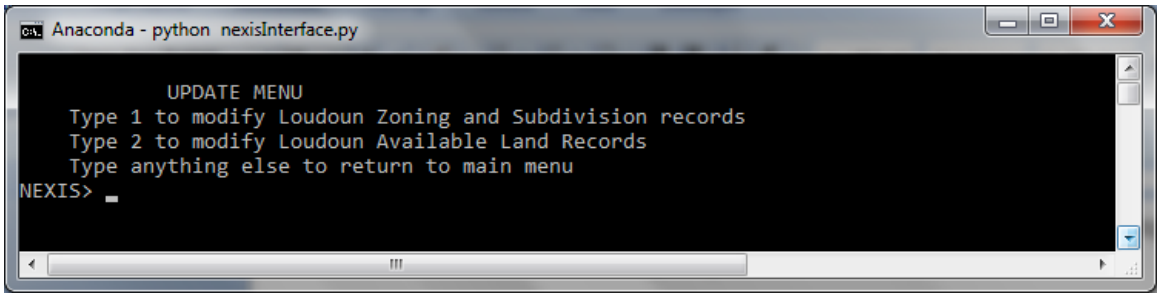


4. The user will then be able to choose between modifying the records and creating a report.

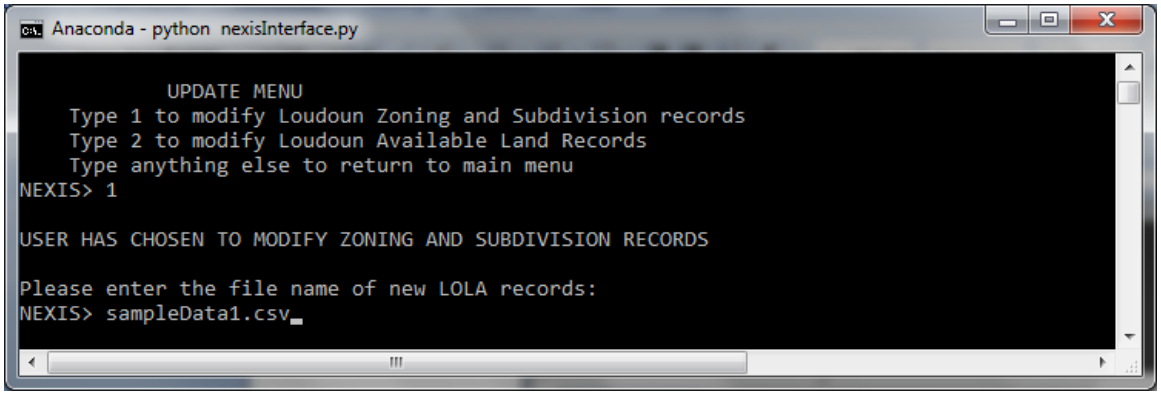


Modifying Records

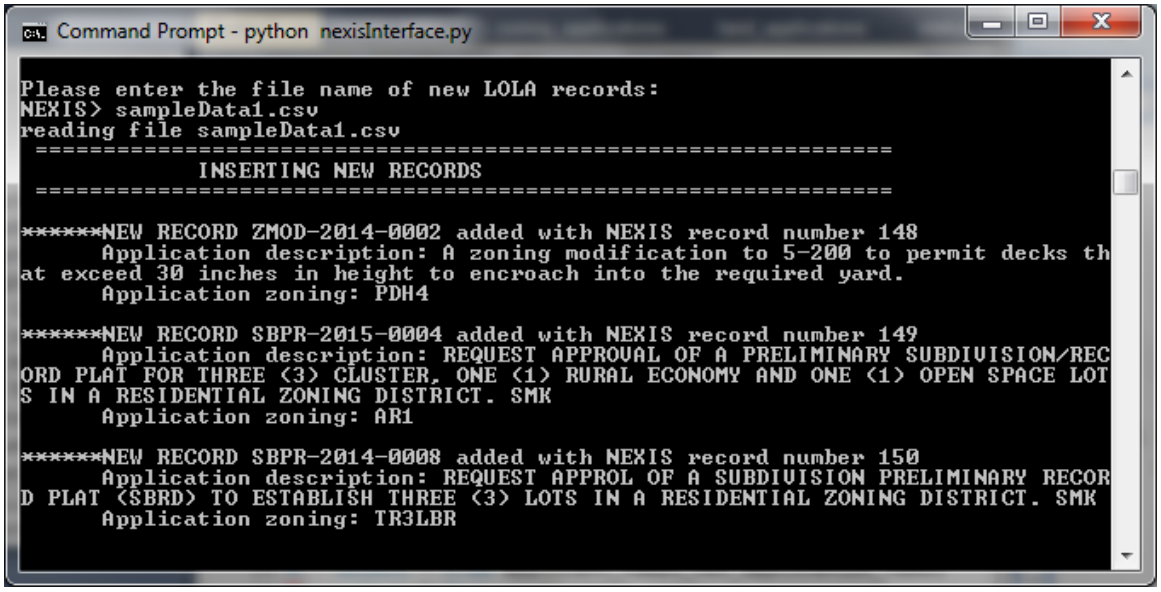
1. To modify the records, choose 1. You will then proceed to the update, menu. Currently only the Loudoun Zoning and Subdivision records can be modified.

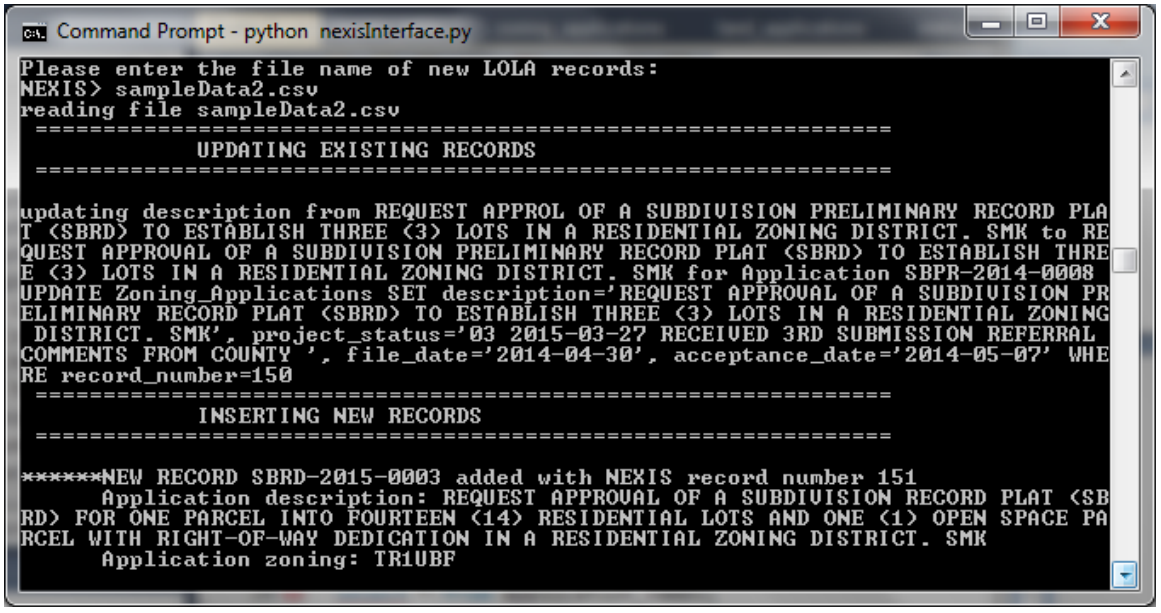


2. Input the file name (or file name and path if the file is not in the same directory as nexisInterface.py) of the csv with modified records.

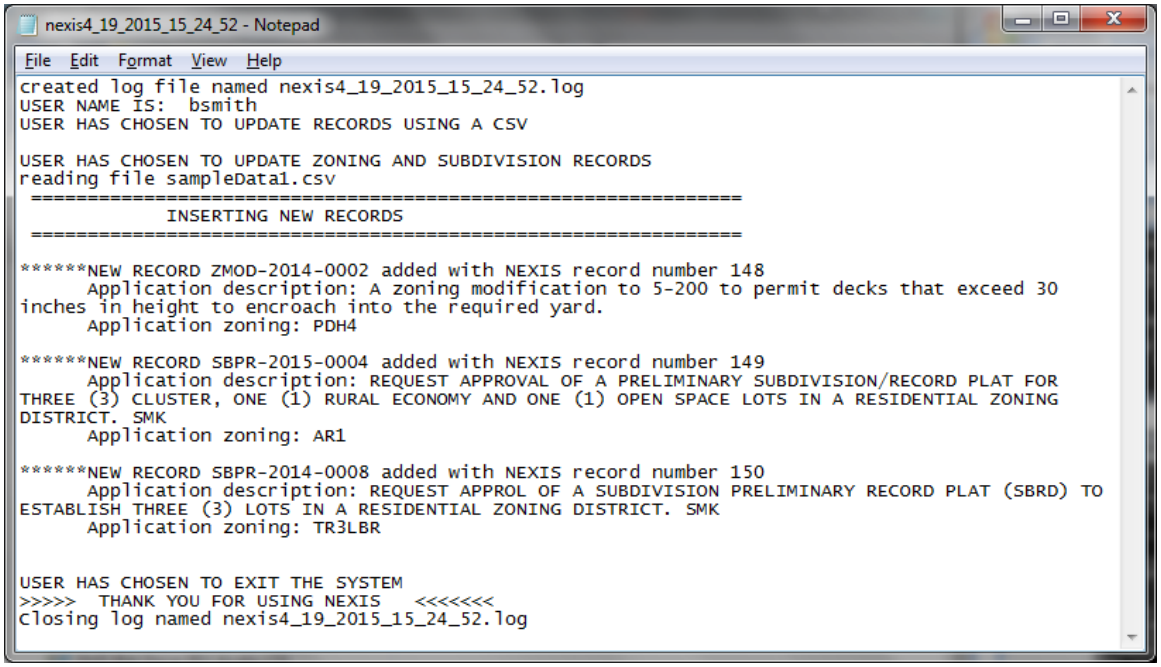


3. The system will search for existing records first and insert new records second. For existing records, it will report where the records are altered. For new records, it will report the Application ID, the NEXIS record number, the application description and the application zoning.

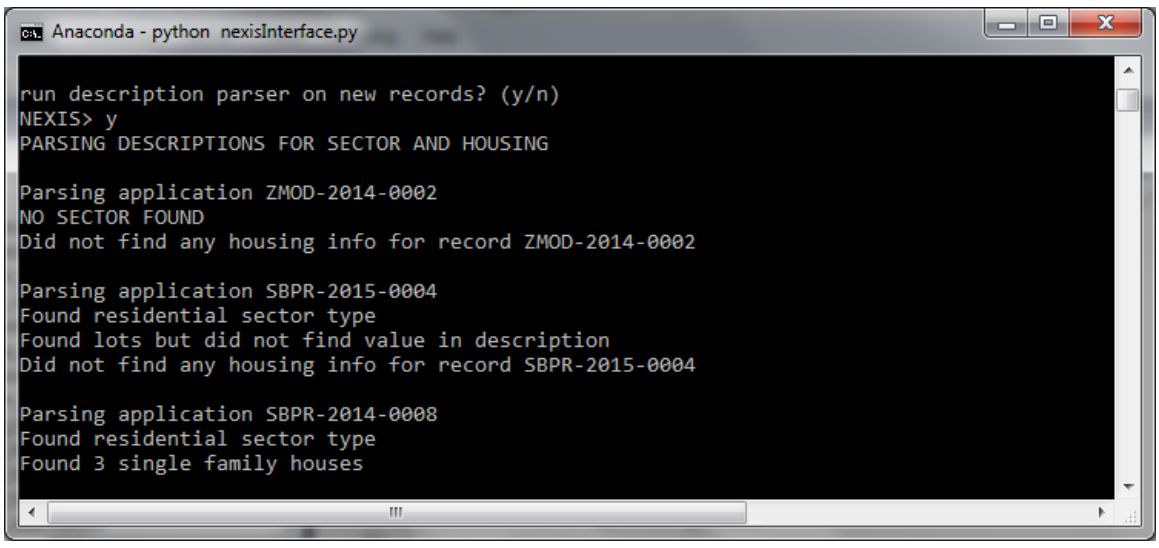




4. The information that is recorded on the screen is also recorded in the log file.

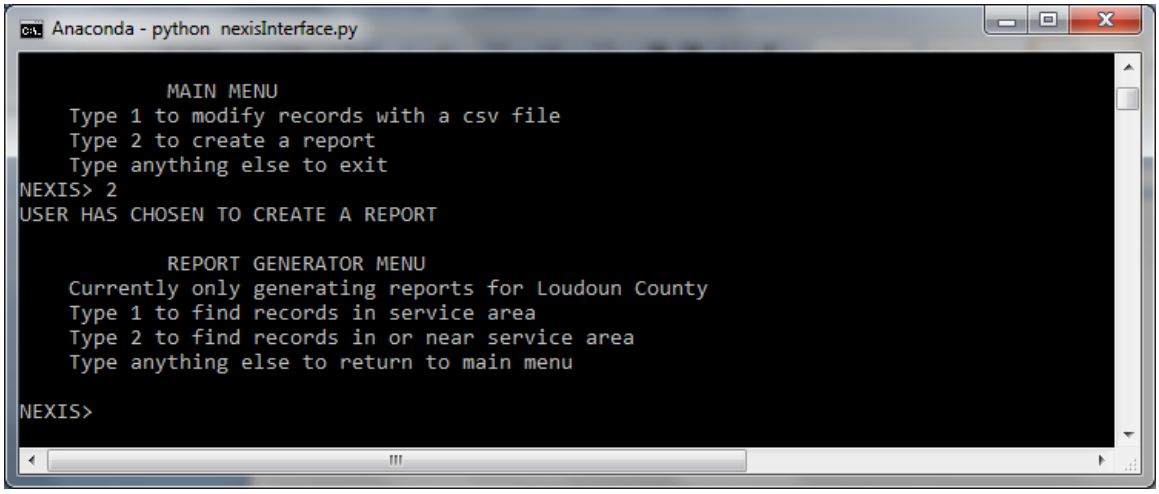


5. After the system completes modifying existing records and inserting new records, the user is given the option to run the description parser on new records. The description parser will attempt to identify the sector for each record and identify any housing units. For details on description parser rules, see descParser.py in Section 5.



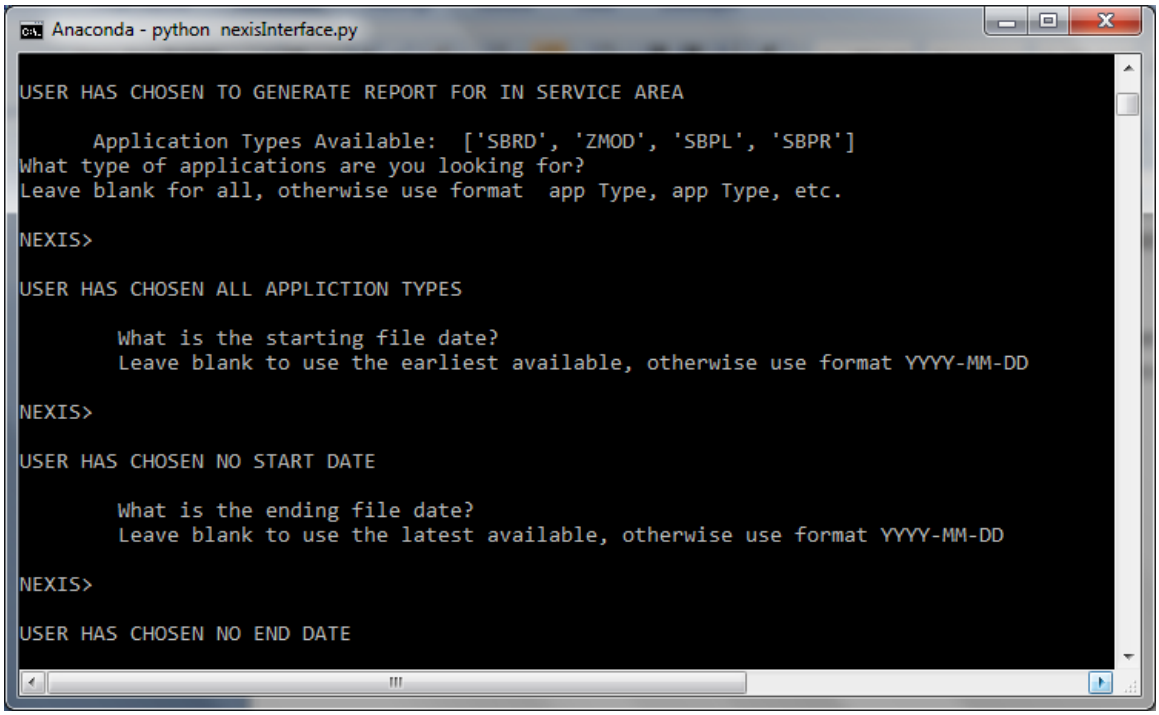
Creating Reports

1. To create a report, choose 2 from the main menu.

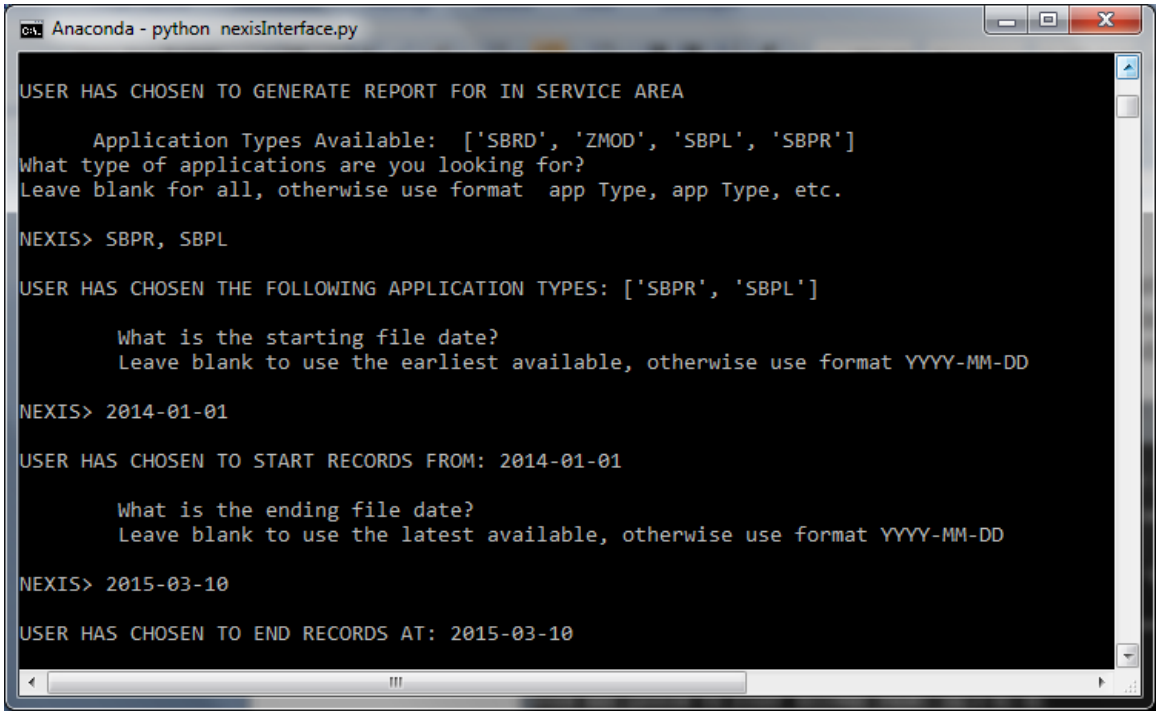


2. The user can choose to determine whether the records are in NOVEC's service area (1) or whether the records are in or near the service area (2).

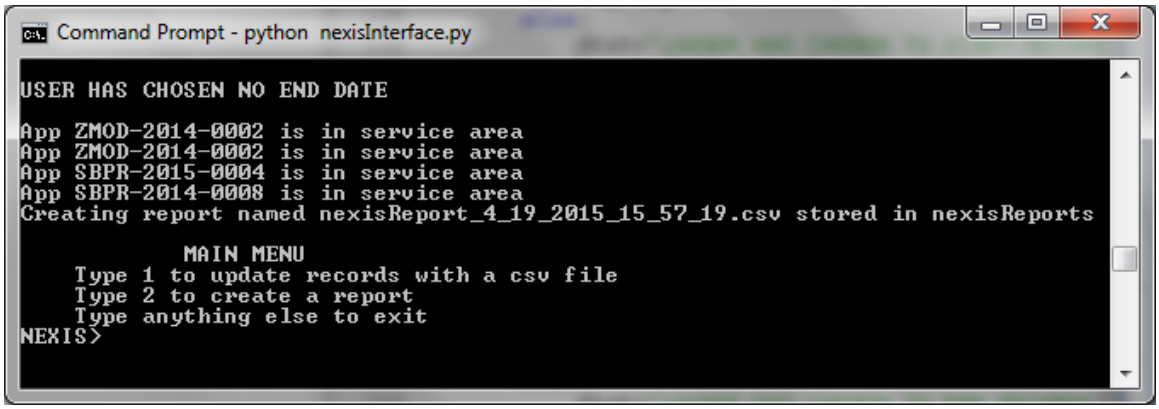
3. Both reports are customizable as to application type and starting file date and ending file date. To search for all, leave that field blank.



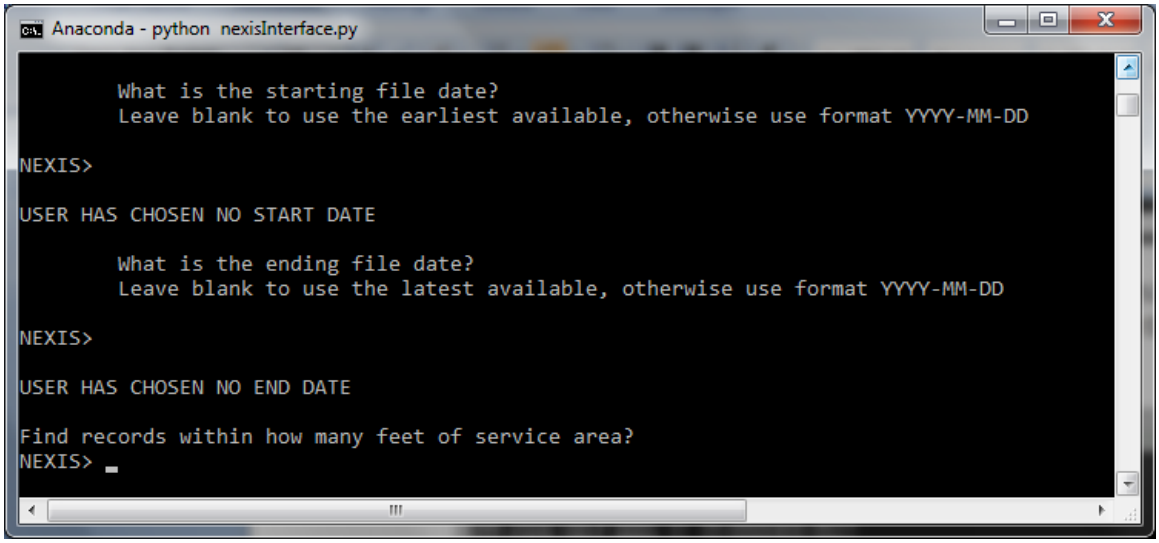
4. To specify application types, input in the form *application type, application type, application type*. To specify a date, input in the form YYYY-MM-DD (for example, 2015-05-08 for May 8, 2015)



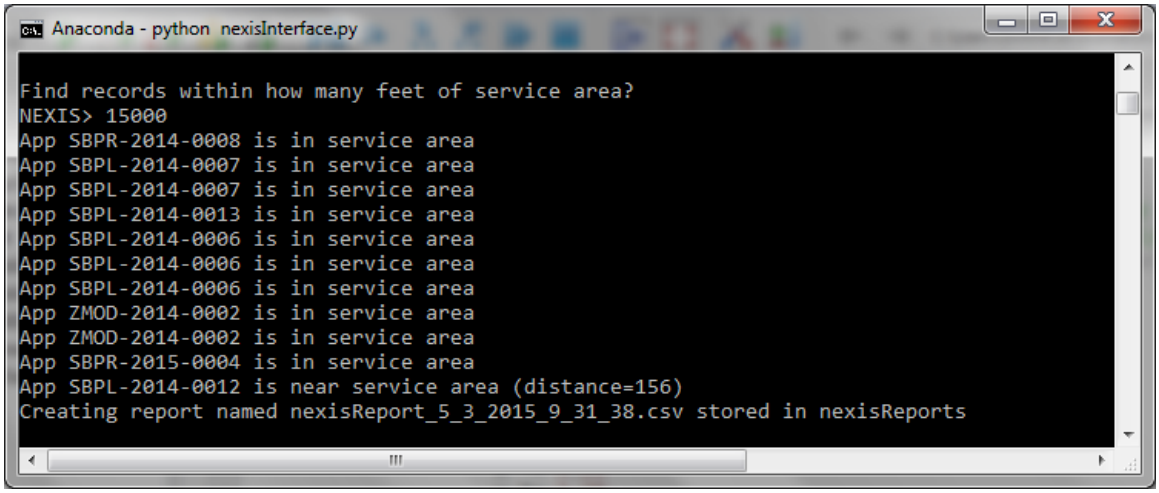
5. For determining which records are in the service area, NEXIS will report to the screen any records that it found and write those records into a report.



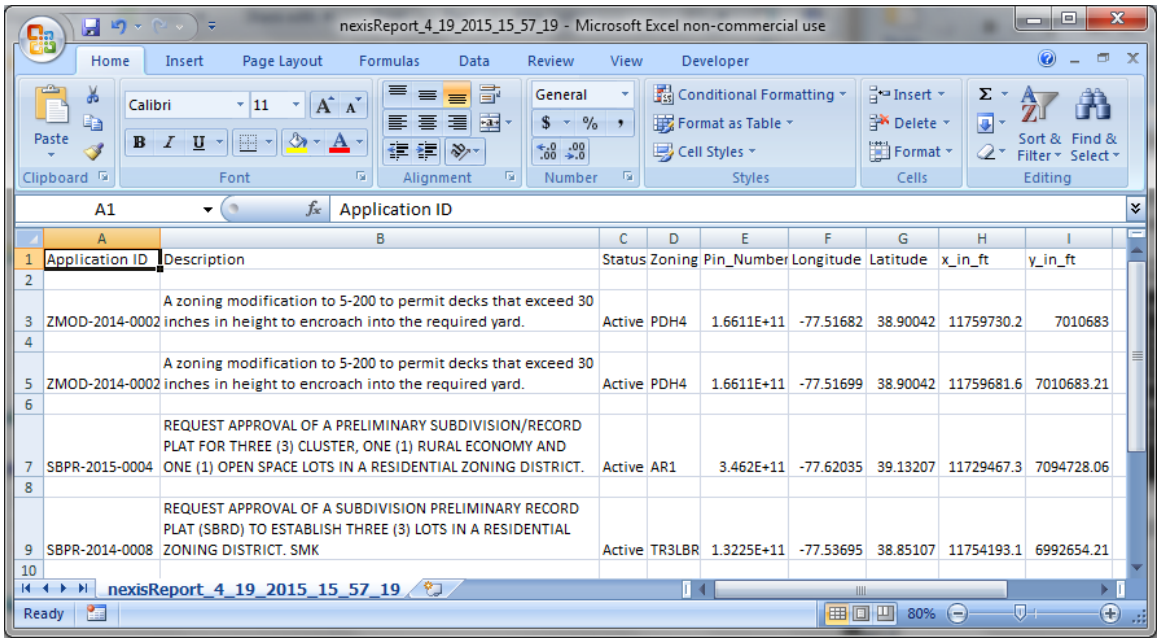
6. For determining which records are in the service area and near the service area, specify the desired proximity (in feet) to the service area border.



7. NEXIS will first determine which records are within the service area. For the remaining records, it will determine whether the records are within the specified proximity threshold. It will write all identified records to a report.

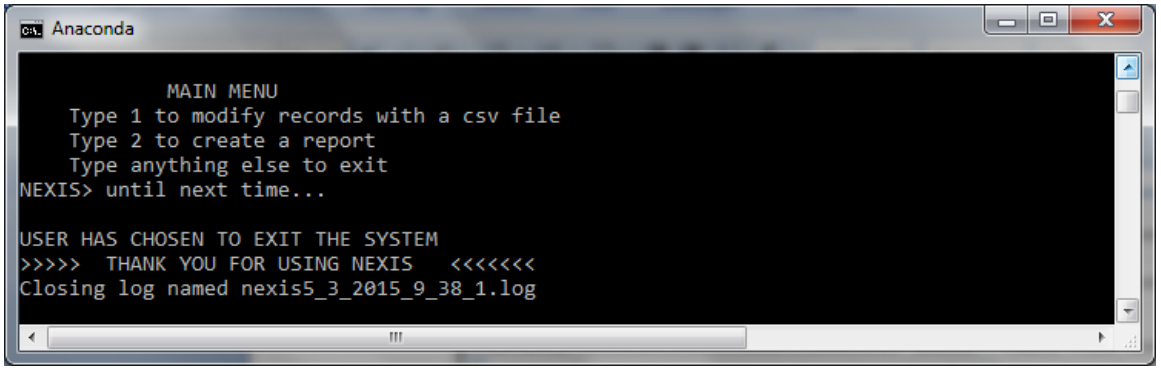


8. The report file will be written in to the *nexisReports* folder in the NEXIS directory. The report will be written to a file named "nexisReports" followed by the date and time and ".csv"



	A	B	C	D	E	F	G	H	I
1	Application ID	Description	Status	Zoning	Pin_Number	Longitude	Latitude	x_in_ft	y_in_ft
2									
3	ZMOD-2014-0002	A zoning modification to 5-200 to permit decks that exceed 30 inches in height to encroach into the required yard.	Active	PDH4	1.6611E+11	-77.51682	38.90042	11759730.2	7010683
4									
5	ZMOD-2014-0002	A zoning modification to 5-200 to permit decks that exceed 30 inches in height to encroach into the required yard.	Active	PDH4	1.6611E+11	-77.51699	38.90042	11759681.6	7010683.21
6									
7	SBPR-2015-0004	REQUEST APPROVAL OF A PRELIMINARY SUBDIVISION/RECORD PLAT FOR THREE (3) CLUSTER, ONE (1) RURAL ECONOMY AND ONE (1) OPEN SPACE LOTS IN A RESIDENTIAL ZONING DISTRICT.	Active	AR1	3.462E+11	-77.62035	39.13207	11729467.3	7094728.06
8									
9	SBPR-2014-0008	REQUEST APPROVAL OF A SUBDIVISION PRELIMINARY RECORD PLAT (SBRD) TO ESTABLISH THREE (3) LOTS IN A RESIDENTIAL ZONING DISTRICT. SMK	Active	TR3LBR	1.3225E+11	-77.53695	38.85107	11754193.1	6992654.21
10									

9. When an action is complete, the user will return to the main menu and can exit by typing anything other than 1 or 2. NEXIS will close the log and remind the user the name of the log file.



Input Formats

Loudoun Zoning and Subdivision records

The Loudoun Zoning and Subdivision records must be in csv format. The information must be in the following order: Application ID, Status, Application Name, Description, Project Status, Applicant Name 1, Applicant Name 2, File Data(DD/MM/YYYY), Acceptance Date(DD/MM/YYYY), Completion Date(DD/MM/YYYY), Existing Zoning, Pin Number, Tax Map, Latitude, Longitude. This file should not include a header row.

Service Area

The folder containing the shape file showing NOVEC's service area must be included to determine reports. To update the service area, please refer to section 2. Installation Instructions.

Using NEXIS Source Code

nexisInterface.py

nexisInterface controls the flow of actions for NEXIS. If new options to update records are added or new reports are generated, the user would need to update these menus to reflect the additional functions

logger.py

logger is the controls for creating, reading and writing to the log file. If the user wanted to change those functions, this is where they would do it.

records.py

records contains the records as objects. If new record types are added, the user can create them here. For fields that may contain multiple records, the user may want to make extra functions. For example, a zoning application may have multiple applicant names, so a list of applicant names is created. The zoning record may have multiple pins with associated tax map, latitude and longitude, so a dictionary is created to store these pins.

nexisDButils.py

nexisDButils contains all interactions with the NEXIS database. This shows examples of selecting various records, updating records and inserting records.

lolaUpdate.py

lolaUpdate updates the Loudoun Zoning and Subdivision records. It checks the new records against the existing records and then inserts new records. Functions included:

updateLolaRecords: controls the main flow through the update, sorts records from csv into existing and new records.

readNewDataFile: reads the csv file and returns lola records.

checkForNewApplicationTypes (or StatusTypes or ZoningTypes): checks the new records type verses the existing types in the database. Currently, new types are added to the database. In the future, if the types are known and static, this can be used to validate incoming data. If a type is not in the existing set, an error can be thrown to signal invalid data.

findGeoUpdates (or StatusUpdates or AppInfoUpdates or AppNameUpdates): checks whether the information is the same for a record in the csv compared to its record in the database.

runDescriptionParser: runs the description parser on new records.

adjustDateFormat: If valid date is received, converts date from mm/dd/yyyy format to yyyy-mm-dd. If no date is received, returns an empty string. If an invalid date is received, an exception is raised.

rowStripper: If a string is received, it trims spaces from the beginning and ending of the string. Otherwise, an empty string is returned.

descriptionCleaner: Trims extra spaces from descriptions.

loudounReports.py

loudounReports creates reports by checking whether records are in NOVEC's service area in Loudoun or near NOVEC's service area. Functions included:

inServiceAreaReportGenerator: controls flow to create reports of records in NOVEC's service area.

inAndNearServiceAreaReportGenerator: controls flow to create reports of records in and near NOVEC's service area.

getRecordsByAppAndDate: asks the user if they would like a subset of records by application type, starting file date and ending file date. It returns all applications according to the user's request.

findRecordsInServiceArea: determines which records are in the service area.

findRecordsInAndNearServiceArea: determines which records are in the service area. It determines for the remaining records which records are near the service area.

writeReport: writes to a csv the records that were sent to it.

validateAppChoice (or Date): checks to make sure the user inputted valid data.

descParser.py

descParser checks an applications description to attempt to determine its sector or the number of housing units.

findSectorType: checks if any of the following words are in the description: residential, commercial, industrial, mixed. If more than one is found, it is considered mixed.

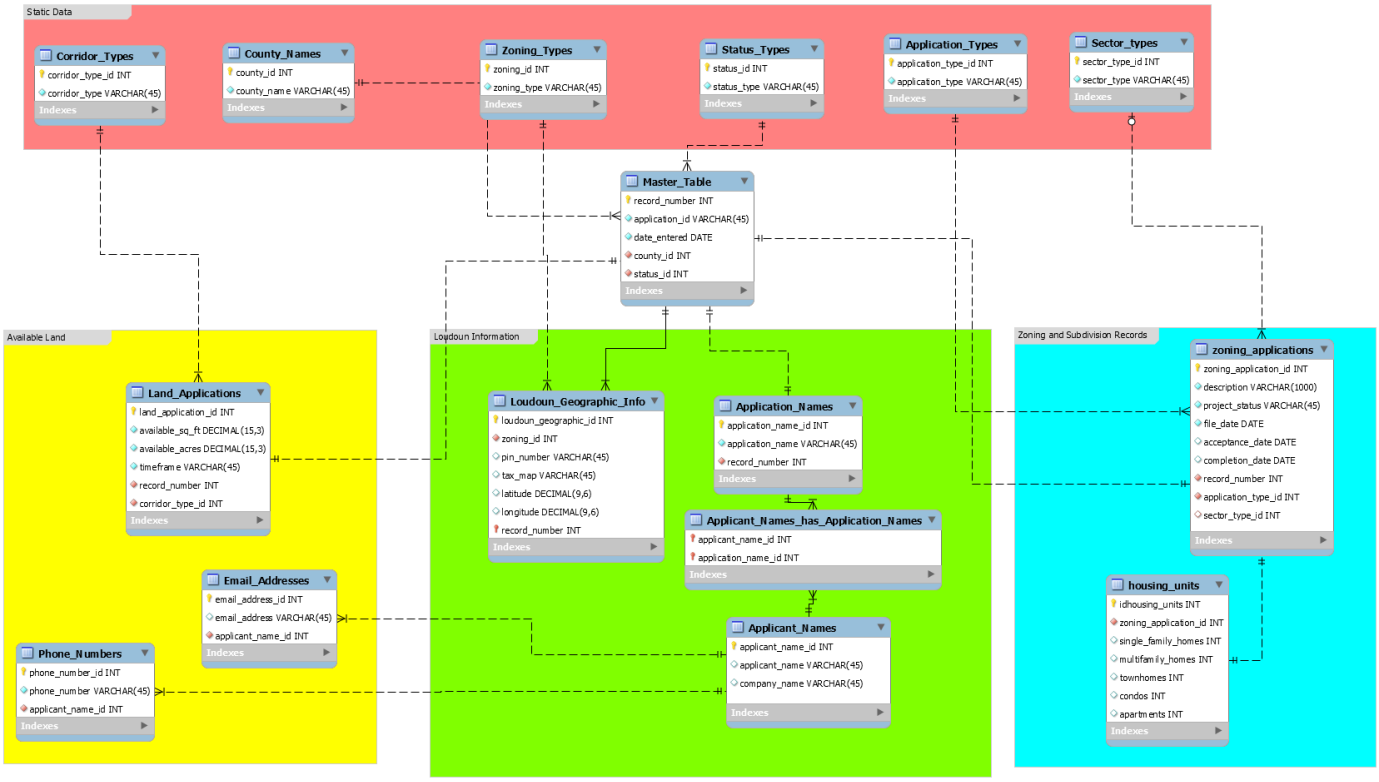
findHousingUnits: attempts to determine from the description the number of housing units for an application. If a specific keyword is found, it checks if there is a number before that keyword (or a number in parenthesis or brackets). The keywords are listed below:

Housing Unit	Keywords
Single Family Homes	sfh, single family home, single family house, lots
Multifamily Homes	mfh, multi family home, multifamily home, multifamily house, multi family house
Townhomes	th, townhome, townhouse, town home, town house
Condominiums	condo, condos, condominium, condominiums
Apartments	apt, apartment

Using NEXIS Database

Querying records

To query records from the database, identify the information required. For tables that are connected, identify the id field that is in common between the two tables.



Deleting records

To delete records from the database, delete records from tables in the following order:

1. applicant_names_has_application_names
2. applicant_names
3. application_names
4. housing_units
5. zoning_applications
6. loudoun_geographic_info
7. master_records

Known Issues

1. If a new application, zoning, or status type is found, the updater inserts that type and highlights that in the log.

2. For lolaUpdate, if the csv has a different applicant name or pin number, it adds the new applicant name or pin number (with tax map, latitude and longitude). It does not remove any existing applicant names or pins.