SAND2022-6419C

Sandia National Laboratories

Exceptional service in the national interest

Connecting Python to Vitech's GENESYS

Alex Bencoe

R&D, System Engineer, 02496

Integrate22 DE Symposium

San Antonio, TX

June 6th – 9th

integrate22™

The Digital Engineering Symposium

Connecting Python to Vitech's GENESYS

Alex Bencoe
R&D, System Engineer

Vitech
A Zuken Company

Agenda

- Suggested resources and references
- Python installation
- Configuration guide
- Testing Pythonnet functionality
- Connecting to server
- Extracting data
- Example vignettes of tool usage

Resources and References

Needed Resources

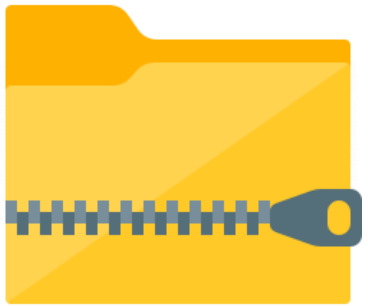
- Active license to Vitech's MBSE Tool GENESYS
  - If you do not have an active license proceed to for install instructions:
    - Corporate Hyperlink
- Anaconda3
  - Python 3.8 vs New releases of Python
- Pythonnet Installation
- PysimpleGui Installation
- Windows OS (example is running Windows 10)

References

- C:\Program Files (x86)\Vitech\GENESYS 2021 Collaborative Edition\Documentation
  - API Getting Started Guide
  - API.CHM
- https://pythonnet.github.io/
- https://pysimplegui.readthedocs.io/en/latest/
- Corporate Repoitory

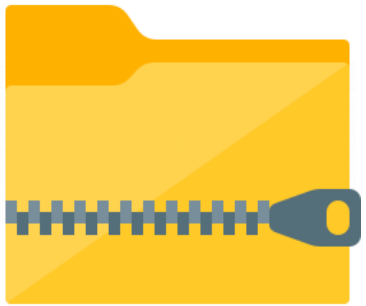
Reference Files

Required DLLs, Configuration File, and .condarc



licenseDLLs_configFile_Condarc.zip

Python Scripts



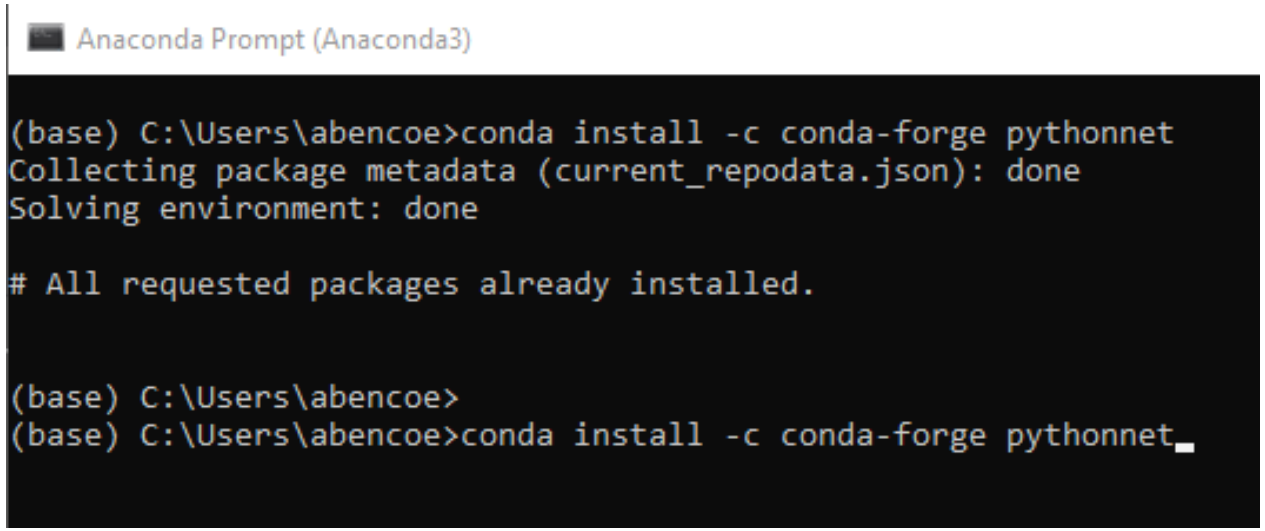
scripts.zip

Files can also be located at: corporate repository

Python Library Installations

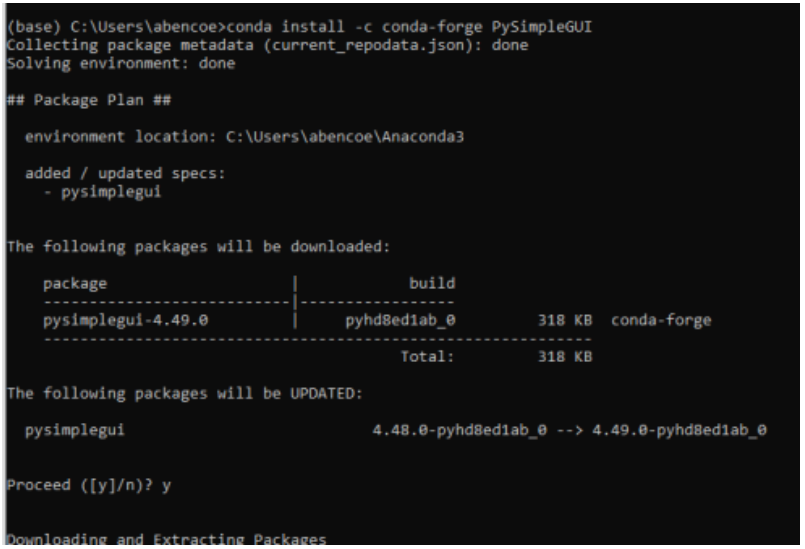
Pythonnet Installation

1. Type Anaconda Prompt in your Window's Start menu
2. Once the prompt appears enter *conda install -c conda-forge pythonnet* into the command line
3. If a proxy error occurs:
    1. Copy *.condarc* file from LicenseDLLs_configFile_condarc.zip found on Reference Files slide
    2. Paste the *.condarc* file in your home directory c:\users\<username>\
    3. Close Anaconda Prompt and restart instructions at step 1.



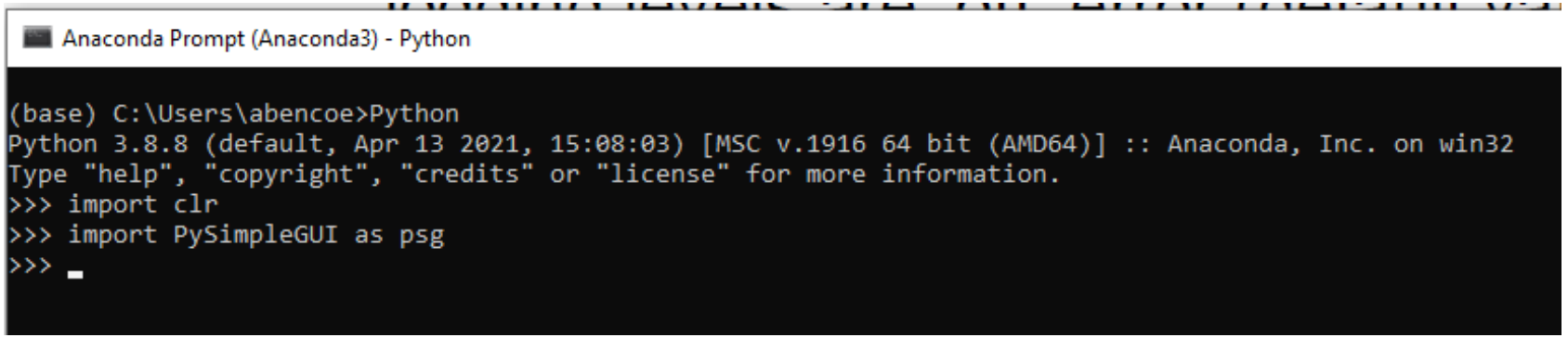
PySimpleGUI Installation

1. Open Anaconda Prompt or Use the current Anaconda Prompt
2. Enter *conda install -c conda-forge PySimpleGUI* into the command line
3. If a proxy error occurs:
    1. Copy *.condarc* file from LicenseDLLs_configFile_condarc.zip found on Reference Files slide
    2. Paste the *.condarc* file in your home directory c:\users\<username>\
    3. Close Anaconda Prompt and restart instructions at step 1.



Validate Pythonnet and PySimpleGUI libraries can be imported

1. Open Anaconda Prompt or use current promp
2. Enter *import clr* into the command line
3. Enter *import PySimpleGUI as psg* into the command line



- If an error occurs, double check you have properly installed the libraries and your syntax matches exactly

Configuring Python to GENESYS Connection

UNCLASSIFIED UNLIMITED RELEASE

Configuring GENESYS License

1. With Windows Explorer, navigate to where your python script (.py) file will exist that will connect to GENESYS server
2. Copy the below DLLs contained within the *LicenseDLLs_configFile_condarc.zip* found on Reference Files slide
   - *hasp_net_windows*
   - *hasp_windows_82194*
   - *hasp_windows_x64_82194*
   - *apidsp_windows*
   - *apidsp_windows_x64*
3. Paste the five DLLs at the location found in step 1.

Configuring GENESYS License

1. Navigate to where your PYTHON.exe exists within your Anaconda install
   - Most likely: C:\Users\<username>\Anaconda3\
2. Copy *python.exe.config* from LicenseDLLs_configFile_condarc.zip found on Reference Files slide
3. Paste to the location found in step 1.

Testing pythonnet Functionality

Using Pythonnet Library

1. Open Anaconda3 Navigator application and Launch Spyder IDE
2. Copy *test_clr.py* from *Scripts.zip* found in Reference Files slide
3. Paste *test_clr.py* to your current working python directory where the GENESYS license DLLs exist.
4. Open *test_clr.py* file from Spyder
5. Run *test_clr.py* in Spyder by clicking green run button 

Using Pythonnet Library

Sample expected output from *test_clr.py*

```
# ============================================================================
# the purpose of this script is to validate that you can properly run the
# pythonnet library and use existing windows DLLs
# ============================================================================
# Importing libraries
# ============================================================================
import clr
import sys
# ============================================================================

#-------------------------------------------------------------
# Calling in System and System.IO .NET API libraries
#-------------------------------------------------------------
# ============================================================================
clr.AddReference('System')
clr.AddReference('System.IO')

from System import *
from System.IO import *
# ============================================================================
#-------------------------------------------------------------



#-------------------------------------------------------------
#Testing to make sure CLR library is appropriately working
# -------------------------------------------------------------
# ============================================================================
my_mach_name = Environment.MachineName
my_OS_version = Environment.OSVersion
my_UN = Environment.UserName
cd = Directory.GetCurrentDirectory()
#
print('My Machine is called: {}'.format(my_mach_name))
print('My OS Version is: {}'.format(my_OS_version))
print('My User Name is: {}'.format(my_UN))
print('Current Directory: {}'.format(cd))
# ============================================================================
#-------------------------------------------------------------
```

```
Usage

Here you can get help of any obj
Help can also be shown automat

Console 1/A

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (A
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/abencoe/source/repos/clr_test/test_clr.py
  File "C:\Users\abencoe\source\repos\clr_test\test_clr.py", line 2
    my_mach_name = Environment.MachineName
    ^
IndentationError: unexpected indent


In [2]: runfile('C:/Users/abencoe/source/repos/clr_test/test_clr.py
My Machine is called: S1061371
My OS Version is: Microsoft Windows NT 10.0.19042.0
My User Name is: abencoe
Current Directory: C:\Users\abencoe\source\repos\clr_test

In [3]:
```

Connection to GENESYS Server

Connecting Python to GENESYS

1. Open Anaconda3 Navigator application and Launch Spyder IDE
2. Copy GENESYS_connection.py from Scripts.zip found in Reference Files slide
3. Paste GENESYS_connection.py to your current working python directory where the GENESYS license DLLs exist.
4. Open GENESYS_connection.py file from Spyder
5. Run GENESYS_connection.py in Spyder by clicking green run button 

Connecting Python to GENESYS

1. A pop-up window will appear
2. Select a GENESYS project
3. Click “OK”
4. You are now connected to a GENESYS project through the project variable in Spyder
5. Once you no longer need to have an active connection to the GENESYS server enter Repo.Logout() into Spyder command line

Do not forget to release API license by typing Repo.Logout() into Spyder command line

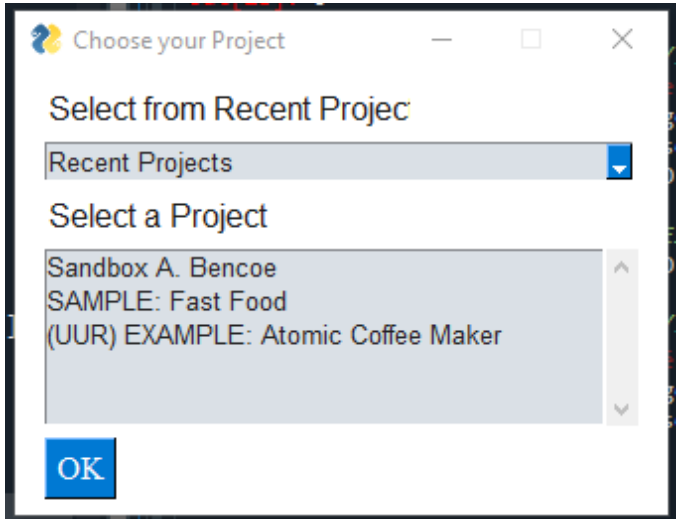
Connecting Python to GENESYS

1. The script GENESYS_connection.py makes use of PySimpleGUI to allow for the user to select a GENESYS project
2. Once the script executes, you now have the ability to make any GENESYS API calls from the installed Vitech Libraries.
3. Refer to API.CHM on References slide namespace Methods and Properties
   - A good starting place is to search IProject Interface

```
In [23]: runfile('C:/Users/abencoe/source/repos/clr_test/test_GENESYS_connection.py', wdir='C:/Users/abencoe/source/repos/clr_test')
Reloaded modules: clr, Vitech, Vitech.Genesys, Vitech.Genesys.Client, Vitech.Genesys.Common
Connecting to GENESYS Server (as03genesysnt  ...)
Successfully coneected to GENESYS server as GENESYS user Allex Bencoe with session ID 52e7dca1-e8ad-4b14-a19f-177a795e07fc.
Connecting to GENESYS project (Sandbox A. Bencoe)

In [24]: project.Name
Out[24]: 'Sandbox A. Bencoe'

In [25]:
```



Extracting Data from GENESYS

UNCLASSIFIED UNLIMITED RELEASE

Retrieve GENESYS Data

All Entity Attributes, Parameters, and Relationships (+relationship attributes)

1. Copy *Entity_Data.py* from *Scripts.zip* found in Reference Files slide
2. Paste *Entity_Data.py* to your current working python directory where the GENESYS license DLLs exist.
3. Open *Entity_Data.py* from Spyder
4. Run *Entity_Data.py* from spyder by clicking green Run button 
   - You now have access to GENdata function
5. Enter *data = GENdata(Repo, projName, server, port)* into Spyder command line
   - *Assumes GENESYS_connection.py has previously been executed*
6. Enter *DFsClassAtts = data[0]* into Spyder command line
7. Enter *DFEntityRelationships = data[1]* into Spyder command line
   -

Retrieve GENESYS Data

All Entity Attributes, Parameters, and Relationships (+relationship attributes)

5. Enter *data = GENdata(Repo, projName, server, port)* into Spyder command line
   - *Assumes GENESYS_connection.py has previously been executed*
6. Enter *DFsClassAtts = data[0]* into Spyder command line
7. Enter *DFEntityRelationships = data[1]* into Spyder command line

Retrieve GENESYS Data

All Entity Attributes, Parameters, and Relationships (+relationship attributes)

1. DFsClassAtts: A dictionary of Pandas Data Frames
   - The dictionary keys are all Folder classes that have at least 1 entity
   - The data frames is a table of values of every entity and class attributes along with a parameter dictionary of ever entity parameter
   - Example: DFsClassAtts.keys() returns a list of folders
   - Example: DFsClassAtts[‘Requirement’] returns a dataframe of all requirement entities and their attributes.
   - Example: DFsClassAtts[‘Requirement’][‘Description’] returns a list of all requirement descriptions.
   - The index for the DataFrame are requirement name attributes

Retrieve GENESYS Data

All Entity Attributes, Parameters, and Relationships (+relationship attributes)

2. DFEntityRelationships: A Pandas Data Frame

- The DataFrame contains every entity and relationship in the GENESYS project.
- It also contains relationship attributes if they exist

```
In [20]: data = GENdata(Repo, projName)
Connecting to GENESYS Server (as03genesysnt ...)
Successfully coneected to GENESYS server as GENESYS user Allex Bencoe with session ID 1d6e2972-e2a8-427b-a0cb-7b8227c0e782.
Connecting to GENESYS project (Sandbox A. Bencoe)
Retrieving and storing data from project (Sandbox A. Bencoe).
Data retrieval and storage is complete
Execution Time (HH:MM:SS.sss): 0:01:01.003205
GENESYS user Allex Bencoe with session ID 1d6e2972-e2a8-427b-a0cb-7b8227c0e782 is now logged out of the server.

In [21]: DFsClassAtts = data[0]

In [22]: DFEntityRelationshipos = data[1]

In [23]: DFsClassAtts['Requirement']['Description']
Out[23]:
Be Customized Based On What                 This page should be customized based on what p...
Be Able To Support X                        The system should be able to support (TBD) sim...
Provide Accounting With Accurate Purchase   The system shall provide accounting with accur...
Display Information That Is Customized_001  The system shall display information that is c...
Provide A Search Facility That_001          The system shall provide a search facility tha...
                                                                                       ...
Child 6                                     Example child 6 requirement text.\r\nMany desk...
Child 2                                                    Example child 2 requirement text.
Req parent
req grandchild
req child
Name: Description, Length: 62, dtype: object

In [24]:
```

Using PySimpleGUI to View and Edit GENESYS Entity Attributes

UNCLASSIFIED UNLIMITED RELEASE

View and update Entity Attributes

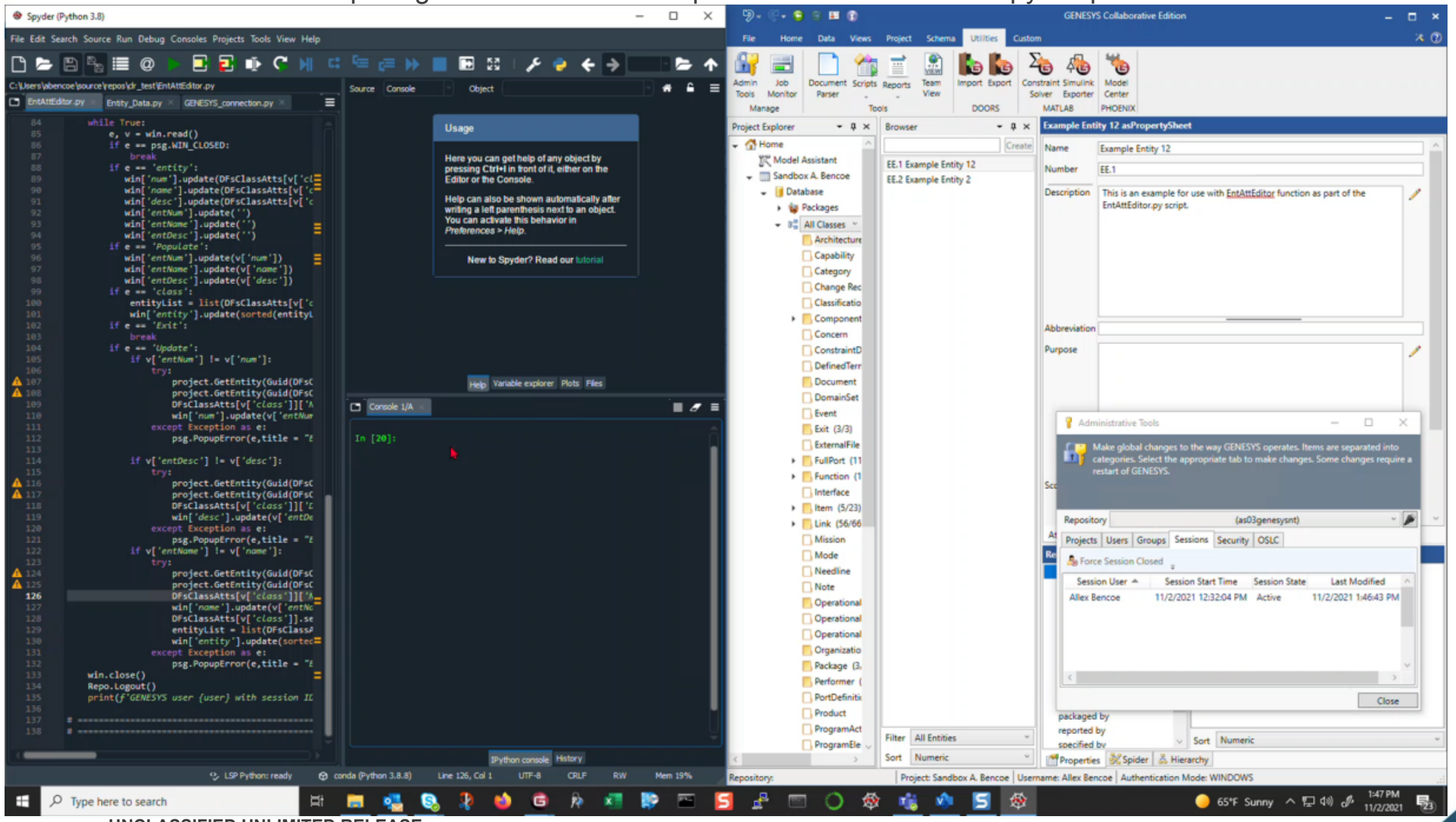
1. Copy *EntAttEditor.py* from *Scripts.zip* found in Reference Files slide
2. Paste *EntAttEditor.py* to your current working python directory where the GENESYS license DLLs exist.
3. Open *EntAttEditor.py* from Spyder
   - DFsClassAtts variable needs to be defined. Refer to Retrieve GENESYS Data slides for defining variable
4. Run *EntAttEditor.py* from Spyder by clicking green Run button 
5. Enter EntAttEditor into Spyder Command Line
6. A pop-up GUI will appear

View and update Entity Attributes

7. Select the class of entities you want to view/edit from the combo box
8. Select the entity you want to view/edit from the list
9. Click “Populate” if you want to edit the attributes
10. Edit the attributes in the bottom right section of the GUI
11. Once you have completed your edits click “Update”
12. The GENESYS project has now been updated with your
13. Click “Exit” or Close the window

View and update Entity Attributes

Example vignette of the view/edit process with EntAttEditor.py script



Updating Entity Attributes and Parameters

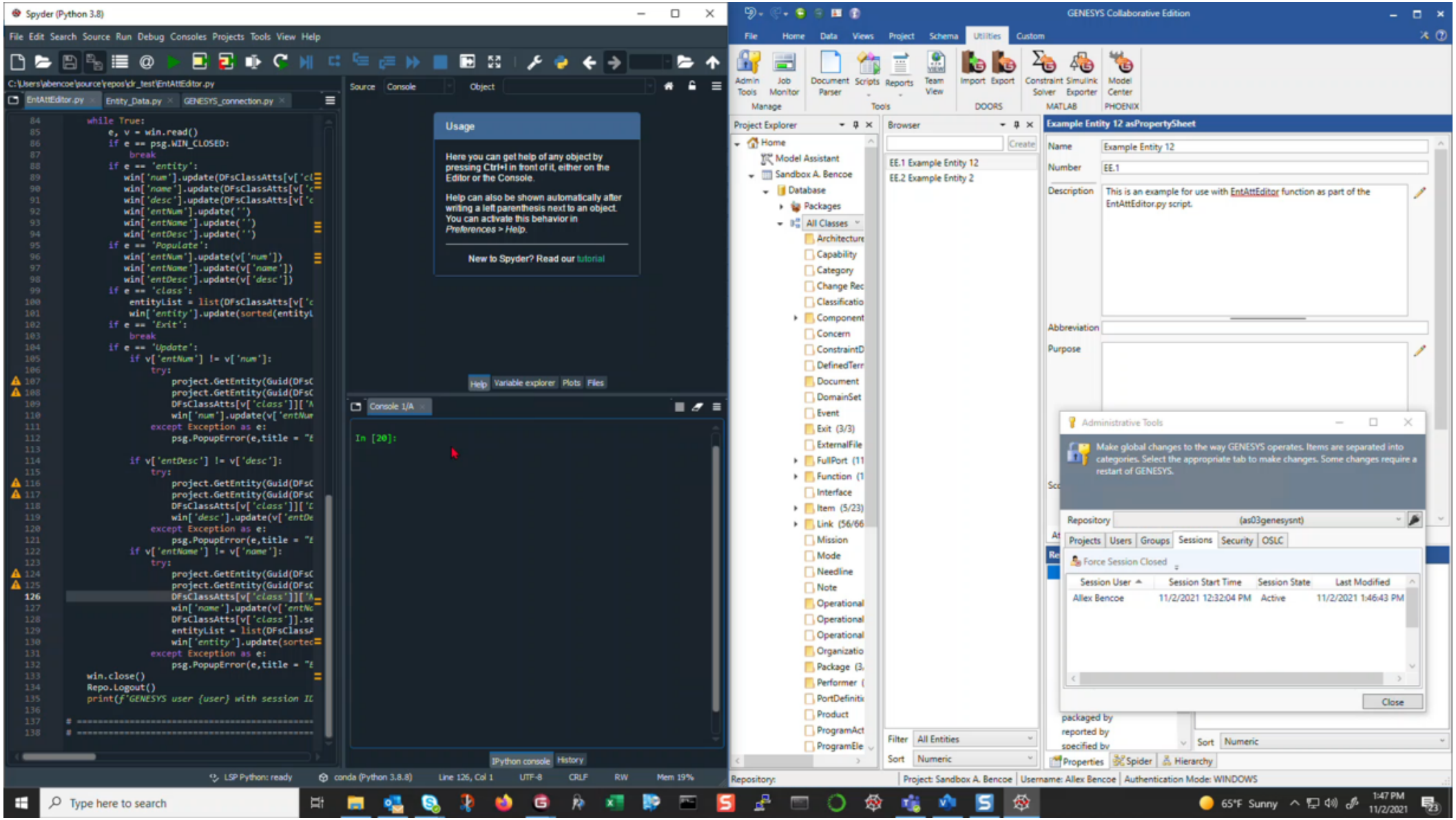
5/16/2022

Update Entity Attribute

1. Run *Entity_Data.py* from Spyder by clicking green Run button 
2. Create entGuid variable using DFsClassAtts
   - i.e. entGuid = DFsClassAtts[*className*][‘ID’][entityName]
3. Create attName variable
   - i.e. attName = ‘name’
4. Create entValue variable for the new attribute value
   - i.e. entValue = ‘Ex. Entity Name 5000’
5. Enter Repo = GENlogin(server, port)
   - Server = “as03genesysnt.srn.sandia.gov”; port = 39101
6. Enter updateEntAtt(Repo, projName, entGuid, attName, entValue) into Spyder command line
7. GENESYS project entity is now updated
8. Enter Repo.Logout() to release API license from server

Update Entity Attribute

Example vignette of the using updateEntAtt function for updating entity attribute



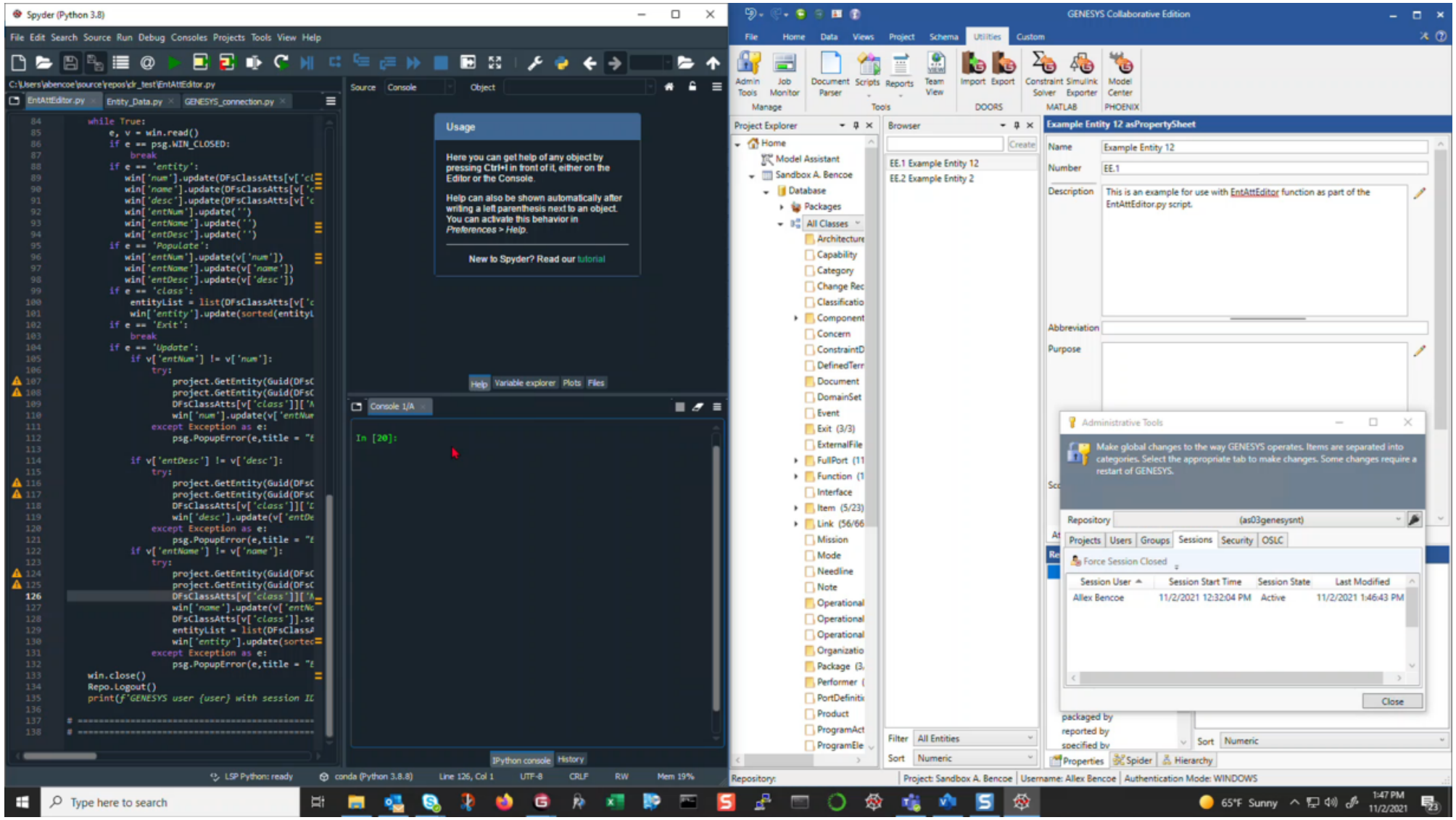
5/16/2022

Update Entity Parameter

1. Run *Entity_Data.py* from Spyder by clicking green Run button 
2. Create entGuid variable using DFsClassAtts
   - i.e. entGuid = DFsClassAtts[*className*][‘ID’][entityName]
3. Create paramName variable
   - For this example, paramName = ‘exampleParam’
4. Create paramValue variable for the new attribute value
   - i.e. paramValue = 15.12345
5. Enter Repo = GENlogin(server, port)
   - Server = “as03genesysnt…”; port = 39101
6. Enter updateDesignEntParam(Repo, projName, entGuid, paramName, paramValue) into Spyder command line
7. GENESYS project entity is now updated
8. Enter Repo.Logout() to release API license from server

Update Entity Parameter

Example vignette of the using updateDesignEntParam function for updating entity Design parameter



Questions?

integrate22™