



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 6<sup>th</sup> – 9<sup>th</sup>





# integrate22<sup>TM</sup>

The Digital Engineering Symposium

## Connecting Python to Vitech's GENESYS

Alex Bencoe  
*R&D, System Engineer*





# 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)



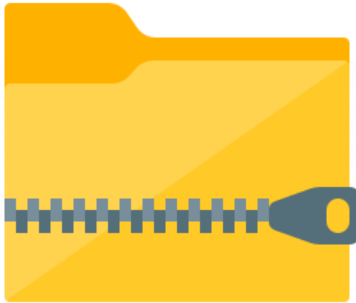
## References

- 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 Repository



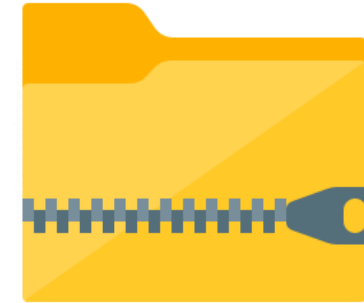
## Reference Files

**Required DLLs,  
Configuration File,  
and .condarc**



licenseDLLs\_configFile\_Conduc.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.

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\abencoe>conda install -c conda-forge pythonnet
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

(base) C:\Users\abencoe>
(base) C:\Users\abencoe>conda install -c conda-forge pythonnet_
```



# 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.

```
(base) C:\Users\abencoe>conda install -c conda-forge PySimpleGUI
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\abencoe\Anaconda3

added / updated specs:
- pysimplegui

The following packages will be downloaded:

package | build | size | source
-----|-----|-----|-----
pysimplegui-4.49.0 | pyhd8ed1ab_0 | 318 KB | conda-forge
-----|-----|-----|-----
Total: | | 318 KB |

The following packages will be UPDATED:

pysimplegui | 4.48.0-pyhd8ed1ab_0 --> 4.49.0-pyhd8ed1ab_0

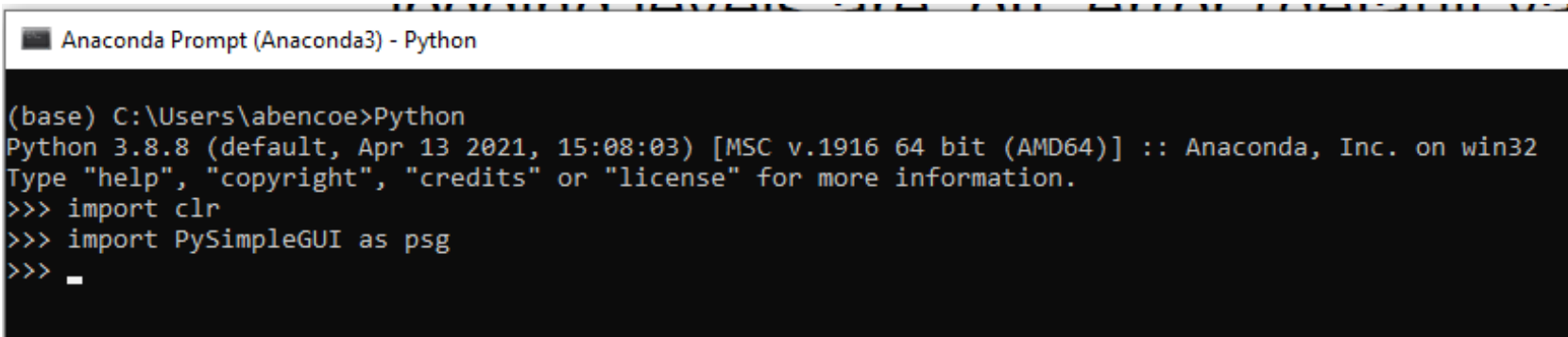
Proceed ([y]/n)? y

Downloading and Extracting Packages
```



# Validate Pythonnet and PySimpleGUI libraries can be imported

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



```
Anaconda Prompt (Anaconda3) - Python
(base) C:\Users\abencoe>Python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import clr
>>> import PySimpleGUI as psg
>>> _
```

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

# Configuring Python to GENESYS Connection



## 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

5  
/  
1  
6  
/  
2  
0  
2  
2



## 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*

```
1 # =====
2 # the purpose of this script is to validate that you can properly run the
3 # pythonnet library and use existing windows DLLs
4 # =====
5 # Importing libraries
6 # =====
7 import clr
8 import sys
9 # =====
10
11 #-----
12 # Calling in System and System.IO .NET API libraries
13 #-----
14 # =====
15 clr.AddReference('System')
16 clr.AddReference('System.IO')
17
18 from System import *
19 from System.IO import *
20 # =====
21 #-----
22
23
24 #-----
25 #Testing to make sure CLR library is appropriately working
26 # -----
27 # =====
28 my_mach_name = Environment.MachineName
29 my_OS_version = Environment.OSVersion
30 my_UN = Environment.UserName
31 cd = Directory.GetCurrentDirectory()
32 #
33 print('My Machine is called: {}'.format(my_mach_name))
34 print('My OS Version is: {}'.format(my_OS_version))
35 print('My User Name is: {}'.format(my_UN))
36 print('Current Directory: {}'.format(cd))
37 # =====
38 #-----
```

Usage

Here you can get help of any object  
Help can also be shown automatically

Console 1/A

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] 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', line 28, namespace={'my\_mach\_name': Environment.MachineName})

IndentationError: unexpected indent

In [2]: runfile('C:/Users/abencoe/source/repos/clr\_test/test\_clr.py', line 33, namespace={'my\_mach\_name': Environment.MachineName})

My Machine is called: 51061371  
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

5  
/  
1  
6  
/  
2  
0  
2  
2



## 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**



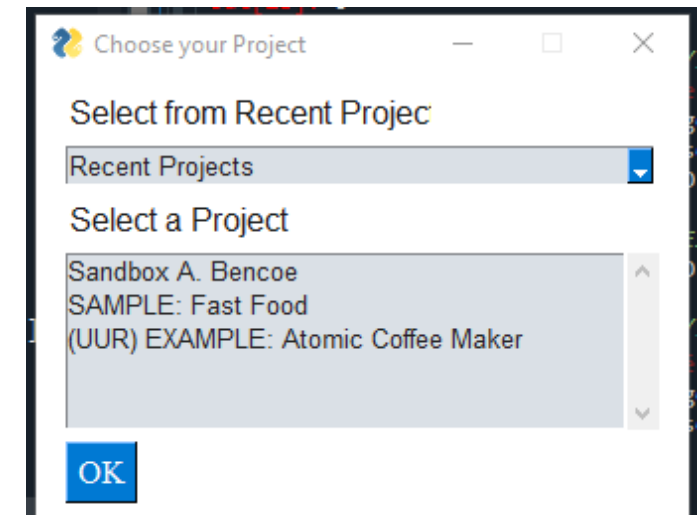
# Connecting Python to GENESYS

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 connected 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



# 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 every 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

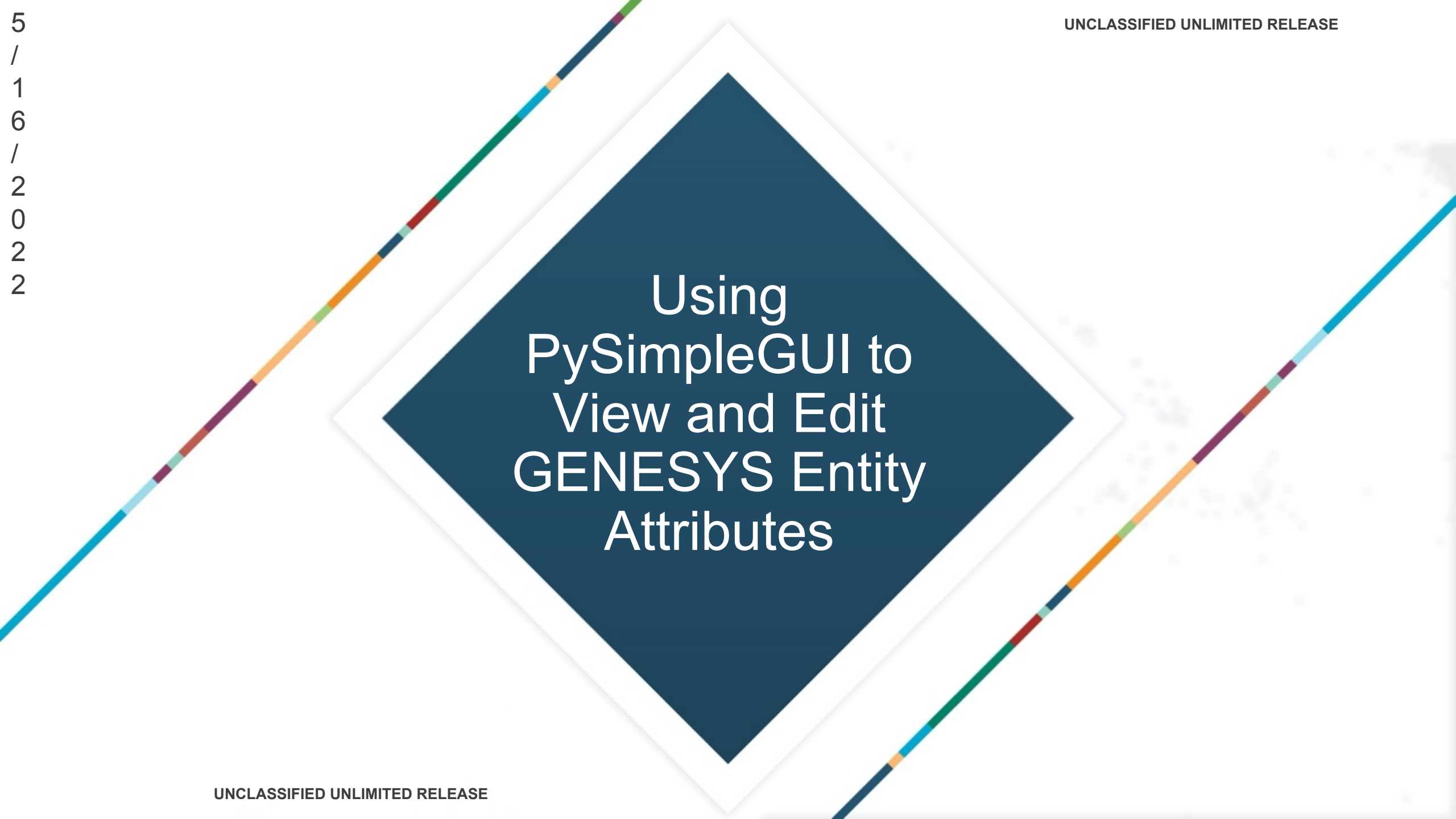
```
In [20]: data = GENdata(Repo, projName)
Connecting to GENESYS Server (as03genesysnt ...)
Successfully connected to GENESYS server as GENESYS user Alex Bencoe with session ID 1d6e2972-e2a8-427b-a0cb-7b8227c0e782.
Connecting to GENESYS project (Sandbox A. Bencoe)
Retrieving and storing data from project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

In [21]: DFsClassAtts = data[0]

In [22]: DFEntityRelationships = 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



## 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

UNCLASSIFIED UNLIMITED RELEASE

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

The screenshot displays a development environment with three main components:

- Spyder (Python 3.8):** The left pane shows the `EntAttEditor.py` script. The script is a Python program that interacts with the GENESYS database to view and update entity attributes. It includes a `while True:` loop that processes user input to update entity details like name, description, and number. The right pane shows a `Usage` dialog box with instructions on how to use the script.
- GENESYS Collaborative Edition:** The middle pane shows the GENESYS interface. The `Project Explorer` on the left lists the project structure, including `Model Assistant`, `Sandbox A. Bencoe`, and `Database`. The `Browser` on the right shows the `Example Entity 12` and `Example Entity 2`. The `Example Entity 12 asPropertySheet` dialog box is open, showing the entity's attributes: `Name` (Example Entity 12), `Number` (EE.1), `Description` (This is an example for use with EntAttEditor function as part of the EntAttEditor.py script), `Abbreviation`, and `Purpose`.
- Administrative Tools:** The bottom right pane shows the `Administrative Tools` dialog box. It contains a table of session information:

Session User	Session Start Time	Session State	Last Modified
Alex Bencoe	11/2/2021 12:32:04 PM	Active	11/2/2021 1:46:43 PM


The status bar at the bottom of the GENESYS interface shows the repository as `as03genesysnt`, the project as `Sandbox A. Bencoe`, the username as `Alex Bencoe`, and the authentication mode as `WINDOWS`.

UNCLASSIFIED UNLIMITED RELEASE



# Updating Entity Attributes and Parameters

## 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

UNCLASSIFIED UNLIMITED RELEASE

Example vignette of the using updateEntAtt function for updating entity attribute

The screenshot displays the GENESYS Collaborative Edition interface, which is divided into several panes. The left pane shows a Python script in the 'EntAttEditor.py' file, which is part of a larger project. The script is a while loop that handles various commands like 'entity', 'Populate', 'Exit', and 'Update'. The 'Update' command is currently selected, and it calls the 'updateEntAtt' function to update the 'num' attribute of an entity. The right pane shows the 'Example Entity 12 asPropertySheet' dialog, which contains fields for Name, Number, Description, Abbreviation, and Purpose. The 'Number' field is set to 'EE.1'. The bottom pane shows the 'Console' window, which displays the output of the script, including the command 'In [20]:'. The top pane shows the 'Project Explorer' and 'Browser' panes, which list the project's structure and the entities being managed.

Usage

Here you can get help of any object by pressing Ctrl+H in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

Console 1/A

In [20]:

Repository: (as03genesysnt)

Session User	Session Start Time	Session State	Last Modified
Alex Bencoe	11/2/2021 12:32:04 PM	Active	11/2/2021 1:46:43 PM


packaged by reported by specified by Sort Numeric

Properties Spider Hierarchy

Repository: Project: Sandbox A. Bencoe Username: Alex Bencoe Authentication Mode: WINDOWS

UNCLASSIFIED UNLIMITED RELEASE

## 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

The screenshot displays the Spyder Python IDE (Python 3.8) on the left, showing the `EntAttEditor.py` file. The code defines a function `updateDesignEntParam` that updates entity parameters. The function uses `project.GetEntity` to retrieve entity data and `win.update` to update specific parameters. The code is as follows:

```
84 while True:
85     e, v = win.read()
86     if e == psg.MIN_CLOSED:
87         break
88     if e == 'entity':
89         win['num'].update(DfClassAtts[v['c']])
90         win['name'].update(DfClassAtts[v['c']])
91         win['desc'].update(DfClassAtts[v['c']])
92         win['entNum'].update(v['entNum'])
93         win['entName'].update(v['entName'])
94         win['entDesc'].update(v['entDesc'])
95     if e == 'populate':
96         win['entNum'].update(v['num'])
97         win['entName'].update(v['name'])
98         win['entDesc'].update(v['desc'])
99     if e == 'class':
100         entityList = list(DfClassAtts[v['c']])
101         win['entity'].update(sorted(entityList))
102     if e == 'Exit':
103         break
104     if e == 'Update':
105         if v['entNum'] != v['num']:
106             try:
107                 project.GetEntity(Guid(DfClassAtts[v['c']]))
108                 project.GetEntity(Guid(DfClassAtts[v['c']]))
109                 win['num'].update(v['entNum'])
110             except Exception as e:
111                 psg.PopupError(e, title = "Error")
112         if v['entDesc'] != v['desc']:
113             try:
114                 project.GetEntity(Guid(DfClassAtts[v['c']]))
115                 project.GetEntity(Guid(DfClassAtts[v['c']]))
116                 win['desc'].update(v['entDesc'])
117             except Exception as e:
118                 psg.PopupError(e, title = "Error")
119         if v['entName'] != v['name']:
120             try:
121                 project.GetEntity(Guid(DfClassAtts[v['c']]))
122                 project.GetEntity(Guid(DfClassAtts[v['c']]))
123                 win['name'].update(v['entName'])
124             except Exception as e:
125                 psg.PopupError(e, title = "Error")
126         win['name'].update(v['entName'])
127         win['desc'].update(v['entDesc'])
128         win['entNum'].update(v['entNum'])
129         entityList = list(DfClassAtts[v['c']])
130         win['entity'].update(sorted(entityList))
131     except Exception as e:
132         psg.PopupError(e, title = "Error")
133 win.close()
134 Repo.Logout()
135 print(f"GENESYS user {user} with session ID {sessionID}")
136
137 # =====
138 # =====
```

The right side of the image shows the GENESYS Collaborative Edition interface. The 'Example Entity 12 asPropertySheet' window is open, displaying the following information:

- Name: Example Entity 12
- Number: EE.1
- Description: This is an example for use with EntAttEditor function as part of the EntAttEditor.py script.
- Abbreviation:
- Purpose:

The 'Administrative Tools' window is also open, showing a table of sessions:

Session User	Session Start Time	Session State	Last Modified
Alex Bencoe	11/2/2021 12:32:04 PM	Active	11/2/2021 1:46:43 PM

# Questions?

