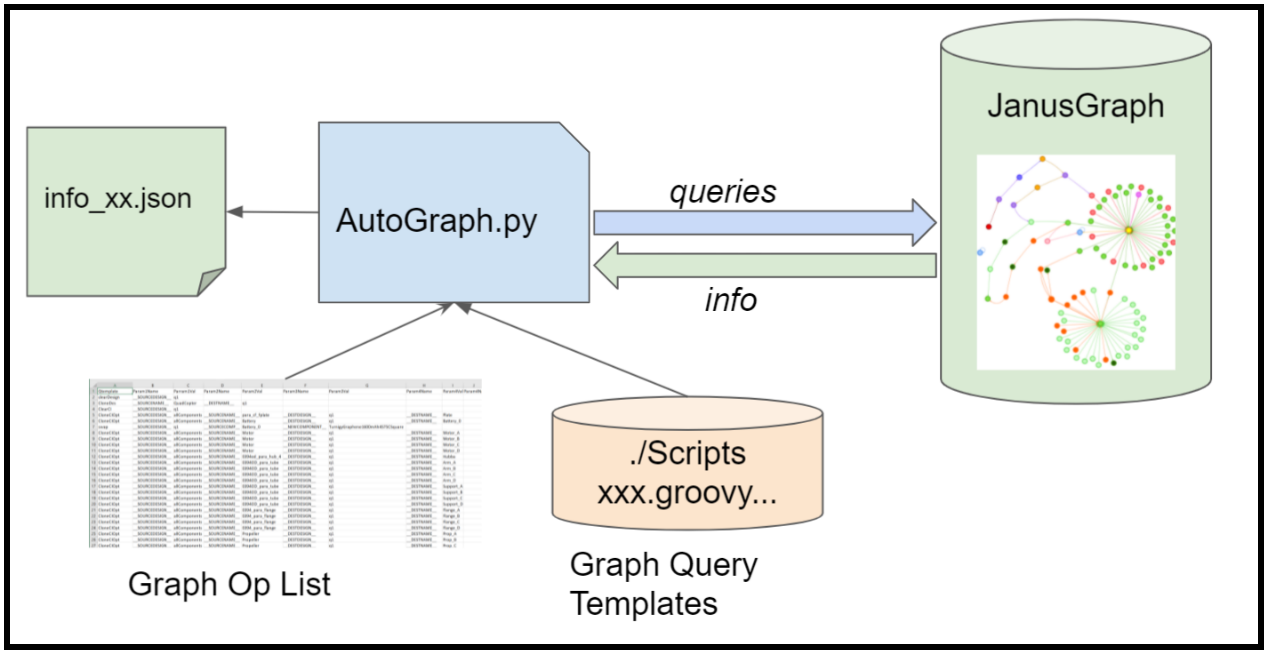
Graph Automation Tool: AutoGraph

Note: This tool is still under active development. It can be used as-is, or as an example for incorporation into SymCPS TA1/2 tools.



Purpose:

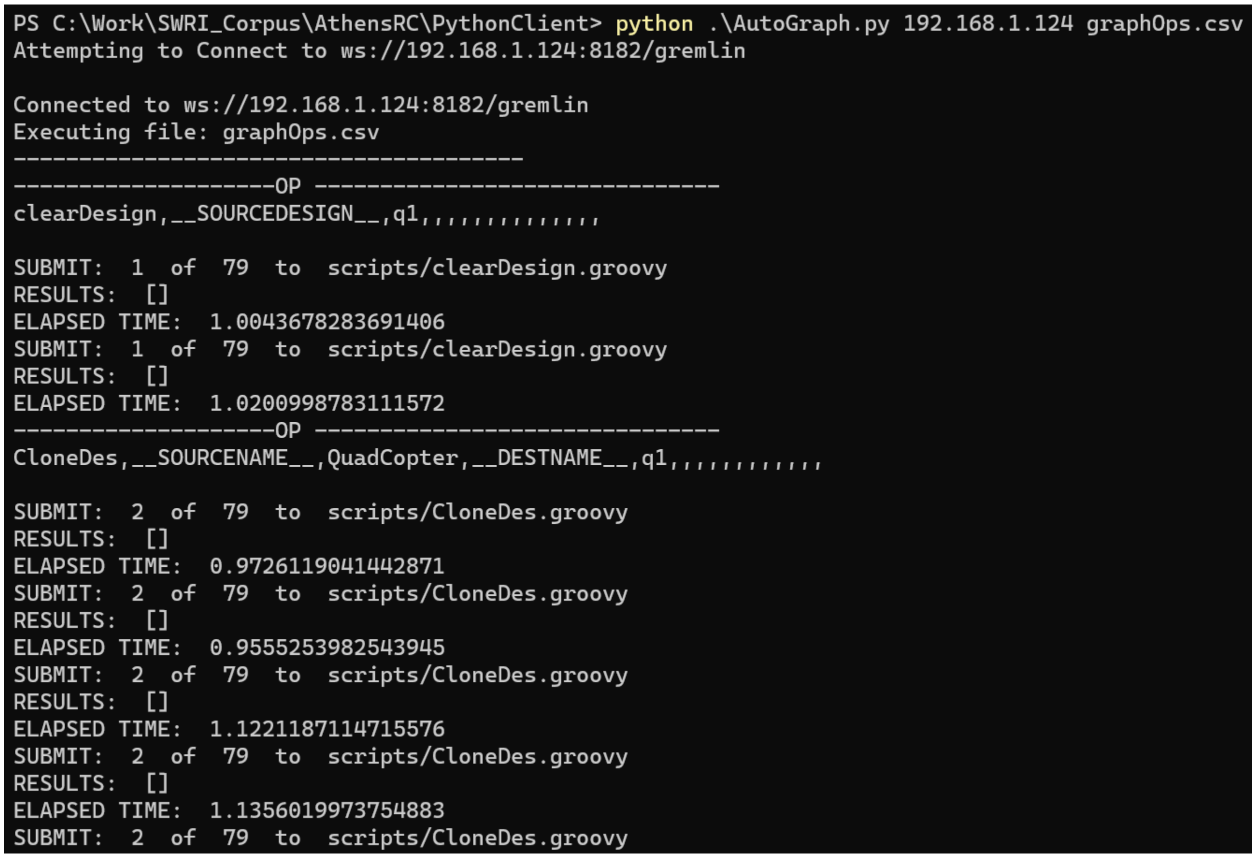
The tool executes a series of graph queries, specified in a CSV format. The tool connects to a specified JanusGraph instance and sends the queries to the graph database server synchronously, i.e. waiting for each query to return before submitting the next. Using the tool, an entire design can be constructed, updates to the graph can be made, or information can be extracted to a local json file.

Prerequisites: <NEED TO VALIDATE>

This script uses Python 3x, and several python packages:
Pip install gremlin_python
Pip install json
<FILL IN>….

Execution:

Python ./AutoGraph.py <graphServer_HostIP> <QueryListFileName>
E.g. (partial output):

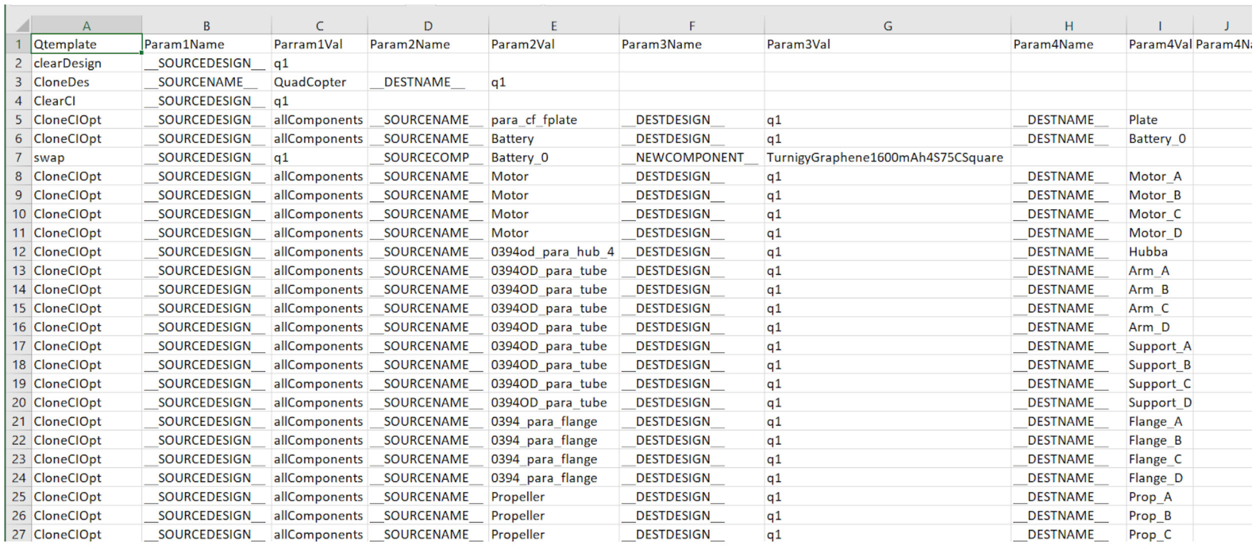


```
PS C:\Work\SWRI_Corpus\AthensRC\PythonClient> python .\AutoGraph.py 192.168.1.124 graphOps.csv
Attempting to Connect to ws://192.168.1.124:8182/gremlin

Connected to ws://192.168.1.124:8182/gremlin
Executing file: graphOps.csv
---------------------------------------
--------------------OP ------------------------------
clearDesign,__SOURCEDESIGN__,q1,,,,,,,,,,,,,,

SUBMIT:  1  of  79  to  scripts/clearDesign.groovy
RESULTS:  []
ELAPSED TIME:  1.0043678283691406
SUBMIT:  1  of  79  to  scripts/clearDesign.groovy
RESULTS:  []
ELAPSED TIME:  1.0200998783111572
--------------------OP ------------------------------
CloneDes,__SOURCENAME__,QuadCopter,__DESTNAME__,q1,,,,,,,,,,,,

SUBMIT:  2  of  79  to  scripts/CloneDes.groovy
RESULTS:  []
ELAPSED TIME:  0.9726119041442871
SUBMIT:  2  of  79  to  scripts/CloneDes.groovy
RESULTS:  []
ELAPSED TIME:  0.9555253982543945
SUBMIT:  2  of  79  to  scripts/CloneDes.groovy
RESULTS:  []
ELAPSED TIME:  1.1221187114715576
SUBMIT:  2  of  79  to  scripts/CloneDes.groovy
RESULTS:  []
ELAPSED TIME:  1.1356019973754883
SUBMIT:  2  of  79  to  scripts/CloneDes.groovy
```

Each line in the CSV is printed as a separate OP.
Note that multi-line queries are submitted in sequence. Each submission is timed.

Configuration:

Graph query sequences are specified in the csv file:



	A	B	C	D	E	F	G	H	I	J
1	Qtemplate	Param1Name	Parram1Val	Param2Name	Param2Val	Param3Name	Param3Val	Param4Name	Param4Val	Param4N
2	clearDesign	__SOURCEDESIGN__	q1							
3	CloneDes	__SOURCENAME__	QuadCopter	__DESTNAME__	q1					
4	ClearCI	__SOURCEDESIGN__	q1							
5	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	para_cf_fplate	__DESTDESIGN__	q1	__DESTNAME__	Plate	
6	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Battery	__DESTDESIGN__	q1	__DESTNAME__	Battery_0	
7	swap	__SOURCEDESIGN__	q1	__SOURCECOMP__	Battery_0	__NEWCOMPONENT__	TurnigyGraphene1600mAh4S75CSquare			
8	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Motor	__DESTDESIGN__	q1	__DESTNAME__	Motor_A	
9	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Motor	__DESTDESIGN__	q1	__DESTNAME__	Motor_B	
10	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Motor	__DESTDESIGN__	q1	__DESTNAME__	Motor_C	
11	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Motor	__DESTDESIGN__	q1	__DESTNAME__	Motor_D	
12	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394od_para_hub_4	__DESTDESIGN__	q1	__DESTNAME__	Hubba	
13	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Arm_A	
14	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Arm_B	
15	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Arm_C	
16	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Arm_D	
17	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Support_A	
18	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Support_B	
19	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Support_C	
20	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394OD_para_tube	__DESTDESIGN__	q1	__DESTNAME__	Support_D	
21	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394_para_flange	__DESTDESIGN__	q1	__DESTNAME__	Flange_A	
22	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394_para_flange	__DESTDESIGN__	q1	__DESTNAME__	Flange_B	
23	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394_para_flange	__DESTDESIGN__	q1	__DESTNAME__	Flange_C	
24	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	0394_para_flange	__DESTDESIGN__	q1	__DESTNAME__	Flange_D	
25	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Propeller	__DESTDESIGN__	q1	__DESTNAME__	Prop_A	
26	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Propeller	__DESTDESIGN__	q1	__DESTNAME__	Prop_B	
27	CloneCIOpt	__SOURCEDESIGN__	allComponents	__SOURCENAME__	Propeller	__DESTDESIGN__	q1	__DESTNAME__	Prop_C	

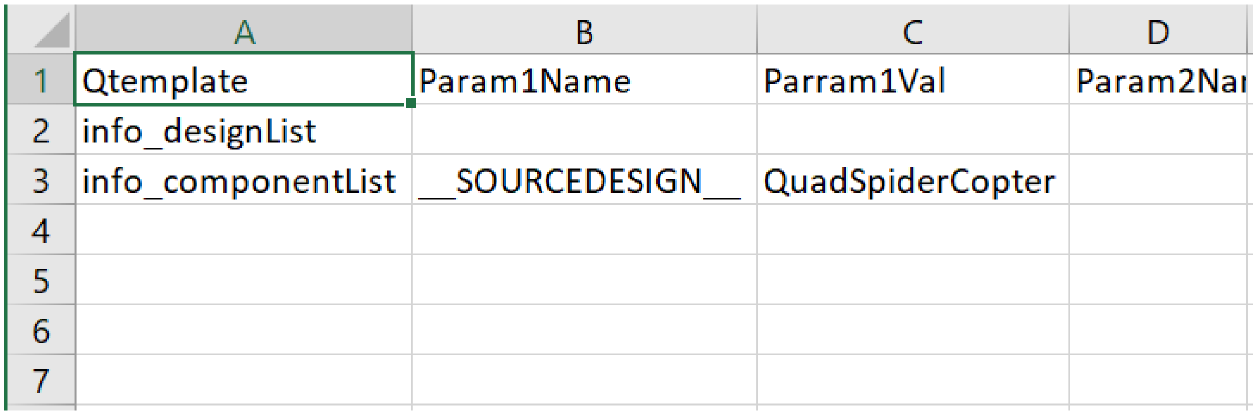
The first line is a column header, and is not executed.
Subsequent lines follow the pattern:
COL 1: Query template name. The suffix ".groovy" will be added and used to reference the query template file in ./scripts
COL 2: Param1Name - the template variable string used in the scripts/QTemplate.groovy file for parameter 1
COL 3: Param1Val - the value to be substituted for the template variable string.
This pattern is repeated for all the template variables in the groovy file.

Graph Building Example

The example (GraphOps.csv) above clears the database of designs named q1, then clones QuadCopter into q1, clears the existing ComponentInstances. The following lines perform a series of ComponentInstance copies from the allComponents design, swap component assignments, add connections, add a parameter, etc.

Graph Info Extraction:

A QTemplate that starts with the string "info" will produce a JSON-formatted file with the results of the query. The info.csv example executes two "info"queries.

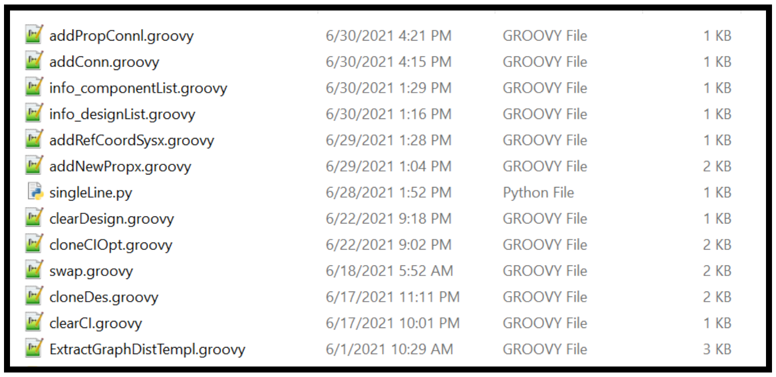


	A	B	C	D
1	Qtemplate	Param1Name	Parram1Val	Param2Nai
2	info_designList			
3	info_componentList	__SOURCEDESIGN__	QuadSpiderCopter	
4				
5				
6				
7				

The query template file info_designList.groovy shows an example of how to return results:
`g.V().hasLabel('[avm]Design').as('designs').select('designs').by('[]Name')`
This returns a JSON-formated file representing the return from the query (A python list):
`["QuadCopter", "QuadCopter", "q1", "QTemplate", "QuadSpiderCopter", "HCopter", "allComponents", "HexRing"]`
The file name is generated following te pattern QTemplate[SEQUENCENUMBER].json, where SEQUENCENUMBER is the counter of queries executed. **(in this case: info_designList1.json)**

Graph Query Scripts (Templates)

Parameterized graph queries are specified in a subdirectory ("Scripts"). These queries will be functionally equivalent to the queries supported in Jenkins.



addPropConnl.groovy	6/30/2021 4:21 PM	GROOVY File	1 KB
addConn.groovy	6/30/2021 4:15 PM	GROOVY File	1 KB
info_componentList.groovy	6/30/2021 1:29 PM	GROOVY File	1 KB
info_designList.groovy	6/30/2021 1:16 PM	GROOVY File	1 KB
addRefCoordSysx.groovy	6/29/2021 1:28 PM	GROOVY File	1 KB
addNewPropx.groovy	6/29/2021 1:04 PM	GROOVY File	2 KB
singleLine.py	6/28/2021 1:52 PM	Python File	1 KB
clearDesign.groovy	6/22/2021 9:18 PM	GROOVY File	1 KB
cloneClOpt.groovy	6/22/2021 9:02 PM	GROOVY File	2 KB
swap.groovy	6/18/2021 5:52 AM	GROOVY File	2 KB
cloneDes.groovy	6/17/2021 11:11 PM	GROOVY File	2 KB
clearCI.groovy	6/17/2021 10:01 PM	GROOVY File	1 KB
ExtractGraphDistTempl.groovy	6/1/2021 10:29 AM	GROOVY File	3 KB

Notes on query scripts:

1. Graph query template files must not contain any newlines within a query.
2. Multiple queries can be encoded within a query template file. These will be executed in sequence.
3. Info_xxx.groovy files should be a single line/query.