Automated Testing

Blue Prism

Document Revision 1.0 (22-Mar-21)

Trademarks and copyrights

Contents

1. Introduction

The purpose of this document is to provide a guide to set up components for Blue Prism Automated Testing.

2. Background

The guide gives a simplistic set up which can be run and testing in lower environments. Users can choose to use the asset as is or modify it to make it more scalable. There are two components which need to be set up.

i. Blue Prism component
ii. Python Web service component

3. Software Installation

Term	**Link**		
Blue Prism	Robotic Process Automation tool - https://portal.blueprism.com/products/current (asset supports v6.6 - v6.10.1)		
GIT	Source code repository. You need to have GIT installed in your system. For this guide a public repo was created at: https://github.com/ashz30/BluePrism-Automated-Testing.git		
Python	Python Version 3.6.8 is used, Download link here. Python IDE used is PyCharm. Download link here. (users can also run the code without the IDE as a command line too)		
Third Party Python Libraries (please refer to links for any license information)	**Library**	**Version**	**License** (at the time of publishing)
	pyodbc	4.0.30	MIT
	Flask	1.1.2	BSD License (BSD-3-Clause)
	Jinja2	2.11.3	BSD License (BSD-3-Clause)
	MarkupSafe	1.1.1	BSD License (BSD-3-Clause)
	WerkZeug	1.0.1	BSD License (BSD-3-Clause)
	click	7.1.2	BSD License (BSD-3-Clause)
	itsdangerous	1.1.0	BSD License

4. Pre-requisites

Prior to starting this process, the below steps need to be configured:

- Clone git repository - open cmd in a blank folder (mine is C:\GIT) and type

  "git clone https://github.com/ashz30/BluePrism-Automated-Testing.git"

- Install Python, open any cmd window and run the below commands:

  (a) Check python version – "python --version"

  (if this does not show the correct python version, you will need to edit windows path to point to python folderand open a new cmd window to try again)

  Copy all contents in python web service folder to a new project folder a new project folder a new project folder (name as per project). Open terminal in this folder and run.

  "pip3 install -r requirements.txt"

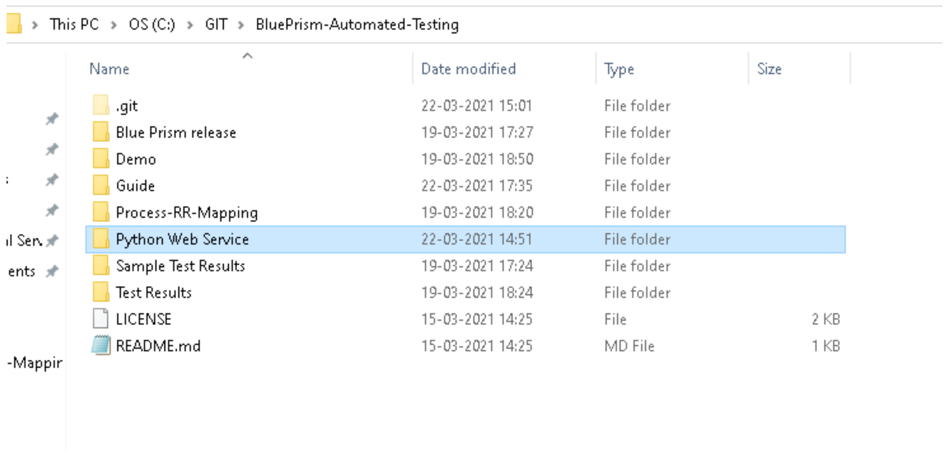
  (to download and install all python libraries)

- User will need to install Blue Prism in any machine.

- The machine which runs the python web service needs to directly connect to Blue prism database with the windows AD credentials and run SQL's explained in Python web service set up steps section.

5. Python Web service set up steps.

- Once cloned this is the structure of the GIT folder –

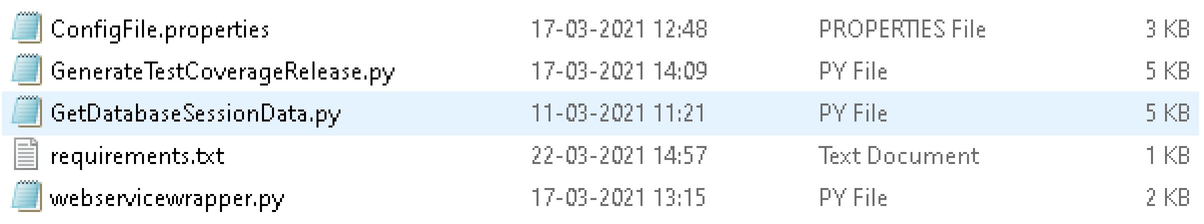


This PC › OS (C:) › GIT › BluePrism-Automated-Testing

Name	Date modified	Type	Size
.git	22-03-2021 15:01	File folder	
Blue Prism release	19-03-2021 17:27	File folder	
Demo	19-03-2021 18:50	File folder	
Guide	22-03-2021 17:35	File folder	
Process-RR-Mapping	19-03-2021 18:20	File folder	
Python Web Service	22-03-2021 14:51	File folder	
Sample Test Results	19-03-2021 17:24	File folder	
Test Results	19-03-2021 18:24	File folder	
LICENSE	15-03-2021 14:25	File	2 KB
README.md	15-03-2021 14:25	MD File	1 KB

- Goto folder - GIT\BluePrism-Automated-Testing\Python Web Service (or your corresponding folder path).

You should be able to view below files:

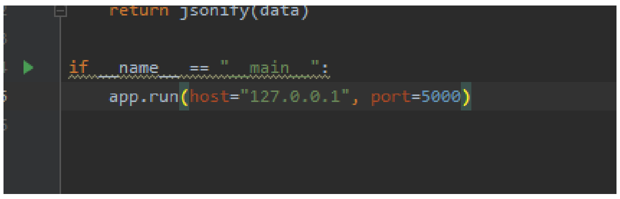


ConfigFile.properties	17-03-2021 12:48	PROPERTIES File	3 KB
GenerateTestCoverageRelease.py	17-03-2021 14:09	PY File	5 KB
GetDatabaseSessionData.py	11-03-2021 11:21	PY File	5 KB
requirements.txt	22-03-2021 14:57	Text Document	1 KB
webservicewrapper.py	17-03-2021 13:15	PY File	2 KB

- Goto file **ConfigFile.properties**, and update these properties:
  - server=<Blue Prism Database details> (for SQL statement runs present in last 2 properties)
  - database=<Blue Prism DBConn name as present in Automate.config> (for Test coverage release import)

Other properties can be left as is unless required. The machine which runs the python web service needs to directly connect to Blue prism database with the windows AD credentials. So the Windows user needs to have access to run the select sql's present in the properties – testcoveragesql, processsessiondatasql objectsessiondatasql and sessiondatasql

- Open file 'webservicewrapper.py', and replace host and port values as required for the machine which runs the python web service.



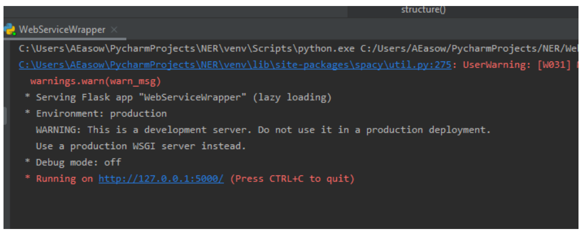
```
    return jsonify(data)

if __name__ == "__main__":
    app.run(host="127.0.0.1", port=5000)
```

- Open command line in folder '**Python Web Service' (or your corresponding project folder)** and run command

  'python webservicewrapper.py'

  alternately run webservicewrapper.py through the IDE (pycharm).

- Provided everything is set up correctly, it should start a web server in your local system.



```
C:\Users\AEasow\PycharmProjects\NER\venv\Scripts\python.exe C:/Users/AEasow/PycharmProjects/NER/We
C:\Users\AEasow\PycharmProjects\NER\venv\lib\site-packages\spacy\util.py:275: UserWarning: [W031]
  warnings.warn(warn_msg)
 * Serving Flask app "WebServiceWrapper" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

- Open your browser to goto link –

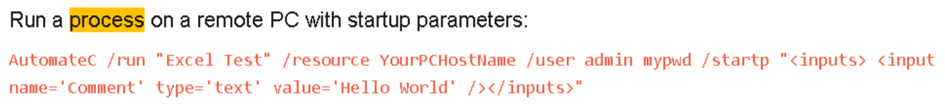
  "127.0.0.1:5000"

  Hit enter

- You should see the result "Hello World"

6. Blue Prism Steps

- Open Blue Prism.
- Import release file in the folders –
  a) IT\BluePrism-Automated-Testing\Blue Prism release - C:\GIT\BluePrism-Automated-Testing\Python Web Service\release\Registration Process.bprelease (*for Demo only*)
  b) GIT\BluePrism-Automated-Testing\Blue Prism release\Test Suite Master v6.bprelease
- Update Excel - GIT\BluePrism-Automated-Testing\Process-RR-Mapping\Process-RR-Mapping.xlsx.
  a) Blue Prism Process name - Process to be tested
  b) Runtime Resource host – Hostname/IP of runtime resource
  c) Runtime Resource Port – Port of Runtime Resource
  d) SSO Flag – True, if AD is used for Blue Prism
  e) RR Credential Name – Credential name to be used if AD is not configured.
  f) DBConName – Connection name present in Runtime Resource Automate.config file.
  g) Start Up Param – Start up parameter for Blue Prism process in xml parameter format. Sample link here.



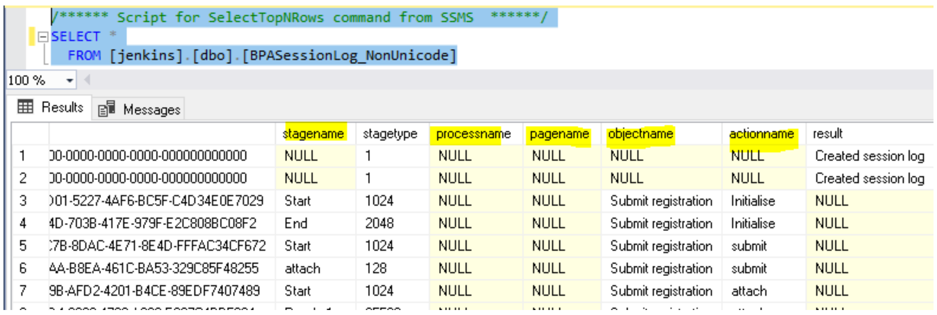
Run a process on a remote PC with startup parameters:

```
AutomateC /run "Excel Test" /resource YourPCHostName /user admin mypwd /startp "<inputs> <input
name='Comment' type='text' value='Hello World' /></inputs>"
```

  h) Test Results Data File location – Excel with the process name to be stored here GIT\BluePrism-Automated-Testing\Test Results. Sample file present here : GIT\BluePrism-Automated-Testing\Sample Test Results.
  i) Import Test coverage release Flag – true, if test coverage needs to be run and imported.
  j) Release file location – Original release of the process being tested.
  k) New release file name – new release file name of the process being tested after test coverage font change.
- Update Excel(s) - GIT\BluePrism-Automated-Testing\Test Results\<processname.xlsx>.
  a) Master Sheet –
    (1) Queues to be processed, Worksheet name for Queue item status : Any queue names from which data needs to be extracted and worksheet names for respective queue names. Worksheets need to be blank if created. If not created automation will create a new worksheet.
    (2) Session data extraction worksheets – One Worksheet for each stage from which session data needs to be extracted. Worksheet created needs to have first two rows filled as shown in template

Stage Name	**Process Name**	**Page Name**	**Object Name**	**Action Name**
Work Queues::Get Next Item	Submit Regstration process	process queue data	Blueprism.Automate.clsWorkQueuesActions	Get Next Item

This data is available in the BPASessionLog_NonUnicode or equivalent table.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT *
  FROM [jenkins].[dbo].[BPASessionLog_NonUnicode]
```

100 %

Results | Messages

		stagename	stagetype	processname	pagename	objectname	actionname	result
1	)0-0000-0000-0000-000000000000	NULL	1	NULL	NULL	NULL	NULL	Created session log
2	)0-0000-0000-0000-000000000000	NULL	1	NULL	NULL	NULL	NULL	Created session log
3	)01-5227-4AF6-BC5F-C4D34E0E7029	Start	1024	NULL	NULL	Submit registration	Initialise	NULL
4	4D-703B-417E-979F-E2C808BC08F2	End	2048	NULL	NULL	Submit registration	Initialise	NULL
5	:7B-8DAC-4E71-8E4D-FFFAC34CF672	Start	1024	NULL	NULL	Submit registration	submit	NULL
6	4A-B8EA-461C-BA53-329C85F48255	attach	128	NULL	NULL	Submit registration	submit	NULL
7	9B-AFD2-4201-B4CE-89EDF7407489	Start	1024	NULL	NULL	Submit registration	attach	NULL

b) Input Sheet – populated for ease of formulae. Tester should know the input required for the test.

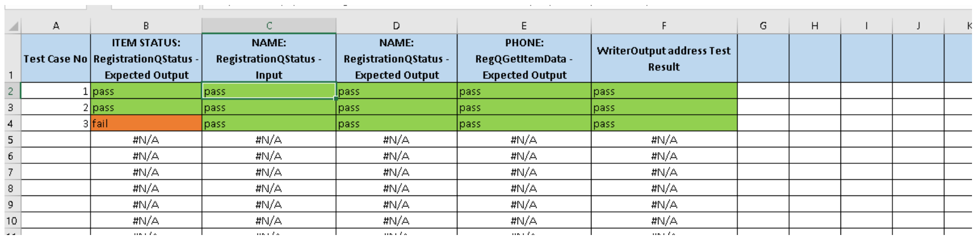
c) Expected Output Sheet - populated for ease of formulae. Tester should know the Expected Output required for the test.

d) Validation Sheet – Formulae for Validating output. This can be written either as simple excel formulae, or macros can be written when output validation requires complex logic. Sheet is locked with password admin to avoid formaes getting corrupted when sheets are deleted or overwritten.

- Update Credentials -Test Suite credentials: User needs to have access to run Blue Prism processes, Check status of processes and import releases. If SSO is used, the commands runs with the user context of the SSO user currently logged in the Runtime Resource., in that case the logged in SSO user needs to have the rights to run and import Blue Prism processes.
- Update Web API endpoint – In Blue Prism, goto System – Objects -Web API Services – Automated Testing. Update base url, replace host and port as updated in file 'webservicewrapper.py'

7. Demo Run

- If everything is set up correctly, run the BP process. It will sequentially execute each process in the specified runtime resource and output will be obtained in Test results folder. Validation sheet of each process will show the status of the Test cases.



	A	B	C	D	E	F	G	H	I	J	K
1	Test Case No	ITEM STATUS: RegistrationQStatus - Expected Output	NAME: RegistrationQStatus - Input	NAME: RegistrationQStatus - Expected Output	PHONE: RegQGetItemData - Expected Output	WriterOutput address Test Result					
2	1	pass	pass	pass	pass	pass					
3	2	pass	pass	pass	pass	pass					
4	3	fail	pass	pass	pass	pass					
5		#N/A	#N/A	#N/A	#N/A	#N/A					
6		#N/A	#N/A	#N/A	#N/A	#N/A					
7		#N/A	#N/A	#N/A	#N/A	#N/A					
8		#N/A	#N/A	#N/A	#N/A	#N/A					
9		#N/A	#N/A	#N/A	#N/A	#N/A					
10		#N/A	#N/A	#N/A	#N/A	#N/A					

- After every run a new Test results sheet needs to be placed in folder.
- For Registration Process, GIT\BluePrism-Automated-Testing\Python Web Service\release\Registration Form.html is the end application. Open it in ie for the first 2 test cases to pass ITEMSTATUS test cases, if not only the 3rd case will pass for ITEMSTATUS.

8. Limitations

- Performance has not been evaluated, this asset is to be used in a lower environment only, if it is to be made scalable, performance , risk and best practice recommendations need to be implemented.
- Asset queries BPASessionLog_NonUnicode tables to get information. Queries present in config.properties should be evaluated from performance viewpoint if this asset needs tobe used at scale.
- Python code is for POC purposes and should be reviewed and modified, if required to be used at scale.
- Asset gets the output from internal Blue Prism tables and not the end application, this results in certain exception conditions, few egs.
    - If Global Send keys are used, data would be sent to the end application but without correctly clicking on the correct field data will not be input correctly. Assumption is that the main Automation will throw exception in such cases.
    - Object uses only Index as the unqiue criteria for spying a field, but due to any reason the index gives a valid identification and data written by the object for the field identified by the index does not throw an exception, but in the end application the actual field to be identified is not the same.

    This is not an exhaustive list.

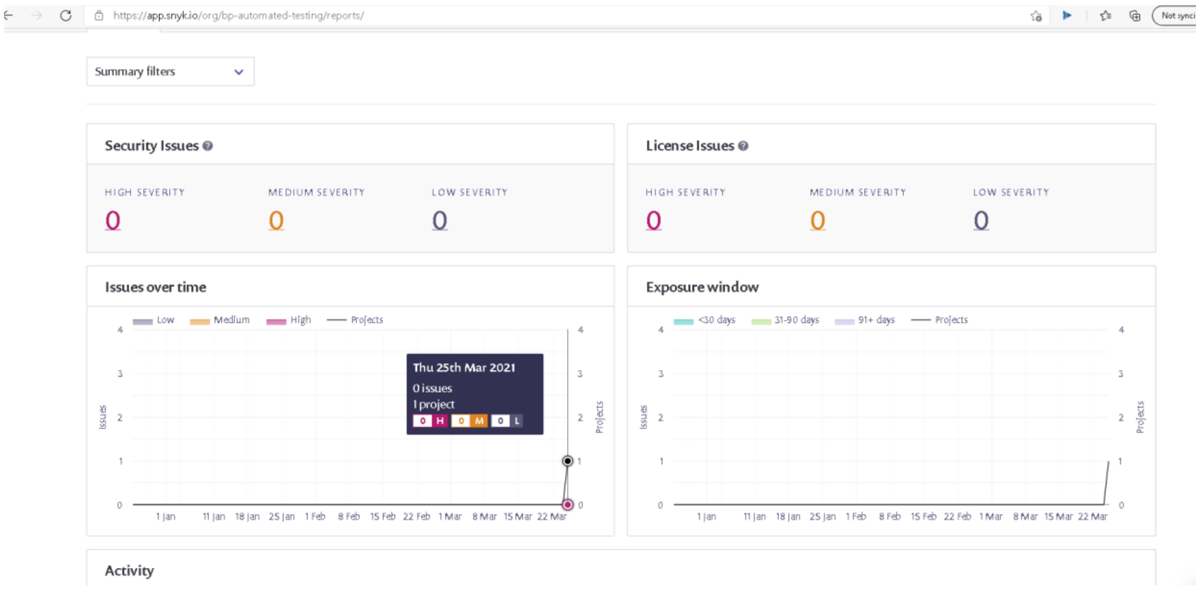
    In such cases users need to write an output extraction process customised to be automation to populate test results excel sheets which will have the correct output extracted from the end application (rather than BP database). Test Suite master needs to be modified to accommodate adding output extraction processes.
- Test coverage feature imports the release file as a last step, this can be customised if not needed, but every import will add to the audit table in Blue Prism.

9. License and Support

The guide and supporting bprelease are available for free under MIT license (license present in GIT repository). The Python libraries used have their own license (BSD 3 clause at the time of publishing), please do evaluate the commercial aspects of the libraries and scan for any risks and vulnerabilities as per your organizations guidelines.

The asset itself is community supported, any requirement for support can be directed to the Dx community at this link.

10. Python Source Code Security Scan Results (25-March-2021)

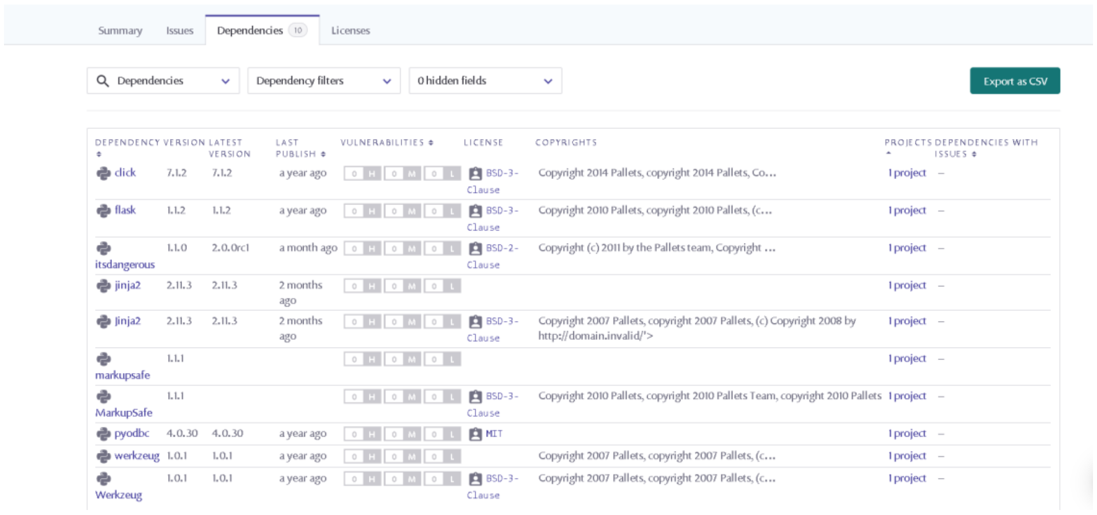


Security Issues

HIGH SEVERITY	MEDIUM SEVERITY	LOW SEVERITY
0	0	0

License Issues

HIGH SEVERITY	MEDIUM SEVERITY	LOW SEVERITY
0	0	0



DEPENDENCY	VERSION	LATEST VERSION	LAST PUBLISH	VULNERABILITIES	LICENSE	COPYRIGHTS	PROJECTS	DEPENDENCIES WITH ISSUES
click	7.1.2	7.1.2	a year ago	0 H 0 M 0 L	BSD-3-Clause	Copyright 2014 Pallets, copyright 2014 Pallets, Co…	1 project	–
flask	1.1.2	1.1.2	a year ago	0 H 0 M 0 L	BSD-3-Clause	Copyright 2010 Pallets, copyright 2010 Pallets, (c…	1 project	–
itsdangerous	1.1.0	2.0.0rc1	a month ago	0 H 0 M 0 L	BSD-2-Clause	Copyright (c) 2011 by the Pallets team, Copyright …	1 project	–
jinja2	2.11.3	2.11.3	2 months ago	0 H 0 M 0 L			1 project	–
Jinja2	2.11.3	2.11.3	2 months ago	0 H 0 M 0 L	BSD-3-Clause	Copyright 2007 Pallets, copyright 2007 Pallets, (c) Copyright 2008 by http://domain.invalid/">	1 project	–
markupsafe	1.1.1			0 H 0 M 0 L			1 project	–
MarkupSafe	1.1.1			0 H 0 M 0 L	BSD-3-Clause	Copyright 2010 Pallets, copyright 2010 Pallets Team, copyright 2010 Pallets	1 project	–
pyodbc	4.0.30	4.0.30	a year ago	0 H 0 M 0 L	MIT		1 project	–
werkzeug	1.0.1	1.0.1	a year ago	0 H 0 M 0 L		Copyright 2007 Pallets, copyright 2007 Pallets, (c…	1 project	–
Werkzeug	1.0.1	1.0.1	a year ago	0 H 0 M 0 L	BSD-3-Clause	Copyright 2007 Pallets, copyright 2007 Pallets, (c…	1 project	–