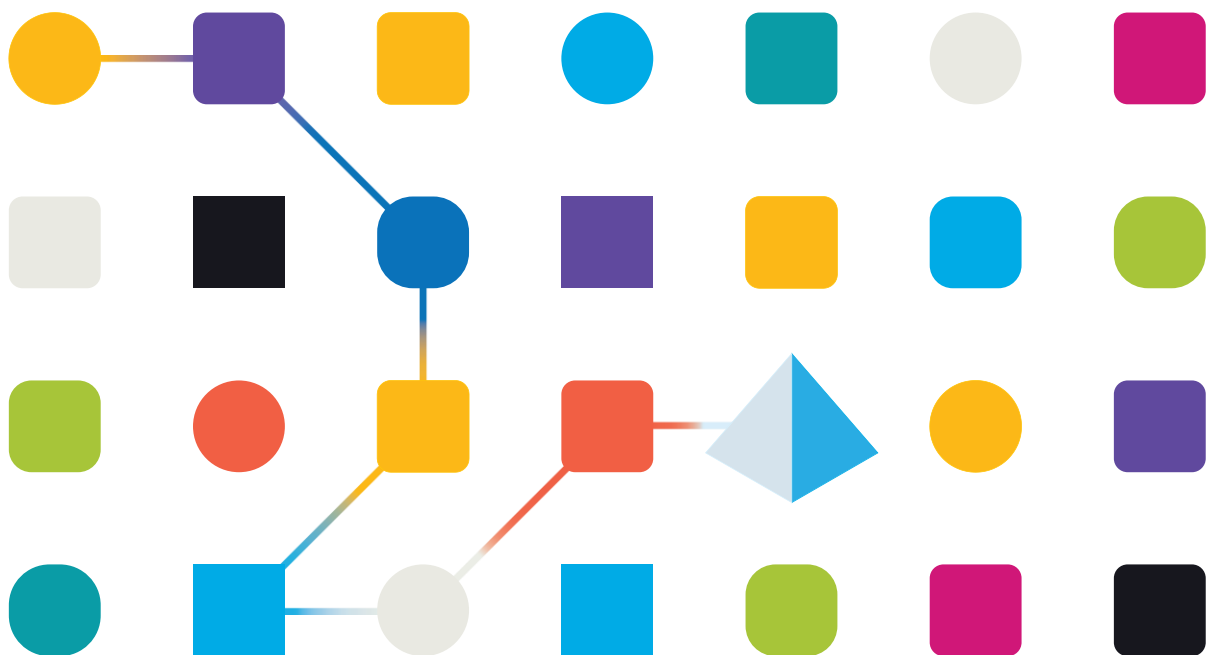# blueprism®

Blue Prism Automated Testing
User Guide

Document Revision 1.0

Trademarks and copyright

The descriptions and screenshots contained in this document are licensed under the Creative Commons Attribution-ShareAlike (CC-BY-SA) 3.0 license https://creativecommons.org/licenses/by-sa/3.0/.

Contents

Introduction

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

Prerequisites

Software	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. (Apache 2 license at the time of publishing) (users can also run the code without the IDE as a command line too or their own IDE)

	Name	Version	Licenses	Copyright
Third Party Python Libraries **(please refer to links in name for any addtional license information)**	click	7.1.2	BSD-3-Clause	["Copyright 2014 Pallets","copyright 2014 Pallets","Copyright 2001-2006 Gregory P. Ward","Copyright 2002-2006 Python Software Foundation"]
	flask	1.1.2	BSD-3-Clause	["Copyright 2010 Pallets","copyright 2010 Pallets","(c) Copyright 2010 by http://domain.invalid/'>","Copyright (c) 2010 by the Pallets team"]
	itsdangerous	1.1.0	BSD-2-Clause	["Copyright (c) 2011 by the Pallets team","Copyright (c) Django Software Foundation and individual contributors","copyright 2011 Pallets Team"]
	Jinja2	2.11.3	BSD-3-Clause	["Copyright 2007 Pallets","copyright 2007 Pallets","(c) Copyright 2008 by http://domain.invalid/'>"]
	MarkupSafe	1.1.1	BSD-3-Clause	["Copyright 2010 Pallets","copyright 2010 Pallets Team","copyright 2010 Pallets"]
	pyodbc	4.0.30	MIT	
	Werkzeug	1.0.1	BSD-3-Clause	["Copyright 2007 Pallets","copyright 2007 Pallets","(c) JS Foundation and

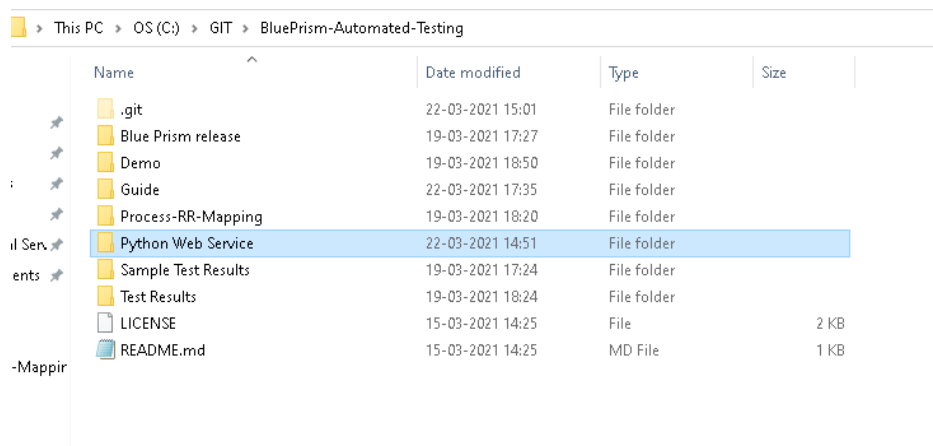|	other contributors","Copyright 2010 Canonical Ltd."]
Blue Prism VBO List **(needs to be imported)**	MS Excel VBO Utility – Collection Manipulation VBO Utility – General VBO Utility – JSON VBO Utility – Environment VBO Utility – Strings VBO

Configuration

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 (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.

Python Web service set up steps.

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

- Goto folder - GIT\BluePrism-Automated-Testing\Python Web Service (or your corresponding project folder path set up in the earliest step).

  You should be able to view below files:

- Goto file **ConfigFile.properties**, and update these properties:
    - server=<Blue Prism Database server details> (for SQL statement runs present in last 4 SQL statement properties)
   - database=<Name of Blue Prism database>

    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.

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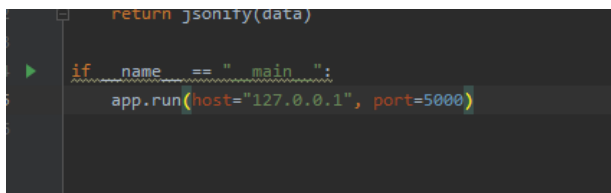
- Open command line in folder '**Python Web Service' (or your corresponding project folder)** and run command

    '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.
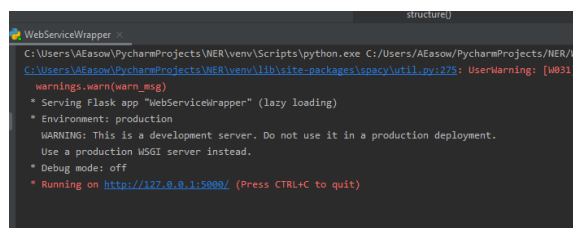
- Open your browser to goto link

    (replace below with host and port which you changed in file 'webservicewrapper.py')

    "127.0.0.1:5000"

    Hit enter

- You should see the result "Hello World"

Blue Prism set up 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. Worksheets 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 for each stage which needs to be set up is available in the BPASessionLog_NonUnicode or equivalent table.

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 (which runs Test Suite Master process), 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'

- Update logging levels - The stages of the Blue Prism Process/Object being tested, for which data needs to be extracted (configured in Session data extraction worksheets), needs to be at full logging level, so that data for those stages can be logged in the Blue Prism database and can be extracted by this framework.

Using the Asset

- If everything is set up correctly, run the BP process – Automated Testing Master 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 the sample 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 its not opened in IE, only the 3rd case will pass for ITEMSTATUS.

Known 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 table to get information. Queries present in config.properties have not been evaluated during high load and need to be analysed for running at scale in a large Blue Prism environment.

- Only the stages for which data is added to BPASessionLog_NonUnicode table can be used for data comparison and extraction. Few stages like Navigate will not leave a data footprint in BPASessionLog_NonUnicode table. To deal with such cases an data point prior to it, for eg, Action (which has the navigate) input parameter may need to be extracted and validated.

- Python code is for POC purposes and should be reviewed and modified as needed, 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 which can cause a few scenarios output evaluation to be uncertain,

  Few examples.

    o 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.

    o Object uses Index as the only unique criteria for identifying a field, but due to any reason

        (a) Index gives a valid identification for an incorrect field and

        (b) Data written by the object for the incorrect field, identified by the index, does not throw an exception

        This results in a data extracted from Blue Prism database to be correct, but the data can be incorrect in the end application.

  Note: This is not an exhaustive list.

  In such cases users need to write an output extraction process customised to the end application which populate Test results excel sheets. This output extraction process will get the correct output extracted from the end application (rather than BP database). Test Suite Master process needs to be modified to accommodate adding Output extraction process.

- 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.

Python Source Code Snyk Vulnerability Scan Results (25-March-2021)
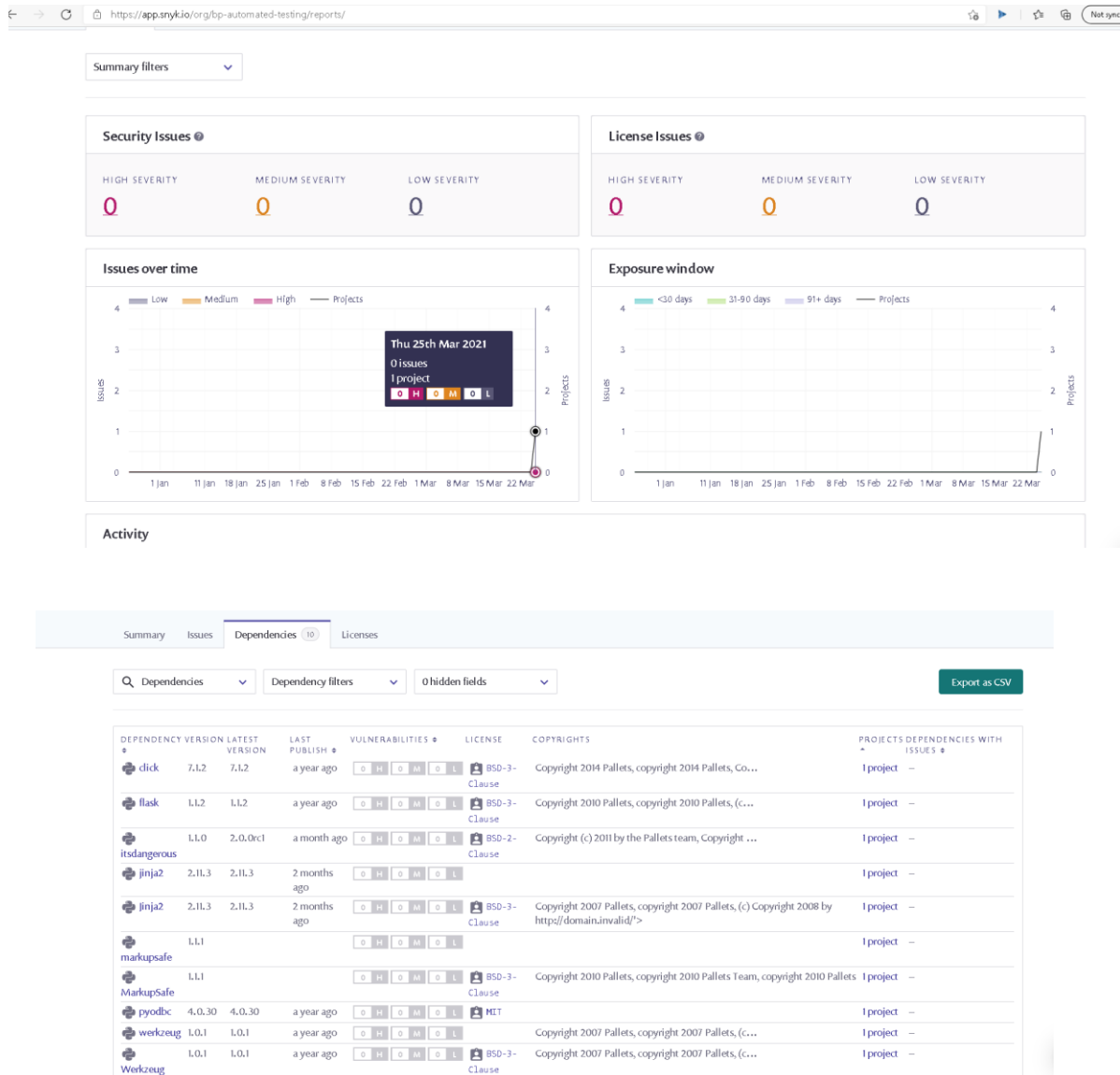
Support

This asset is provided free-of-charge by Blue Prism. Blue Prism does not provide formal support of this asset. Please direct any questions you have, related to this asset, to the Digital Exchange Community page:

https://community.blueprism.com/communities/community-home?CommunityKey=1e516cfe-4d1f-4de9-a9eb-58d15bf38c81