IxNetwork Automation getting started guide

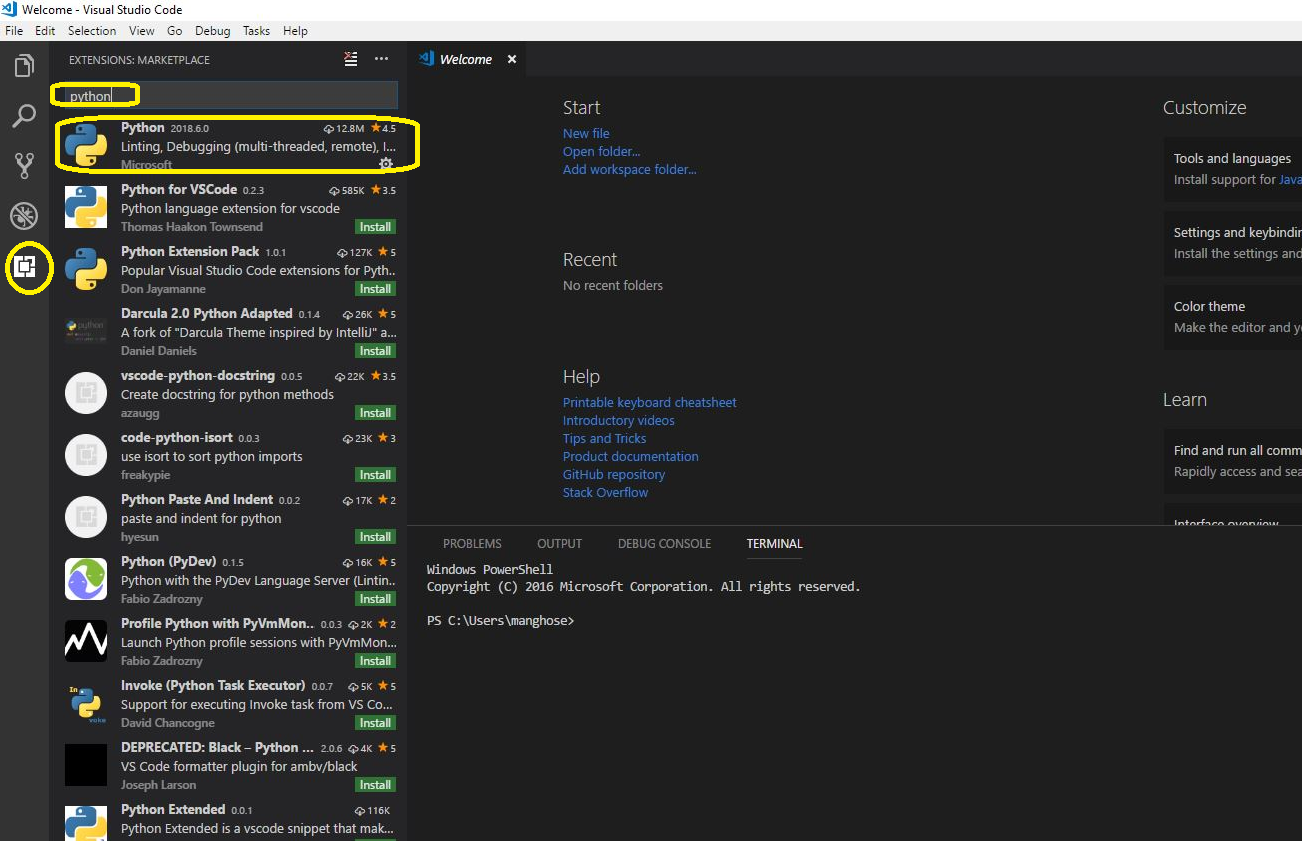
In today’s world as networks become more automatable, automation becomes inevitable and the most popular network automation technology is based on open source tools, python and REST APIs. There is a push to migrate test automation from legacy approaches, like CLI and Tcl, towards the same technology used to automate the networks, even if there is existing automation infrastructure. Keysight’s products are leading the way by incorporating the same approaches and languages as the network elements. But, Keysight is investing heavily in making the automation of its test tools the easiest in the industry.

Here we look at how to setup an automation environment with Keysight’s flagship L23 test product IxNetwork using REST and Python. We use simple, yet very common use cases to minimize the need to understand what the test is doing and focus on how the test is automated.

**Test Environment:**

1. Access to 2 Ixia ports connected back to back
2. Access to 2 Ixia ports surrounding a DUT
3. If you do not have options 1 and/or 2, look [here](#Appendix) to see how you can use Keysight’s virtualized test ports. The automation is the same.
4. Access to a windows system with admin rights (can be a VM or a physical server. For the sake of simplicity this document we will be using Windows. However, we offer full functionality on a Linux based IxNetwork API server as well)

**Steps:**

1. Install IxNetwork windows application on the Windows system
2. Once IxNetwork is installed, it also automatically installs Python. We recommend using 3.x
   1. Starting with IxNetwork 8.50, 2 versions of Python are installed in the system.
      1. C:\Program Files (x86)\Ixia\Python\2.7.6
      2. C:\Program Files (x86)\Ixia\Python\3.6.4
3. Modify your environment variables using the following steps.
   1. Open the Control Panel\System and Security\System 🡪 Advance system settings 🡪 environment variables
   2. In system variables look for the variable called “path” and click edit
   3. In Variable value append this (don’t include the quotes) “;C:\Program Files (x86)\Ixia\Python\3.6.4”
   4. Open a cmd and type python to check if you have the path set properly. If you do, you will see a python prompt, if you don’t, you will see an error message indicating that Windows doesn’t understand the command “python”
4. Open IxNetwork API sever from start menu
   1. By default, the API server listens to TCP port 11009 for REST and TCP port 8009 for Tcl
5. Setup an editor for developing your scripts. We recommended [VSCode](https://code.visualstudio.com/)
   1. Open VSCode and setup Python extension
   2. On the left-hand side menu go to the last section called extension
   3. In the search box type Python and install the first extension from Microsoft that appears in the seach.
   4. 
   5. Once installed you have to reload VS code. (If install fails try to install after closing VS code and reopening VS code with “Run as administrator”)
6. Install GIT client for Windows from [here](https://git-scm.com/download/win). (optional, if you skip this step go to 8)
7. Clone IxNetwork sample suites from <https://github.com/OpenIxia/IxNetwork>
   1. Open VSCode
   2. Open new terminal (Ctrl + `)
   3. Execute command “git clone https://github.com/OpenIxia/IxNetwork c:/IxiaAutomation”
8. (Do this step if you have skipped 6 and 7) Download all files into a zip archive.
   1. Unzip contents to c:/IxiaAutomation
9. In VSCode click on “Add Workspace folder” and open “C:\IxiaAutomation\RestApi\Python”
10. Do your first test run on Ixia b2b ports.
    1. From the workspace open bgpNgpf.py
    2. Edit line 64 *ixChassisIp = ‘<your chassis IP>’* to use the IP address of your chassis identified in the Test Environment steps.
    3. Edit line 66 to match your port number *portList = [[ixChassisIp, '1', '1'], [ixChassisIp, '1', '2']].* This assumes you are using ports 1 and 2 on the first load module in the chassis. If that’s not the right assumption, make sure to use the card and port you identified in the Test Environment steps
    4. Edit line 79 *mainObj = Connect(apiServerIp='<Windows API server IP>',*
    5. Save the file
    6. In VSCode from Debug menu select start debugging and pick Python as environment
    7. This will start execution of the script and the output will be on terminal window.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Congratulations you have successfully run a IxNetwork test through automation using Python+REST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Next try to run Automation in IxNetwork + DUT. We find using Python’s napalm module makes it much easier to automate CLI’s if you are still using devices that don’t provide a REST API. Some napalm integrations use REST as well. The benefit of using napalm is you don’t have to worry about the details. The integration with the DUT is invisible to your script. Read more about napalm and supported devices [here](http://napalm.readthedocs.io/en/latest/). To setup Napalm do the following.
   1. In VSCode open terminal and install python modules. (it may require to run VSCode in administrator mode, if you see errors when running these commands, exit VSCode and reopen in administrator mode)
      1. python -m pip install --upgrade pip
      2. python -m pip install napalm
   2. Sample scripts developed with Arista EOS is located at C:\IxiaAutomation\RestApi\Python\Modules
   3. Napalm uses various connection methods to the DUT. For example, for Arista EOS, it uses http/https using Arista’s EAPI. You may need to setup some basic configuration in your DUT so that napalm can talk to it. For example, in EOS these are the commands to enable napalm usage. These are one-time settings, and don’t create any security risks.
      1. configure
      2. management api http-commands
      3. no shutdown
      4. <verify your configuraition> show management api http-commands
2. Run the test bgpTestDUT.py (C:\IxiaAutomation\RestApi\Python\SampleScripts\bgpTestDUT.py)
   1. The DUT configuration for this file is based on EOS and is located at C:\IxiaAutomation\RestApi\Python\SampleScripts\bgp\_eos.cfg
   2. This test does the following:
      1. configures BGP on 2 ports of IxNetwork and 2 ports of the DUT.
      2. Starts protocol on IxNetwork and checks for the protocol to be up
      3. Fetches BGP peer table from the DUT and prints it
      4. Runs traffic over BGP routes advertised from IxNetwork through the DUT and check if there is any loss, prints traffic statistics

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Congratulations you have successfully run a IxNetwork test through automation using Python+REST

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. You can now further enhance these tests into custom tests according to your test plan

Appendix:

To create a setup for this exercise you will need the following:

1. Access to a Hypervisor (we will use ESXI for this document)
2. Download access to Ixia software downloads site.
3. Evaluation licenses for 2 IxNetwork VE ports
4. Access to Arista EOS software download

Steps(these are overview of steps):

1. Download Virtual Test Appliance ova from Ixia [website](https://support.ixiacom.com/support-overview/product-support/downloads-updates/ixvm). This gives you 2 ports of Ixia.
2. Download IxNetwork software application from Ixia [website](https://support.ixiacom.com/support-overview/product-support/downloads-updates/versions/68).
3. Install IxNetwork on your windows system.
4. Download Arista EOS ova from [Arista’s EOS central](https://eos.arista.com/)
5. Deploy the OVA and add 2 test ports so in total there will be 3 ports 1 for management and 2 test ports.
6. Do necessary configuration like management IP of the switch
7. Ensure the connectivity using vswitches (you should change the vswitch connectivity based on whether you are running the back-to-back or DUT test case)
8. Start testing