

Quali



Virtual Ixia Breaking Point Shell

Release date: October, 2017

Shell version 1.0.0

Document version 1.0

Qualisystems Ltd. Confidential and proprietary.

All Rights Reserved. No part of this software or material may be copied, reproduced, stored in or introduced into a retrieval system, distributed or displayed in any form or manner for any purpose whatsoever and no derivative works may be made without Qualisystems Ltd. advance written consent. All trademarks, brand names, product names and logos are trademarks or registered trademarks of Qualisystems Ltd. or applicable licensor [TestShell, CloudShell, the Qualisystems logo and the TestShell logo. The absence of a trademark from this list does not constitute a waiver of Qualisystems or applicable licensor's intellectual property rights concerning that trademark].

The above copyright and trademark notices shall be included in all such software and/or materials.

Copyright, 2017, Qualisystems Ltd. Software and materials are copyrighted and trademarked by Qualisystems Ltd. and any incorporated third party software is copyrighted by its respective licensor.

Contents

Overview	3
About Ixia Breaking Point Shell	3
Standard version	3
Supported Breaking Point versions	3
Certified models	3
Requirements.....	3
Downloading the Shell.....	3
Automation.....	4
Breaking Point Controller Shell	4
Import and Configure the Shell	5
Importing the Shell into CloudShell.....	5
Offline installation of a Shell	5
Configuring a new applications.....	6
Configure TearDown Script	8
TearDown Script modification	8
Data Model	9
Breaking Point Chassis Families and Models	9
Ixia Breaking Point Chassis Attributes	9
Ixia Breaking Point Module Attributes	9
Virtual Port Attributes.....	10
Typical workflow and scenarios	11
Typical workflow	11
Use cases and scenarios	11
Scenario 1 – Getting test file with network configuration	11
Scenario 2 – Running test	12
Release notes	13
References.....	13

Overview

A Shell implements integration of a device model, application or other technology with CloudShell. A shell consists of a data-model that defines how the device and its properties are modeled in CloudShell along with an automation that enables interaction with the device via CloudShell.

About Ixia Breaking Point Shell

The Virtual Ixia Breaking Point Shell consists from Ixia Breaking Point VChassis Shell Ixia Breaking Point VBlade Shells which provides you with data model and load Breaking Point Chassis and Breaking Point Blade to the resource management, and BreakingPoint Controller Shell, which is a service, gives you functionality to load test configuration, run tests, getting tests results.

Standard version

The Breaking Point Shell is based on the Traffic Shell Standard Version 1.0.0.

Supported Breaking Point versions

- 8.20 and above.

Certified models

- Breaking Point VE

Requirements

- CloudShell version 7.0 and above
- Breaking Point application 8.20 and above.
- Breaking Point Controller Shell 1.2.1 and above.

Downloading the Shell

The Breaking Point Shell is available from the [Quali Download Center](#). Download the files into a temporary location on your local machine.

Note: Registration to the Quali Support Portal is required. If you have not registered, click this link to register [New registration](#).

The Shell comprises:

ixia-breakingpoint-vchassis.zip	Virtual Breaking Point Chassis Shell
ixia-breakingpoint-ve-vblade.zip	Virtual Breaking Point Blade Shell
vbp-offline-package-1.0.0.zip	Shell Python dependencies (for offline installation only)
Ixia-VirtualBreakingPoint-Shell.pdf	Documentation

Automation

This section describes the automation (drivers or scripts) associated with the data model. The automation code (either script or driver) is associated with the model and provided as part of the Shell package (in the .zip file).

The following commands are associated with a model inside the Shell:

Breaking Point Controller Shell

Command	Description
Load Configuration	Load configuration file and reserve necessary ports
Start Traffic	Start test for current configuration
Stop Traffic	Stop running test
Get Results	Get test result file and attach it to the reservation
Get Statistics	Get real time statistics
Get Test File	Download test file to the location specified in the Test Files Location attribute

Import and Configure the Shell

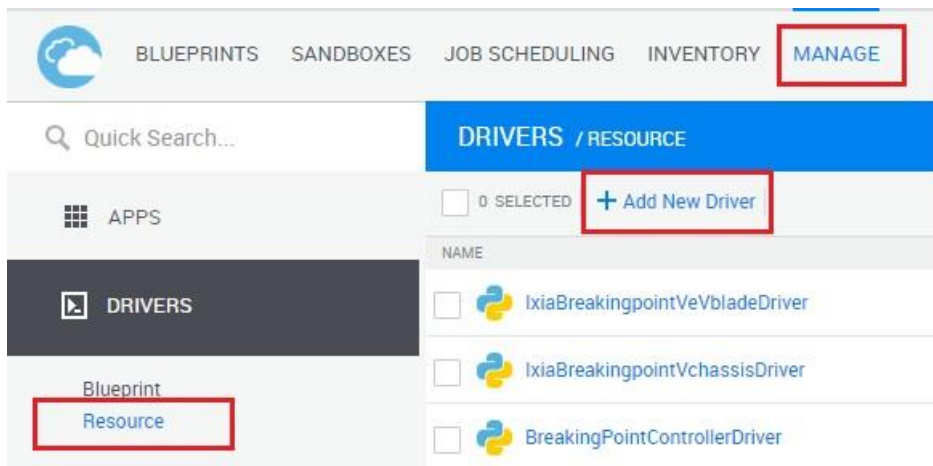
This section describes how to import, configure and modify the Breaking Point Shell.

Importing the Shell into CloudShell

Use the following procedure to import the downloaded Shell:

To import the Shell into CloudShell:

1. Download the Shell from the [Quali Download Center](#).
2. Backup your database.
3. Log in to **CloudShell Portal** as administrator of the relevant domain.
4. In the **Manage** dashboard, section **Drivers**, subsection **Resource** click **Add New Driver**



5. Browse to the location of the downloaded Shell file, select the relevant **.zip** file and Click **Open**. Alternatively, drag the shell's **.zip** file into CloudShell Portal.

Offline installation of a Shell

Note: Offline installation instructions are relevant only if Cloudshell Execution Server has no access to PyPi. You can skip this section if your execution server has access to Pypi.

The Breaking Point Shell uses a variety of Python packages. To work in offline mode:

1. Download the **vbp-offline-package-1.0.0** file (see [Downloading the Shell](#)).
2. Unzip it to a local repository. Make sure the execution server has access to this folder.
3. On the Execution Server, in the **customer.config** file, add the following key:

```
<add key="PythonOfflineRepositoryPath" value="repository  
full path"/>
```

Make sure to update the “repository full path” with path to the repository you unzipped the file to.

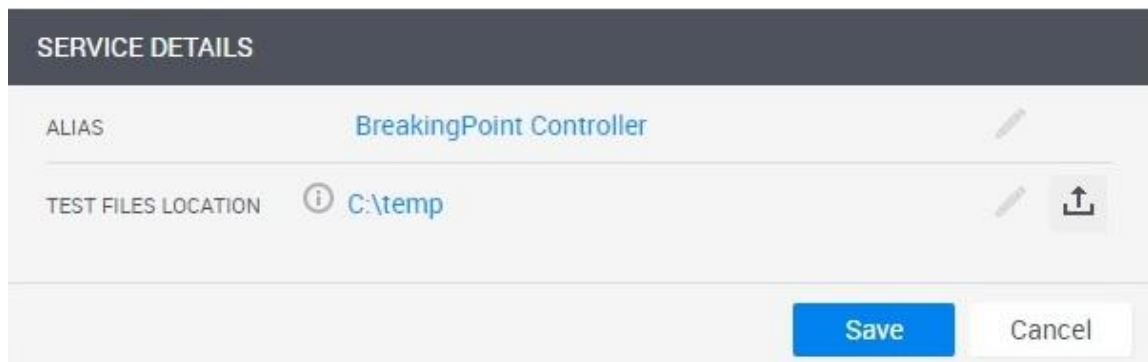
4. Restart the Execution Server.

Configuring a new applications

Use [Managing Apps](#) topic from Quali Online Help to manage an application, which will use this Shell

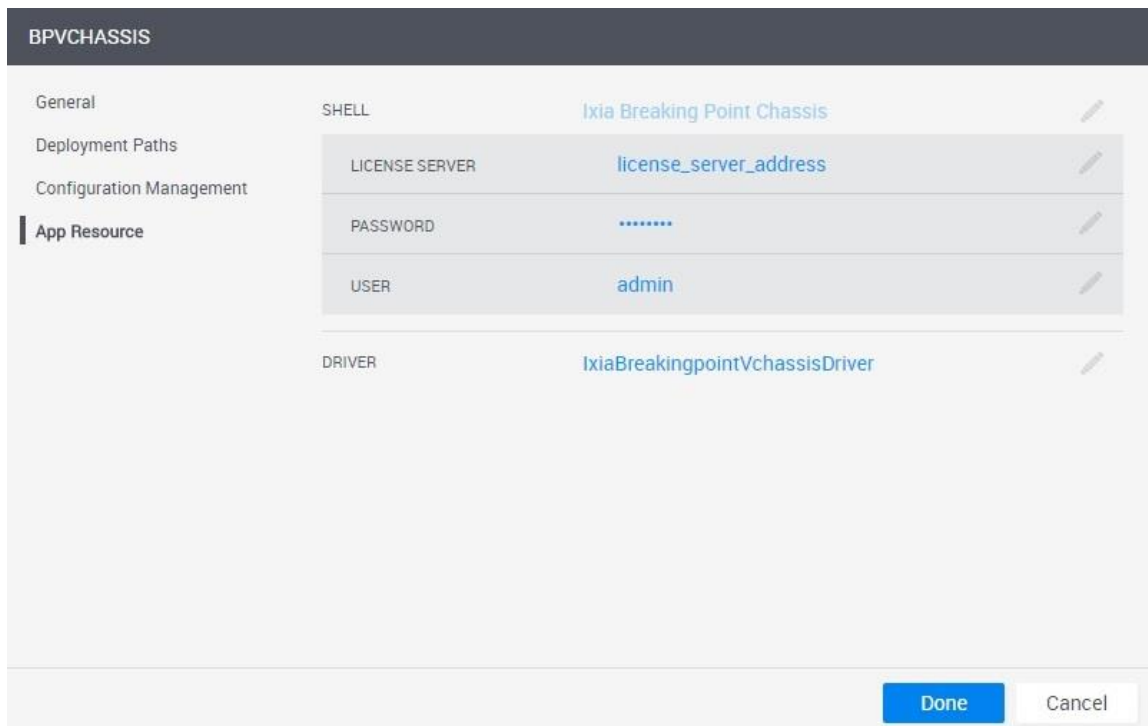
Add at least two applications (**Virtual Breaking Point Chassis** and **Virtual Breaking Point Blade**) and **BreakingPointController** service

Make sure that you specify attribute **Test Files Location**, with value of location where test files will be downloaded, for **BreakingPointController** service,



The screenshot shows a configuration window titled "SERVICE DETAILS" for the "BreakingPoint Controller" service. It features two main fields: "ALIAS" with the value "BreakingPoint Controller" and "TEST FILES LOCATION" with the value "C:\temp". The "TEST FILES LOCATION" field includes an information icon, a file selection icon, and a folder icon. At the bottom right, there are "Save" and "Cancel" buttons.

attributes **License Server**, with IP Address of License Server, **User** and **Password**, with credentials to access Breaking Point Chassis, for **Virtual Breaking Point Chassis** application



The screenshot shows a configuration window titled "BPVCHASSIS" for the "Virtual Breaking Point Chassis" application. It has a sidebar with categories: "General", "Deployment Paths", "Configuration Management", and "App Resource". The "App Resource" category is selected. The main area shows several configuration fields: "SHELL" (Ixia Breaking Point Chassis), "LICENSE SERVER" (license_server_address), "PASSWORD" (masked with dots), "USER" (admin), and "DRIVER" (IxiaBreakingpointVchassisDriver). Each field has an edit icon. At the bottom right, there are "Done" and "Cancel" buttons.

and attribute **Virtual Traffic Generator Chassis**, with name of **Virtual Breaking Point Chassis**, for **Virtual Breaking Point Blade** application(s)

BPVBLADE

General
Deployment Paths
Configuration Management
App Resource

SHELL	Ixia BreakingPoint Module	
MEMORY IN GB	8	
NUMBER OF PORTS	2	
NUMBER OF VCPUS	4	
PASSWORD	
SLOT ID	0	
USER	admin	
VIRTUAL TRAFFIC GENERAT...	BPvChassis	
DRIVER	IxiaBreakingpointVeVbladeDriver	

Done Cancel

Configure TearDown Script

This section describes how to modify teardown script to correct work of Ixia Virtual Breaking Point Generator

Teardown Script modification

Use the following procedure to modify teardown script:

1. Log in to **CloudShell Portal** as administrator of the relevant domain.
2. From **Manage** dashboard, section **Scripts**, subsection **Blueprint** download current teardown script
3. Enter downloaded archive
4. Add `cloudshell-orch-vbp>=1.0.0,<1.1.0` into **requirements.txt** file
5. Add next code into **__main__.py** file:

```
from cloudshell.workflow.orchestration.setup.vbp.configuration_commands  
import configure_virtual_chassis  
  
sandbox.workflow.on_configuration_ended(function=configure_virtual_chassis,  
components=sandbox.components.apps)
```

6. Update teardown script in CloudShell

Data Model

Breaking Point Chassis Families and Models

Family	Model	Description
Virtual Traffic Generator Chassis	Ixia Breaking Point Chassis	Breaking Point Chassis
Virtual Traffic Generator Module	Ixia Breaking Point Module	Modules located on the chassis
Port	Virtual Port	Virtual Traffic Generator Port

Ixia Breaking Point Chassis Attributes

The attribute names and types are listed in the following table:

Attribute	Type	Default value	Description
License Server	String		IP address or hostname of License Server
Password	Password		Password for Ixia Breaking Point Chassis
User	String		Username for Ixia Breaking Point Chassis

Ixia Breaking Point Module Attributes

The attribute names and types are listed in the following table:

Attribute	Type	Default value	Description
Memory in GB	Numeric	8	Amount of memory to set for the VM
Number of Ports	Numeric	2	Amount of ports to set for the VM
Number of vCPU	Numeric	4	Amount of virtual CPU to set for the VM
Password	String		Password for Ixia Breaking Point Chassis

Attribute	Type	Default value	Description
Slot Id	Numeric	0	Chassis Slot number on which vBlade will be assigned
User	String	0	Username for Ixia Breaking Point Chassis
Virtual Traffic Generator Chassis	String		Name of Virtual Ixia Breaking Point Chassis application

Virtual Port Attributes

The attribute names and types are listed in the following table:

Attribute	Type	Default value	Description
Logical Name	String		The port's logical name in the test configuration
Requested vNIC	String		

Typical workflow and scenarios

Typical workflow

NA

Use cases and scenarios

Scenario 1 – Getting test file with network configuration

You cannot change predefined Tests and Network Neighborhoods. Predefined Network Neighborhoods will not be included in Test files. This scenario will help you to use predefined Tests and Network Neighborhoods.

1. *Duplicating Breaking Point Network Neighborhood configuration.*
 - Open Breaking Point UI
 - Go to **CONTROL CENTER>Open Neighborhood**
 - Find and select Network Neighborhood from the list
 - Press **Save As** and enter **New Network Neighborhood Name, Ok**
2. *Duplicating Breaking Point Test.*
 - Go to **TEST>Open Test**
 - Find and select **Test** from the list
 - Press **Save As** and save it with new name.
3. *Changing Network Neighborhood in the duplicated test.*
 - Find and select duplicated Test from the list and open it.
 - In the section Network Neighborhood press ‘...’, find and select duplicated Network Neighborhood.
 - Press **Save**
4. *Running **GetTestFile** BreakingPointController command.*
 - Enter into the Cloudshell, newly created environment and reserve it.
 - Run BreakingComandController service command **GetTestFile** with duplicated test name.
 - Enter to the folder specified in the attribute “**Test Files Location**”+<reservation_id>, and you can see file with name of your duplicated test with extension “bpt”

Scenario 2 – Running test

1. *Enter into your reservation.*
2. *Running BreakingPointController service Load Configuration command.*
 - Press BreakingPointController **Commands** and enter into the service commands
 - Find Load Configuration command and enter to run menu
 - Specify **Breaking Point config file** with value of path to you test configuration file. It can be full path, or relative path under the location specified in the attribute "**Test Files Location**", like "<reservation_id>/test_name.bpt", or only "test_name.bpt" if it is current reservation.
 - Press **Run**, it will load test and network configuration from this file and reserve necessary ports
3. *Running **Start Test***
 - Press BreakingPointController **Commands** and enter into the service commands
 - Find **Start Traffic** command and enter to run menu
 - Specify **Blocking** to true, if you have to wait till test will be finished, or false and press Run
4. *Running **Stop Test***

If you ran test with "Blocking" false you can immediately stop the test

 - Run command **Stop Traffic**
5. *Getting result file*
 - Run command **Get Result**
 - You can find result file attached to the reservation

Release notes

NA

References

Additional technical documentation is available in the [Quali Download Center](#).

For Quali discussion forums, you can access the [Quali Support Portal](#).