Quali



IxLoad Controller Shell

Release date: August 2017

Shell version 1.2.3

Document version A

***Legal notice***

*Information in this document is subject to change without notice. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Quali Ltd. Quali may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except if expressly provided in any written license agreement from Quali, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Quali, CloudShell, CloudShell Authoring, CloudShell Resource Manager, CloudShell Remote Runner, CloudShell Runtime, CloudShell Monitor, CloudShell Spy, CloudShell Portal, the Quali logo, the CloudShell logo, and the CloudShell application logos, and all other Quali product names and logos are trademarks or registered trademarks of Quali Ltd. The absence of a trademark from this list does not constitute a waiver of Quali intellectual property rights concerning that trademark.*

*All other trademarks, brand and product names are property of their respective holders.*

*© 2016 Quali Ltd. All rights reserved.*

Contents

[Overview 3](#_Toc488307038)

[About IxLoad Controller Shell 3](#_Toc488307039)

[Standard version 3](#_Toc488307040)

[Supported OS 3](#_Toc488307041)

[Requirements 3](#_Toc488307042)

[Downloading the Shell 3](#_Toc488307043)

[Import and Configure the Shell 4](#_Toc488307044)

[Importing the Shell into CloudShell 4](#_Toc488307045)

[Offline installation of a Shell 4](#_Toc488307046)

[Configuring a new service 5](#_Toc488307047)

[Updating Python Dependencies for Shells 6](#_Toc488307048)

[Updating offline Python dependencies 6](#_Toc488307049)

[Updating online Python dependencies 6](#_Toc488307050)

[Typical workflow 7](#_Toc488307051)

[Scenario 1 – Use a controller to run IxLoad traffic 7](#_Toc488307052)

[References 10](#_Toc488307053)

[Release notes 11](#_Toc488307054)

# 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 automation that enables interaction with the device via CloudShell.

## About IxLoad Controller Shell

This Shell provides you with connectivity and management capabilities such as loading configuration, running traffic and getting results for IxLoad application.

## Standard version

The IxLoad Controller Shell 1.2.3 is based on the Traffic Shell standard version 3.0.0.

For detailed information about the Shell’s structure and attributes, see the Traffic Shell standard on [cloudshell-standards repository](https://github.com/QualiSystems/shell-traffic-standard) in GitHub.

## Supported OS

* Windows

## Requirements

* CloudShell version 7.1 and above
* IxLoad client should be installed on the ES machine.

## Downloading the Shell

The IxLoad Controller Shell is available from the [Quali Developer Center](http://community.quali.com/spaces/12/index.html?__hstc=46213176.aaafbe5adb338215377a985e0c025079.1467146361756.1471392182746.1471395614692.11&__hssc=46213176.1.1471395614692&__hsfp=2437115919)[.](https://support.qualisystems.com/entries/87063688-Solution-Pack-Download-Center) Download the files into a temporary location on your local machine.

The Shell comprises:

|  |  |
| --- | --- |
| ixia\_IxLoad\_controller.zip | The Shell Package. |
| ixia\_IxLoad\_controller\_offline\_requirements.zip | Shell Python dependencies (**for offline installation only**) |
| IxLoad Controller Shell Doc.pdf | Documentation |

# Import and Configure the Shell

This section describes how to import, configure and modify the IxLoad Controller 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 Developer Center](http://community.quali.com/spaces/12/index.html?__hstc=46213176.aaafbe5adb338215377a985e0c025079.1467146361756.1471392182746.1471395614692.11&__hssc=46213176.1.1471395614692&__hsfp=2437115919)[.](https://support.qualisystems.com/entries/87063688-Solution-Pack-Download-Center)
2. Back up your database.
3. Log in to **CloudShell Portal** as administrator and access the relevant domain.
4. From the **User** menu, select **Import Package.**



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 apply only if Cloudshell Execution Server has no access to PyPi. You can skip this section if your execution server has access to PyPi. *For additional information, see the online help topic on offline dependencies.*

The Shell uses a variety of Python packages.

**To work in offline mode:**

1. Download the ixia\_chassis\_shell\_offline\_requirments.zip 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 machine, in the customer.configfile, add the following key:

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

Make sure to update the value with the path to the repository containing the unzipped file.

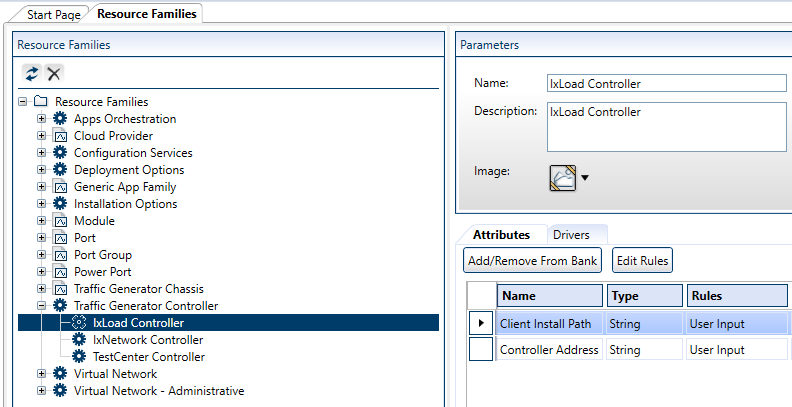
1. Restart the Execution Server.

## Configuring a new service

Perform this procedure to configure the service after importing the Shell.

**To configure the service:**

1. Go to the **Admin** tab and select the **Resource Families**.
2. Under **Traffic Generators Controllers**, select **IxLoad Controller**.



1. Define default values for the **IxLoad Controller** service.

|  |  |
| --- | --- |
| Name | Description |
| Client Install Path | The path in which the traffic client is installed on the Execution Server. For example 'C:/Program Files (x86)/Ixia/IxOS/8.01-GA'. |
| Controller Address | The IP address of the API server. Default is localhost. |

# Updating Python Dependencies for Shells

This section guides through on how to update your Python dependencies folder. This is required when you upgrade a Shell, driver that has new/updated dependencies. It applies to both online and offline dependencies.

## Updating offline Python dependencies

**To update offline Python dependencies:**

1. Download the latest Python dependencies package zip file locally.
2. Extract the zip file to the suitable offline package folder(s).
3. Restart any Execution Server that has a live instance of the relevant driver or script. This requires running the TestShell Execution Server's configuration wizard, as explained in the Configure the TestShell Execution Server topic of the CloudShell Suite Installation Guide - see the [CloudShell Docs & Training](http://www.quali.com/community/training/) page.

## Updating online Python dependencies

In online mode, the Execution Server automatically downloads and extracts the appropriate dependencies file to the online Python dependencies repository every time a new instance of the driver or script is created.

**To update online Python dependencies:**

* If there is a live instance of the Shell's driver or script, restart the execution server, as explained above. If an instance does not exist, the execution server will download the Python dependencies the next time a command of the driver or script runs.

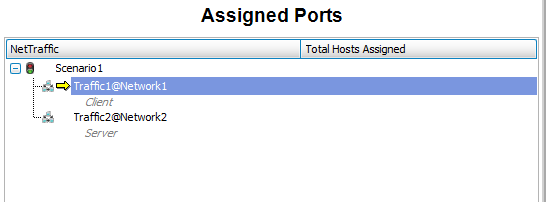
# Typical workflow

## Scenario 1 – Use a controller to run IxLoad traffic

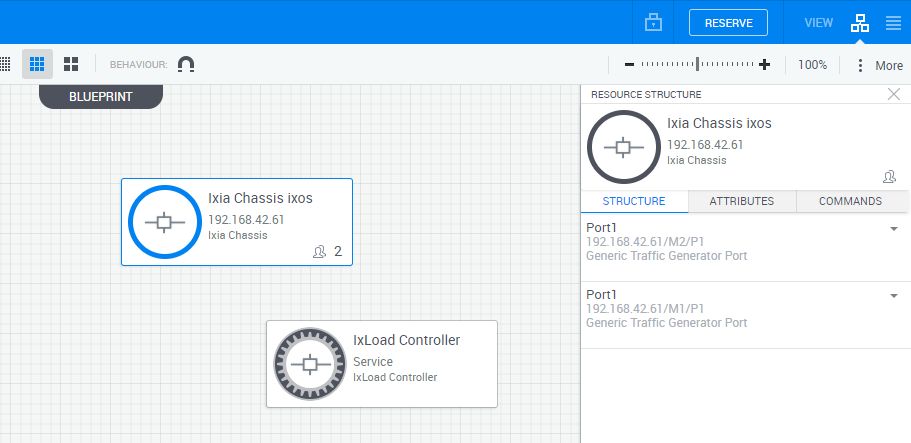
* Create Blueprint

Create a Blueprint with IxLoad controller service and IxChassus resource ports. Number of IxChariot ports should match the number of ports in the IxLoad configuration.

For example, if we have configuration with two ports:



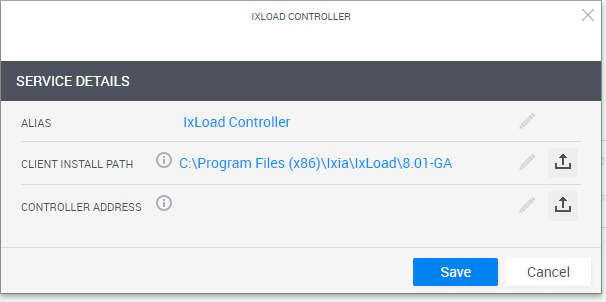
We create Blueprint with two IxLoad ports



* Reserve Sandbox

Create a Sandbox from the Blueprint.

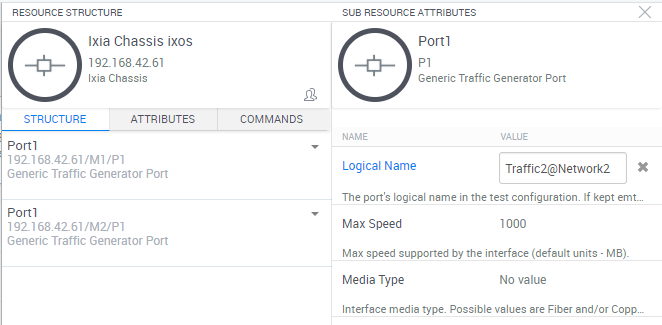
Edit IxLoad Controller Service parameters if required.



See ‘Configuring a new service’ above.

* Map configuration ports to Sandbox ports

For each port in the IxLoad configuration assign physical port from the ports in the sandbox. Open the attributes tab and set the Logical Name to the port name in the configuration.



* Controller Commands

|  |  |  |  |
| --- | --- | --- | --- |
| Command | Description | Parameters | |
|  |  | Parameter | Description |
| Load Configuration | Load configuration and reserve ports | Ixia config file name | Full path to IxLoad configuration file name (rxf) |
| Start Test | Start Test | Blocking | True - return after traffic finish to run, False - return immediately |
| Stop Test | Start Test |  |  |
| Get Statistics | Get view statistics | View Name | Name of csv file from the IxLoad results directory (under the shell logs directory). |
| Output Type | CSV or JSON. If CSV. The statistics will be attached to the reservation csv file. |

# References

Additional technical documentation is available in the [Quali's Developer Center](http://community.quali.com/spaces/12/index.html?__hstc=46213176.aaafbe5adb338215377a985e0c025079.1467146361756.1471392182746.1471395614692.11&__hssc=46213176.1.1471395614692&__hsfp=2437115919)[.](https://support.qualisystems.com/entries/22858046-download-center)

For Quali discussion forums, click [here](http://community.quali.com/spaces/13/index.html?__hstc=46213176.aaafbe5adb338215377a985e0c025079.1467146361756.1471392182746.1471395614692.11&__hssc=46213176.1.1471395614692&__hsfp=2437115919)[.](https://support.qualisystems.com/)

# Release notes

**What’s new:**

NA

**Known issue:**

* All Execution Servers that are used to run Sandboxes with IxLoad controller should have the same Client Install Path. This means that all Execution Servers must be either Windows or Linux.
* IxLoad can run up to two instances per machine (execution server). If there are more than two instances running, Load Configuration of any additional reservation will hang out and eventually fail for timeout.
* When using Linux Execution Server (not officially supported) the Shell assumes that there is a shared disk between the IxLoadGatewayService machine (Windows) and the Linux Execution sever and that disk is mounted and mapped on both machines to the following path:
  + Windows – c:\IxLoadResults
  + Linux – /IxLoadResults