



Xena Controller Shell

Release date: November 2017

Shell version 1.5.0

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.

"Xena", "Xena Networks" and the "X" logo are trademarks of Xena Networks ApS, Denmark.

© 2018 Xena Networks & Quali Limited. All rights reserved

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

Contents

Overview	3
About Xena Controller Shell	3
Standard version	3
Supported OS	3
Requirements.....	3
Downloading the Shell.....	3
Import and Configure the Shell.....	4
Importing the Shell into CloudShell.....	4
Offline installation of a Shell	4
Configuring a new service	5
Updating Python Dependencies for Shells	6
Updating offline Python dependencies	6
Updating online Python dependencies	6
Typical workflow.....	7
Scenario 1 – Use a controller to run Xena traffic	7
Release notes	11

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 Xena Controller Shell

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

Standard version

The Xena Controller Shell 1.5.0 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](#) in GitHub.

Supported OS

- Windows

Requirements

- CloudShell version 8.1 and above

Downloading the Shell

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

The Shell comprises:

Xena_controller.zip	The Shell Package.
Xena_controller_offline_requirements.zip	Shell Python dependencies (for offline installation only)
Xena Controller Shell Doc.pdf	Documentation

Import and Configure the Shell

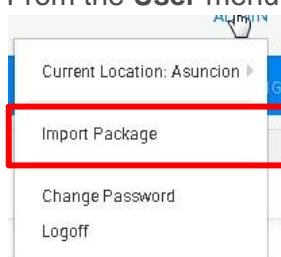
This section describes how to import, configure and modify the Xena Controller Shell.

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](#).
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 `xena_controller_shell_offline_requirements.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.config` file, 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.

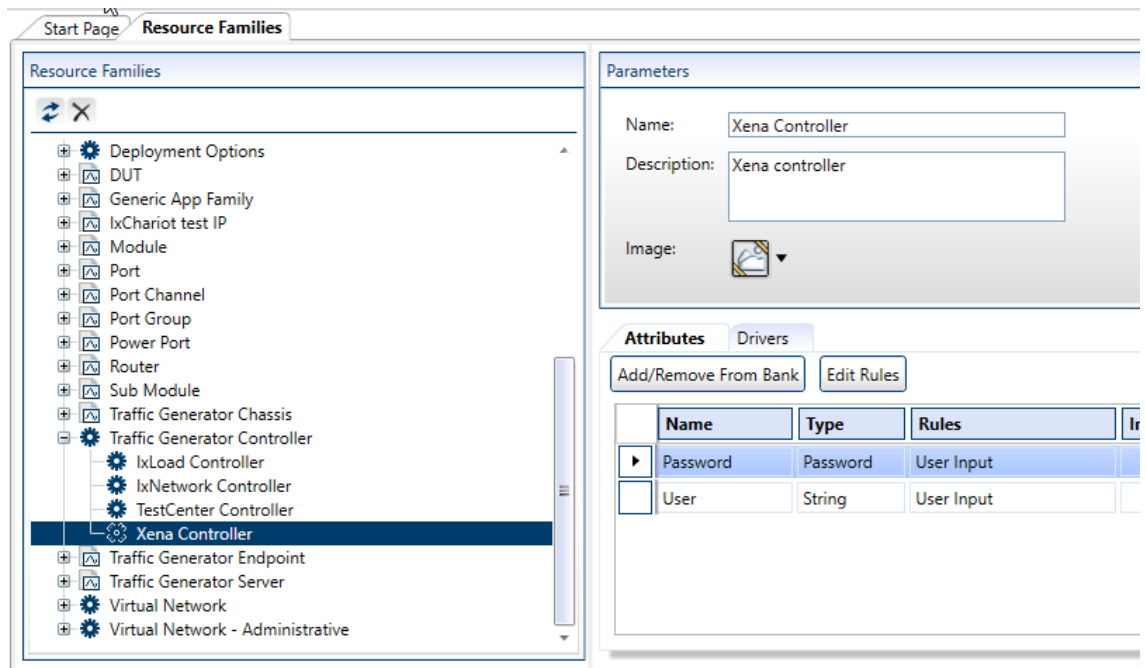
- Restart the Execution Server.

Configuring a new service

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

To configure the service:

- Go to the **Admin** tab and select the **Resource Families**.
- Under **Traffic Generators Controllers**, select **Xena Controller**.



- Define default values for the **Xena Controller** service.

Name	Description
User	Owner of the shell scripting session
Password	Chassis password
Controller Port Number	Chassis TCP port number (default=22611)

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](#) 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.

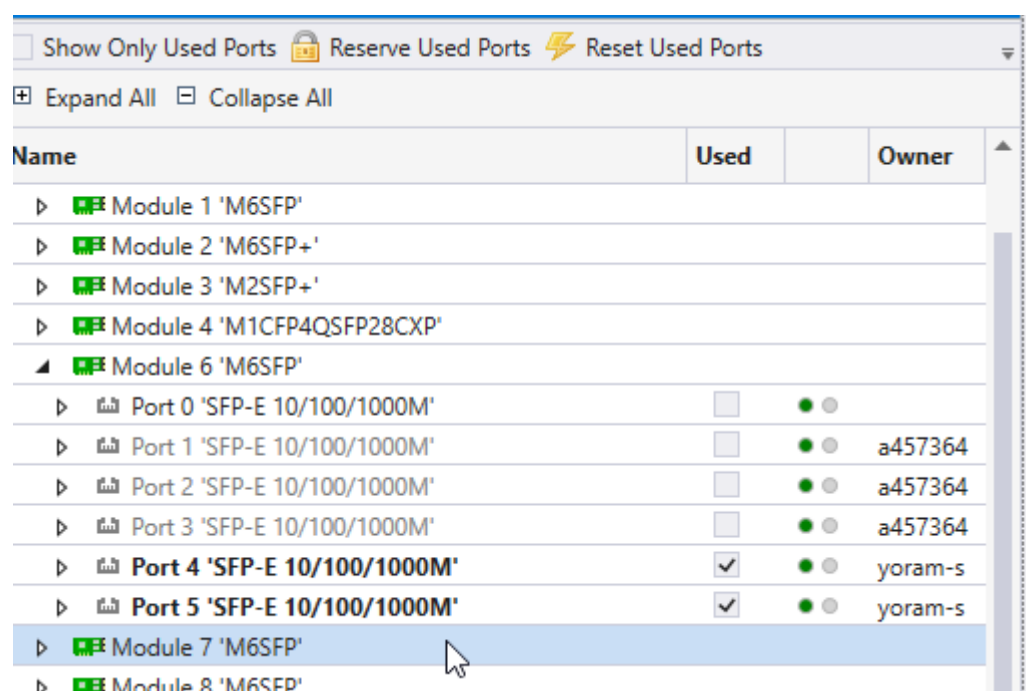
Typical workflow

Scenario 1 – Use a controller to run Xena traffic

- Create Blueprint

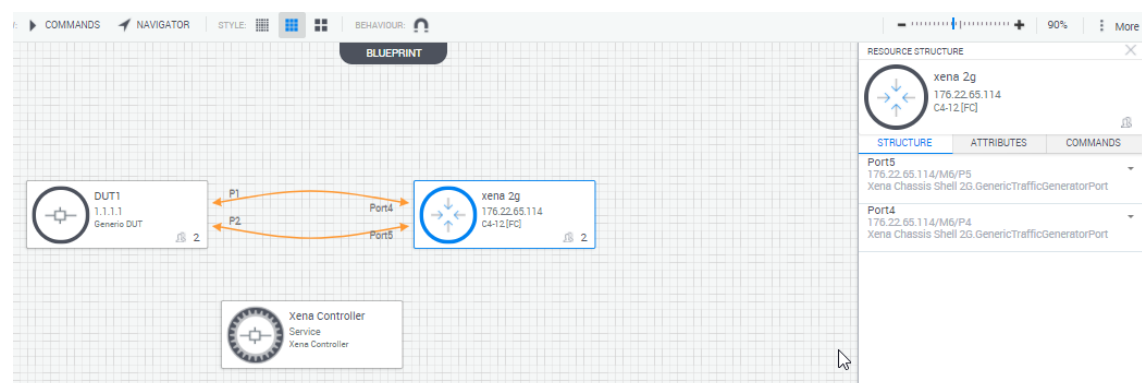
Create a Blueprint with Xena controller service and Xena chassis resource ports. Number of Xena ports in the blueprint should match the number of ports in the Xena configuration.

For example, if we have configuration with two ports:



Name	Used	Owner
▶ Module 1 'M6SFP'		
▶ Module 2 'M6SFP+'		
▶ Module 3 'M2SFP+'		
▶ Module 4 'M1CFP4QSFP28CXP'		
▶ Module 6 'M6SFP'		
▶ Port 0 'SFP-E 10/100/1000M'	<input type="checkbox"/>	
▶ Port 1 'SFP-E 10/100/1000M'	<input type="checkbox"/>	a457364
▶ Port 2 'SFP-E 10/100/1000M'	<input type="checkbox"/>	a457364
▶ Port 3 'SFP-E 10/100/1000M'	<input type="checkbox"/>	a457364
▶ Port 4 'SFP-E 10/100/1000M'	<input checked="" type="checkbox"/>	yoram-s
▶ Port 5 'SFP-E 10/100/1000M'	<input checked="" type="checkbox"/>	yoram-s
▶ Module 7 'M6SFP'		
▶ Module 8 'M6SFP'		

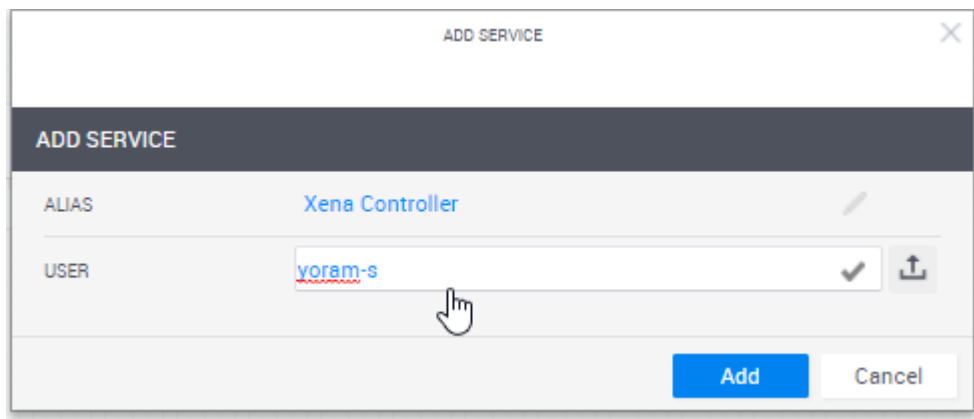
We create Blueprint with two Xena ports



- Reserve Sandbox

Create a Sandbox from the Blueprint.

Edit Xena Controller Service parameters if required.



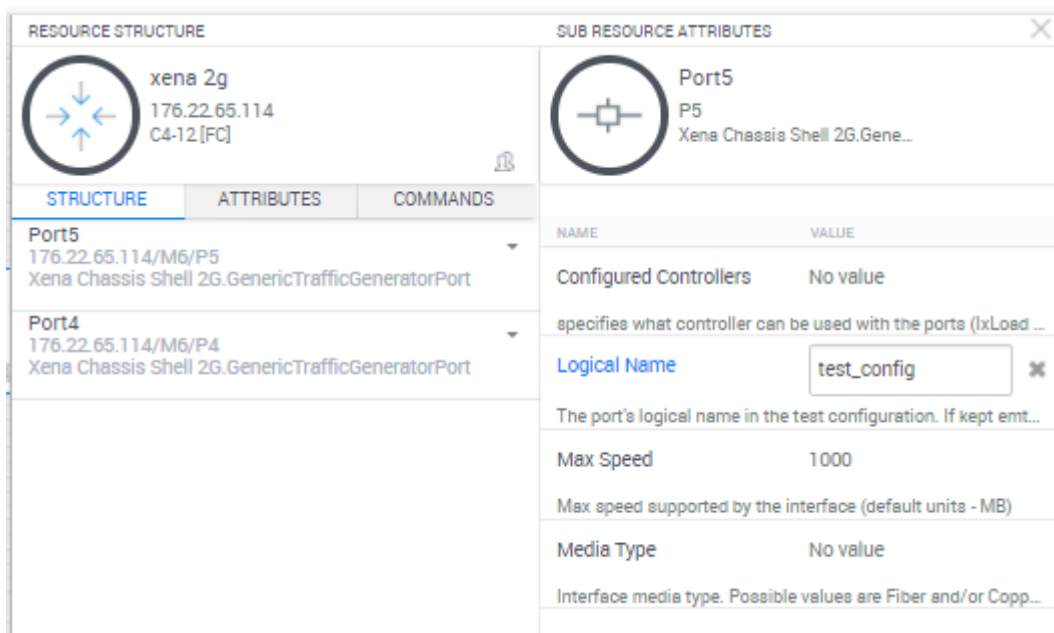
See 'Configuring a new service' above.

- Map configuration ports to Sandbox ports

For each port in the Xena configuration assign physical port from the ports in the sandbox. Open the attributes tab and set the Logical Name to the configuration file name that should be loaded onto it.

Enter the file name as base name (not full path) and without the '.xpc' extension.

Note that all configuration files should be located in the same folder.



- Controller Commands

Command	Description	Parameters	
		Parameter	Description
Load Configuration	Load configuration and reserve ports	Xena configs folder	Full path to Xena configuration files directory
Start Traffic	Start ports traffic	Blocking	True - return after traffic finish to run, False - return immediately
Stop Traffic	Stop ports traffic		
Get Statistics	Get view statistics	View Name	Port, Stream or TPLD.
		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](#).

For Quali discussion forums, click [here](#).

For technical documentation on Xena Networks' solutions, please see [Xena Resources](#).

For support, please send email to support@xenanetworks.com

Release notes

What's new:

NA

Known issue:

- Multiple sandboxes on the same execution server are not supported.