



Xena Chassis 2nd Generation Shell

Release date: January 2018

Shell version 2.0.0

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 Chassis 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 device	5
Updating Python Dependencies for Shells	6
Updating offline Python dependencies	6
Updating online Python dependencies	6
Typical workflow	7
References	8
Release notes	9

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 Chassis Shell

This 2nd generation Shell provides you with connectivity and management capabilities such as device structure discovery and power management for Xena chassis.

Standard version

The Xena Chassis Shell 2.0.0 is based on CloudShell traffic generator chassis standard 1.0.2.

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

The Shell comprises:

xena_chassis_shell_2g.zip	The Shell Package.
xena_chassis_shell_2g_offline_requirements.zip	Shell Python dependencies (for offline installation only)
xena Chassis Shell Doc.pdf	Documentation

Import and Configure the Shell

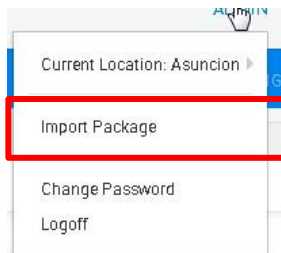
This section describes how to import, configure and modify the Xena 2nd generation Chassis Shell.

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. Backup your database.
3. Log in to **CloudShell Portal** as administrator of the relevant domain.
4. In 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 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. *For additional information, see the online help topic on offline dependencies.*

The Shell uses a variety of Python packages. To work in offline mode perform the following steps:

1. Download the xena_chassis_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"/>
```
-

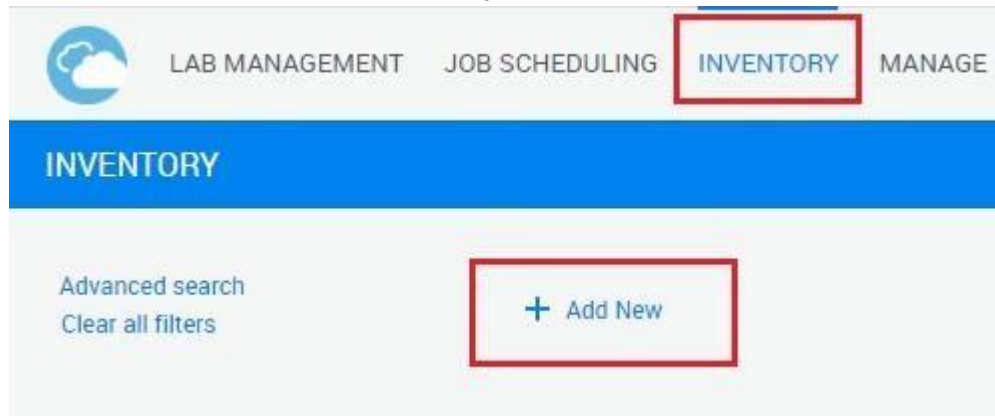
Make sure to update the value with the path to the repository containing the unzipped file.

4. Restart the Execution Server.

Configuring a new device

Use the following procedure to load a device, which will use this Shell, into CloudShell:

1. In the CloudShell Portal, in the **Inventory** dashboard, click **Add New**.



2. From the list, select the **Xena Chassis 2G Shell**.
3. Enter the Xena chassis Name and IP address.
4. Click **Create**.
5. In the **Resource** dialog box, enter the device's settings, as following:
 - a User – user name
 - b Password – chassis password
6. Click **Start Discovery**.

This command discovers the device, fills in its attributes and creates the device's structure in CloudShell (if such structure exists).

Updating Python Dependencies for Shells

This section 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 have 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 – Discover a new Device

See [Configuring a new device.](#)

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

NA