



Xena OpenAutomation Test Configuration Converter Guide

Release 2.0.1

Xena Networks

Aug 21, 2023

TABLE OF CONTENT

1	Introduction	3
2	Getting Started	5
3	API Documentation	13
4	Glossary of Terms	15
5	Indices and Tables	17
	Python Module Index	19
	Index	21

Xena OpenAutomation (XOA) Test Configuration Converter is a supporting tool for users to quickly migrate their Valkyrie test suite configurations into XOA.

Xena's test suite applications have only been for Windows platform for a long time. Moving forward, all of Xena's existing and future test suites will be included in Xena OpenAutomation, which is not limited to Windows anymore. To help users easily migrate their existing Windows test suite configurations .v2544 for [Valkyrie2544](#), .v2889 for [Valkyrie2889](#), .v3918 for [Valkyrie3918](#), and .v1564 for [Valkyrie1564](#) into [XOA](#), we have developed this test configuration converter.

The *Xena OpenAutomation Test Configuration Converter Documentation* provides information about how to install the Python package, and how to use the converter.

The target audience of this document is test specialists who develop and run automated test scripts/programs using Xena TGA hardware and software. Users of this document should have the following knowledge and experience:

- Ability to program with Python language.
- Familiarity with the operating system of your development environment.
- Familiarity with Xena test equipment.

INTRODUCTION

The XOA Test Config Converter is an open-source tool hosted on Xena Networks' GitHub repository. It is designed to help users migrate their existing Valkyrie test suite configurations into the XOA format, enabling a seamless transition to the XOA ecosystem for network automation and testing.

Key features of the XOA Test Config Converter include:

1. Conversion support: The tool supports conversion of Valkyrie test suite configuration files to XOA-compatible format, facilitating the integration of existing test cases into the XOA framework.
2. Ease of use: The XOA Test Config Converter is designed to be user-friendly, with a straightforward process for converting test suite configuration files.
3. Compatibility: The converter ensures that the migrated test suite configurations are compatible with XOA Core and can be executed within the XOA ecosystem.

The [official documentation](#) offers a comprehensive guide on how to install and use the XOA Test Config Converter. It covers topics such as setting up the environment, converting test suite configuration files, and understanding the output format.

Note: The purpose of XOA Converter is ONLY to convert Xena Valkyrie test suite applications' configuration files into XOA's configuration files. Thus only four test suite types are supported by XOA Converter as the source config files.

GETTING STARTED

2.1 Installing XOA Test Config Converter

XOA Test Config Converter is available to install and upgrade via the [Python Package Index](#). Alternatively, you can also install and upgrade from the source file.

- If you prefer installing/upgrading/uninstalling automatically, go to Section *[From PyPi Using pip](#)*.
- If you prefer installing/upgrading manually, go to Section *[Manually From Source](#)*.

2.1.1 Prerequisites

Before installing XOA Test Config Converter, please make sure your environment has installed [Python](#) and [pip](#).

Python

XOA Test Config Converter requires that you [install Python](#) on your system.

Note: XOA Test Config Converter requires Python ≥ 3.8 .

pip

Make sure [pip](#) is installed on your system. [pip](#) is the [package installer for Python](#) . You can use it to install packages from the [Python Package Index](#) and other indexes.

Usually, [pip](#) is automatically installed if you are:

- working in a [virtual Python environment](#) ([virtualenv](#) or [venv](#)). It is not necessary to use `sudo pip` inside a virtual Python environment.
- using Python downloaded from [python.org](#)

If you don't have [pip](#) installed, you can:

- Download the script, from <https://bootstrap.pypa.io/get-pip.py>.
- Open a terminal/command prompt, cd to the folder containing the `get-pip.py` file and run:

Listing 1: Install pip in Windows environment.

```
> py get-pip.py
```

Listing 2: Install pip in macOS/Linux environment.

```
$ python3 get-pip.py
```

See also:

Read more details about this script in [pypa/get-pip](#).

Read more about installation of pip in [pip installation](#).

2.1.2 From PyPi Using pip

Install

pip is the recommended installer for XOA Test Config Converter. The most common usage of pip is to install from the [Python Package Index](#) using [Requirement Specifiers](#).

Listing 3: Install XOA Test Config Converter in Windows environment from PyPi.

```
> pip install xoa-converter          # latest version
> pip install xoa-converter==1.0.1   # specific version
> pip install xoa-converter>=1.0.1   # minimum version
```

Listing 4: Install XOA Test Config Converter in macOS/Linux environment from PyPi.

```
$ pip install xoa-converter          # latest version
$ pip install xoa-converter==1.0.1   # specific version
$ pip install xoa-converter>=1.0.1   # minimum version
```

Upgrade

To upgrade XOA Test Config Converter package from PyPI:

Listing 5: Upgrade XOA Test Config Converter in Windows environment from PyPi.

```
> pip install xoa-converter --upgrade
```

Listing 6: Upgrade XOA Test Config Converter in macOS/Linux environment from PyPi.

```
$ pip install xoa-converter --upgrade
```

Uninstall

To uninstall XOA Test Config Converter using pip:

Listing 7: Uninstall XOA Test Config Converter in Windows environment.

```
> pip uninstall xoa-converter
```

Listing 8: Uninstall XOA Test Config Converter in macOS/Linux environment.

```
$ pip uninstall xoa-converter
```

See also:

For more information, see the [pip uninstall](#) reference.

2.1.3 Manually From Source

Install or Upgrade

If for some reason you need to install or upgrade XOA Test Config Converter manually from source, the steps are:

First, make sure Python packages [wheel](#) and [setuptools](#) are installed on your system. Install [wheel](#) and [setuptools](#) using [pip](#):

Listing 9: Install [wheel](#) and [setuptools](#) in Windows environment.

```
> pip install wheel setuptools
```

Listing 10: Install [wheel](#) and [setuptools](#) in macOS/Linux environment.

```
$ pip install wheel setuptools
```

Then, download the XOA Test Config Converter source distribution from [XOA Test Config Converter Releases](#). Unzip the archive and run the `setup.py` script to install the package:

Listing 11: Install XOA Test Config Converter in Windows environment from source.

```
> python setup.py install
```

Listing 12: Install XOA Test Config Converter in macOS/Linux environment from source.

```
$ python3 setup.py install
```

If you want to distribute, you can build `.whl` file for distribution from the source:

Listing 13: Build XOA Test Config Converter wheel in Windows environment for distribution.

```
> python setup.py bdist_wheel
```

Listing 14: Build XOA Test Config Converter wheel in macOS/Linux environment for distribution.

```
$ python3 setup.py bdist_wheel
```

2.2 Convert Valkyrie Test Config into XOA

To converter your old test configuration files into new XOA json format, you need to do the following steps.

2.2.1 Step 1. Create Project Folder

First, create a folder on your computer at a location you want. This folder will be the place where you keep your XOA test suites and a simple Python program to load and run them using XOA Core framework.

Let's create a folder called `/my_xoa_project`

Listing 15: Create project folder

```
/my_xoa_project  
|
```

2.2.2 Step 2. Create Necessary Files

Create a `main.py` file inside the folder `/my_xoa_project`.

Then, on the same level as `main.py`, create a folder `/pluginlib` for keeping your test suites.

After that, create a `__init__.py` inside folder `/pluginlib` to make it into a [package](#).

Listing 16: Create necessary files

```
/my_xoa_project
|
|- main.py
|- /pluginlib
|   |- __init__.py
|   |
```

2.2.3 Step 3. Install XOA Core

If you have already installed XOA Core in your system, either to your global namespace or in a virtual environment, you can skip this step.

Read more about ``

2.2.4 Step 4. Copy XOA Test Suite Plugin into Project Folder

Copy a test suite plugin, e.g. /plugin2544 from [XOA Test Suite](#) into /my_xoa_project/pluginlib.

Copy your test configuration json file, e.g. my2544_data.json into /my_xoa_project for easy access.

Listing 17: Copy test suite plugin into project

```
/my_xoa_project
|
|- main.py
|- my2544_data.json
|- /pluginlib
|   |- __init__.py
|   |- /plugin2544
```

2.2.5 Step 5. Convert Test Config from Valkyrie to XOA

First, import `xoa_core` and `xoa_converter` into your Python code. If you haven't installed `xoa-core` Python package in your environment, please go to [XOA Core](#).

```
from xoa_core import controller
from xoa_converter.entry import converter
from xoa_converter.types import TestSuiteType
```

Then, to use the converter, you need the target schema for the old file to be converted into. After that, simply provide the target schema, the old config file to the converter function and get the new config file:

```

import asyncio
import json
from xoa_core import controller
from xoa_converter.entry import converter
from xoa_converter.types import TestSuiteType

# source config file to be converted
OLD_CONFIG_FILE = "my_old2544_config.v2544"
NEW_CONFIG_FILE = "xoa2544.json"
T_SUITE_NAME = "RFC-2544"
PLUGINS_PATH = "pluginlib"

async def start():

    # create an instance of xoa core controller
    c = await controller.MainController()

    # open and read valkyrie test config my_old2544_config.v2544
    with open(OLD_CONFIG_FILE) as f:
        app_data = f.read()

    # register rfc2544 plugin to core
    c.register_lib(str(PLUGINS_PATH))

    # get rfc2544 test suite information from the core's
    ↪registration
    info = c.get_test_suite_info(T_SUITE_NAME)
    if not info:
        print("Test suite is not recognized.")
        return None

    # convert the old config into new config
    new_data = converter(TestSuiteType.RFC2544, app_data, info[
    ↪"schema"])

    # save the new config file xoa2544.json
    with open(NEW_CONFIG_FILE, "w") as f:
        f.write(new_data)

    # you can use the config file below to start the test
    new_config = json.loads(new_data)

    # Test suite name: "RFC-2544" is received from call of c.get_
    ↪available_test_suites()
    # test_id = c.start_test_suite(T_SUITE_NAME, new_config)

if __name__ == '__main__':

```

(continues on next page)

(continued from previous page)

```
loop = asyncio.get_event_loop()
loop.create_task(start())
loop.run_forever()
```


API DOCUMENTATION

3.1 Converter

3.2 Types

```
class TestSuiteType(value, names=None, *, module=None, qualname=None,  
                   type=None, start=1, boundary=None)
```

Bases: [Enum](#)

Four test suite types supported by XOA Converter as the source config files.

RFC2544 = 'RFC-2544'

RFC2889 = 'RFC-2889'

RFC3918 = 'RFC-3918'

Y1564 = 'Y-1564'

GLOSSARY OF TERMS

API

Application Programming Interface.

Valkyrie1564

[Valkyrie1564](#) provides full support for both the configuration and performance test types described in Y.1564. It is installed together with ValkyrieManager and uses the same terminology. The simple intuitive GUI makes it easy to connect one or more ValkyrieCompact and/or ValkyrieBay chassis for testing Layer 2 and Layer 3.

Valkyrie2544

[Valkyrie2544](#) offers full support for the 4 test-types specified in RFC2544, and also lets you partially enable one or more test types. Valkyrie2544 supports different network topologies and traffic flow directions on both Layer 2 and Layer 3, as well as both IPv4 and IPv6.

Valkyrie2889

[Valkyrie2889](#) is a free application for benchmarking the performance of Layer 2 LAN switches.

Valkyrie3918

[Valkyrie3918](#) provides an easy-to-use port configuration panel that lets you add and remove ports, and assign IP addresses and port roles. Ports from multiple ValkyrieBay and ValkyrieCompact chassis can be freely mixed.

XOA

Xena OpenAutomation

XOA Core

[XOA Core](#) is the XOA test suite framework to host different XOA Test Suites as its plugins.

INDICES AND TABLES

- `genindex`
- `search`

PYTHON MODULE INDEX

X

`xoa_converter.types`, [13](#)

INDEX

A

API, [15](#)

M

module

`xa_converter.types`, [13](#)

R

RFC2544 (*TestSuiteType* attribute), [13](#)

RFC2889 (*TestSuiteType* attribute), [13](#)

RFC3918 (*TestSuiteType* attribute), [13](#)

T

TestSuiteType (class in `xa_converter.types`), [13](#)

V

Valkyrie1564, [15](#)

Valkyrie2544, [15](#)

Valkyrie2889, [15](#)

Valkyrie3918, [15](#)

X

XOA, [15](#)

XOA Core, [15](#)

`xa_converter.types`

 module, [13](#)

Y

Y1564 (*TestSuiteType* attribute), [13](#)