**XIQ WiNG Migration Guide**

by Tim Smith, SA – 06/22/2023 – v1.3

Overview**:**

This guide covers how to run the XIQ\_wing\_migrate.py script. This script can migrate the location hierarchy, rf-domains, and floors from WiNG to ExtremeCloud IQ (XIQ). This will save time by creating the location(s), building(s), and floor(s) and setting the location of the Access Points all from data within a WiNG Tech-Dump file.

Target Audience**:** Semi-Technical

**Icon

Description automatically generated**

# Definitions

|  |  |
| --- | --- |
| XIQ | ExtremeCloud IQ |
| VIQ | Virtual ExtremeCloud IQ |

# Table of Contents

[Overview 1](#_Toc113371916)

[Target Audience 1](#_Toc113371917)

[Definitions: 1](#_Toc113371918)

[Table of Contents 2](#_Toc113371919)

[Prerequisites: 3](#_Toc113371920)

[Scripting Environment Preparation: 3](#_Toc113371921)

[Information: 3](#_Toc113371922)

[Device Choice: 3](#_Toc113371923)

[Python Installation: 3](#_Toc113371924)

[Mac OSX Big Sur 4](#_Toc113371925)

[Required Modules: 4](#_Toc113371926)

[Checking for existing Modules 4](#_Toc113371927)

[Installing required modules 4](#_Toc113371928)

[4](#_Toc113371929)

[Geo Coordinates: 5](#_Toc113371930)

[Information: 5](#_Toc113371931)

[Access to XIQ: 5](#_Toc113371932)

[WiNG Tech Dump File: 6](#_Toc113371933)

[Information: 6](#_Toc113371934)

[Buildings: 6](#_Toc113371935)

[Locations: 6](#_Toc113371936)

[Floors: 6](#_Toc113371937)

[APs: 7](#_Toc113371938)

[Running The Script: 8](#_Toc113371939)

[Locations, buildings, floors & floorplans 9](#_Toc113371940)

[Access Points 9](#_Toc113371941)

[Arguments 9](#_Toc113371942)

[Log File 10](#_Toc113371943)

# Prerequisites:

* ExtremeCloud IQ Public Cloud or Private Cloud instance (IQVA on-prem is not supported.)
* Knowledge of XIQ by onboarding access points; creating locations, buildings, and floors; naming and setting locations of access points; Uploading and assigning floorplans to floors
* RadSec Proxy requires TCP Port 2083 to be open on your internet firewall.
* One or more XIQ native access points or campus-based Wi-Fi systems (WiNG or IdentiFi)
* Download XIQ\_Wing\_location script files from GitHub.

* + <https://github.com/timjsmith24/XIQ_Wing_location_migration>
  + wing\_migrate.py
  + app folder
    - Wing\_importer.py
    - mapImportLogger.py
    - xiq\_exporter.py
  + Wing Tech-dump file
    - (Versions tested)
      * 7.5.1.6-001R
      * 7.5.1.4-016R

# Scripting Environment Preparation:

### Information:

The XIQ\_wing\_migrate.py script requires, at minimum, Python 3.8 and tested up to Python 3.11. This script has only been tested with macOS but may be able to be executed from any device with Python and the needed modules installed. This script must be able to access the Wing Tech dump file and reach out to ExtremeCloud IQ.

The script, when run, will create a *map\_importer.log* file in the app folder. This log file will show information about locations, buildings, and floors being created and the APs throughout the migration process. Any API errors experienced will also show up in the log file.

## Device Choice:

This script has only been tested in MacOS, both Monterey and Ventura, and Windows, both versions 10 and 11. This script may be executed from any device that can run Python 3.8 or higher but is currently only supported once testing is completed. The device will need to be able to reach ExtremeCloud IQ. This can be done through a proxy. Proxy config is beyond the scope of this guide.

## Python Installation:

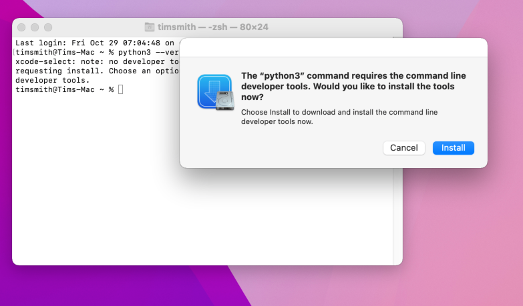
Depending on the device used, you may need to install Python or a different version of Python. The easiest way to check the version of Python is to open the terminal and type this command.

python3 --version

Below is an example of installing python3 for Mac OSX.

### 

### Mac OSX Monterey

* Open the terminal and enter *python3 --version*
  + This triggers the installation of Developer Tools
* Click Install
* Click Agree

This will install Python version 3.8.9

* The Developer tools that installed python3 will also install pip3 in Monterey.
* Mac terminal will be used to install Python modules
* Running this command will check if pip is installed

pip3 --version

## Required Modules:

The **requests, pandas,** and **textfsm** modules are required for the XIQ\_wing\_migration.py script.

### Checking for existing Modules

You can check if the required modules are installed using the terminal. For each module, run the following command.

python3 -c “import requests”

python3 -c “import pandas”

python3 -c “import textfsm”

The module is not installed if a ‘*ModuleNotFoundError: No module named '<module name>*' error is returned.

### Installing required modules

The required modules can be installed using pip3 using the downloaded requirements.txt file with the following command.

## 

pip3 install -r requirements.txt

Or the modules can be installed individually using.

pip3 install requests==2.31.0

pip3 install pandas==2.0.2

pip3 install textfsm==1.1.3

# GeoCoordinates:

## Information:

Graphical user interface, text, application

Description automatically generatedWiNG allows Geo Coordinates to be added to an rf-domain. Currently, XIQ doesn’t support Geo Coordinates; physical addresses can be used for buildings. If geo coordinates are configured in the rf-domain, the script can convert them to the physical addresses and assign that address to the building when creating it in XIQ. To do this, an API call must be made to a third-party location service at [platform.here.com](https://platform.here.com/). To leverage this, an API token will need to be created. There is a free tier that allows up to 1000 requests per day.  
Detailed instructions on creating the API token will be added in the full document, but for now, follow these steps.

1. Go to [account.here.com](http://account.here.com/) and sign up for an account.
2. Once registered on [platform.here.com,](https://platform.here.com/) click the launcher on the top right of the screen and select Access Manager.
3. At the top select Apps, then 'Register a new app.'
4. Give the app a name like 'reverse geo coords' and select 'Register.'
5. Under credentials in the newly created app, select API Keys, then click 'Create API key.'
6. Copy the created API key and add it between the quotes on line 17 of the XIQ\_wing\_migrate.py script.

If no API token exists or the geo-coordinates are not configured in the rf-domains, the buildings will be created with 'Unknown Address' in XIQ.

# Access to XIQ:

By default, the script, when run, will prompt the user for XIQ credentials. This User will have to have administrative access to the XIQ instance, as it will need to have the ability to create location elements. This user account must be a local account on the XIQ instance or have external access. To use an External XIQ account, a flag must be added when running the script. See the -external flag under the [Arguments](#_Arguments) Section for more details.

**Disclaimer: SAML accounts are currently not supported by our API.**

The script does support hardcoding an API access token between the single quotes on line 18. When the token is added to the script, the script will bypass asking for the XIQ username and password and attempt to use the provided token.

For information on generating an API token, see the **Generating Specific Tokens** section in the [A Guide to Getting Started with v2 APIs in XIQ](https://extremeportal.force.com/ExtrArticleDetail?an=000102173) Article. Note that the needed permissions for this script include locations, device, and device:list permissions. It is recommended to add the auth:r permission as well to view information about the token later, but this is not required for the script.

# WiNG Tech Dump File:

## Information:

The XIQ\_Wing\_migrate.py script will use the rf-domain information from the tech dump file to create the needed location, building, and floors in XIQ. The script will replicate the WiNG rf-domain System tree hierarchy in XIQ.

If a location, building, or floor exists in XIQ with the same name used in the WiNG Tech dump, that location element will be used instead of creating a new element.

## Buildings:

A building will be created in XIQ for each rf-domain with associated devices in the tech dump. If no devices are associated with the rf-domain, that building (rf-domain) will not be created in XIQ.

## Locations:

Each element in the rf-domains tree-node will be created in XIQ following the same hierarchy as the tree-node. With a region under the country, a city under the region, and a campus under the city. The hierarchy remains true even if an element is not included in the tree node. For example, the campus would be under the region if no city is defined.

In XIQ, each location will need to have its own unique name. This was not the case with WiNG. The script will append the parent’s name to the child to handle this. For example, if the same city name is used with different regions (or states) like Portland, Maine, and Portland, Oregon, the first city in the config will remain unchanged. The second city would become Portland\_Oregon. If this change does occur, there will be a message in the log about the change.

If any location already exists in XIQ with any of these names, the script will use that location even if the hierarchy above does not match.

## Floors:

Any floors created in an rf-domain will be added to XIQ. This config would be lines with layout area xxxx floor xxx in the rf-domain. The floor will also be created based on floors configured in any devices associated with the rf-domain. This config would be lines with floor “xxxx” in the device config. If neither of these configurations exists (floors not used in WiNG), then a **floor1** will be created, and all devices associated with the rf-domain will be placed there.

## APs:

In XIQ, the APs will be assigned to the floor based on the rf-domain (building in XIQ) and the floor in the WiNG device config. The first floor pulled from the rf-domain config will be used if there is no floor in the device config. If no floors are configured in WiNG (neither in device config nor rf-domain config), the AP will be assigned the **floor1** floor created under the rf-domain.

The devices will be spaced out in the “planning view” in XIQ and not just stacked up in the corner.

# Running The Script:

To run the script, open the terminal to the location of the script and run the following:

Python3 XIQ\_wing\_migrate.py

chmod +x XIQ\_wing\_migrate.py

You can also make the script executable by running

Then you can run the script by typing

./XIQ\_wing\_migrate.py

The script will ask you to enter a Wing tech-dump file. You can either enter the file's name, including the full path, or in macOS, you can simply drag the file into the terminal and hit enter. The script will collect the needed information from the Wing Tech dump file.

* **Note:** This process can take a few minutes, depending on the size of your config.

Once complete, the script will display how many rf-domains and APs were found.

You will then be asked if you would like to preview the data gathered.

Text

Description automatically generatedText

Description automatically generated

If you choose to preview, the script will print a JSON-type file where you can see the discovered hierarchy. Here you can see how the data will be added to XIQ. Buildings will be added under a BUILDINGS object. Each building will be an object with a FLOORS object listing the floors associated with it as well as the address of the building. You will also see the device count for each floor.

Once complete, the script will ask for your XIQ username and password.

* **Note:** The User account will need an administrator role to create the location, buildings, and floors.
* **Note:** If an API access token is added to the script, the script will bypass asking for your XIQ username and password and create the Location elements.

### 

### Text Description automatically generatedLocations, buildings, floors & floorplans

Messages will appear in the terminal window as locations, buildings, and floors are created. If a location exists with the same name, a message will be displayed that the location was found and will be used.

* **Note**: XIQ requires each location and building to have a unique name. Floors within a building also must have their unique name.
* **Note:** Currently, the script must collect the whole location tree from XIQ to validate if a location, building, or floor exists. The script does this recursively, so the script could take a long time, depending on the size of the location tree. There is a feature request open to be able to search for a location by name. Once API is available, the script will be updated to leverage that API call and speed up this process.

All Locations will be created first, followed by all buildings and floors.

If a name is longer than 32 characters, the script will print the name with a message that it is longer than 32 characters and allow you to enter a new name.



### Access Points

After creating all the locations, buildings, and floors, the script will move any WiNG APs found in the tech dump and XIQ to the correct floors. A message will display the count of APs found vs. those in the WiNG Tech dump. The messages will also inform that the script is moving and when it finishes.



## Arguments

Two optional arguments can be added when running the script.

--external

This flag allows you to import the WiNG locations, buildings, and floors into a VIQ on which you are an external user.

To run the script on an externally managed VIQ in the terminal, you would run.

Python3 XIQ\_wing\_migrate.py --external

The script will start as usual, request a WiNG Tech-dump file, and then ask for your XIQ login. After logging in, you will be presented with a choice of which VIQ you want to import into.

Text

Description automatically generated

--noaplog

This flag will suppress the log messages normally created when devices do not have the floor assigned to them in their config. If you typically do not assign a floor to the device, the log file could fill up with warning messages about APs not being set to a floor and the script assigning them to 'floor1'.

Python3 XIQ\_wing\_migrate.py --noaplog

Both flags can be added if needed.

## Log File

Upon running the script, a log file will be created named *map\_importer.log*. This log file contains the same type of information printed on the screen. It is also an excellent place to check if any issues arise as there sometimes is more information in the log file.

Text

Description automatically generated