DN Recording Checker

# Overview

Tool to check the recording configuration for a list of DNs specified in a CSV file, such as a user export from the call recording application.

Requires Python 3.6+ to run, many Linux distros have Python installed by default. For Windows the easiest install is the official Python Windows version, or Miniconda works fine too:

Miniconda distribution of Python: <https://conda.io/miniconda.html>

Official Python distribution: <https://www.python.org/downloads/>

The lxml, Requests, urllib3 and Zeep libraries are required to work.

# Version History

(c) 2018 - 2020, Chris Perkins:

v1.3 – added checking application user device association, improved handling of multiple recording profiles.

v1.2 – code tidying.

v1.1 – fixes some edge cases.

v1.0 – initial release.

All testing was done using Windows with CUCM v11.5.

# Using the Tool

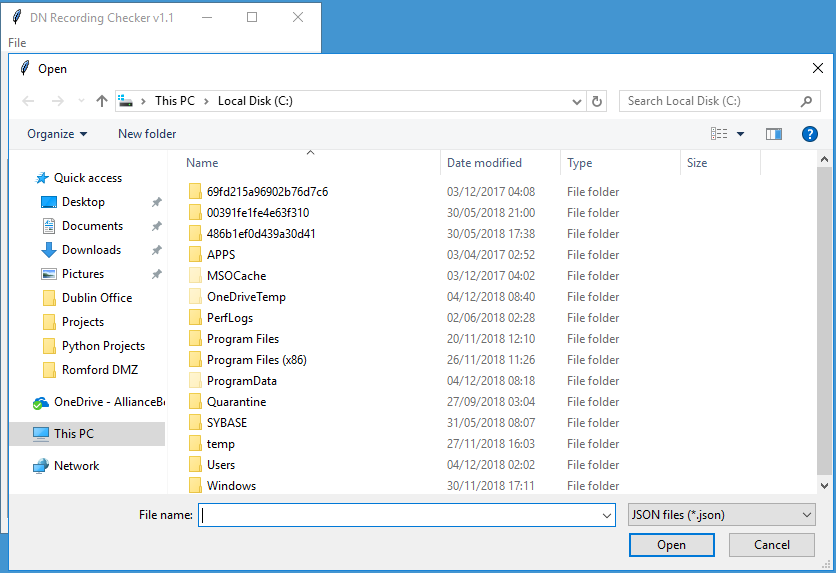
For a list of DNs in a CSV file, the tool finds phones (tkclass=1) & device profiles (tkclass=254) where the built-in bridge isn’t default or privacy isn’t default, automatic call recording isn't enabled, the recording profile doesn't match, recording media source isn't phone preferred, or isn't associated to specified application user. It can optionally output the results to another CSV file.

The input CSV file should contain the list of DNs in a single column, with no header, like the below:

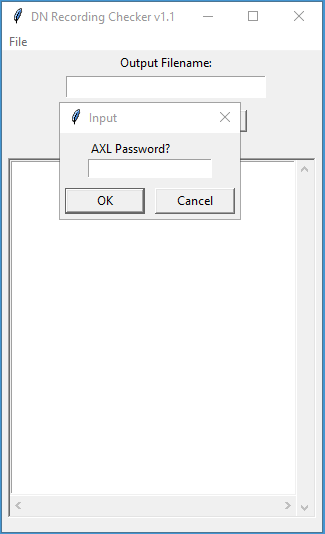


It connects to CUCM via the AXL API, so the AXL schema for the version of CUCM in use is required, this is downloaded from CUCM via **Application > Plugins > Cisco AXL Toolkit**. The required files contained within the .zip file are *AXLAPI.wsdl, AXLEnums.xsd* and *AXLSoap.xsd*.

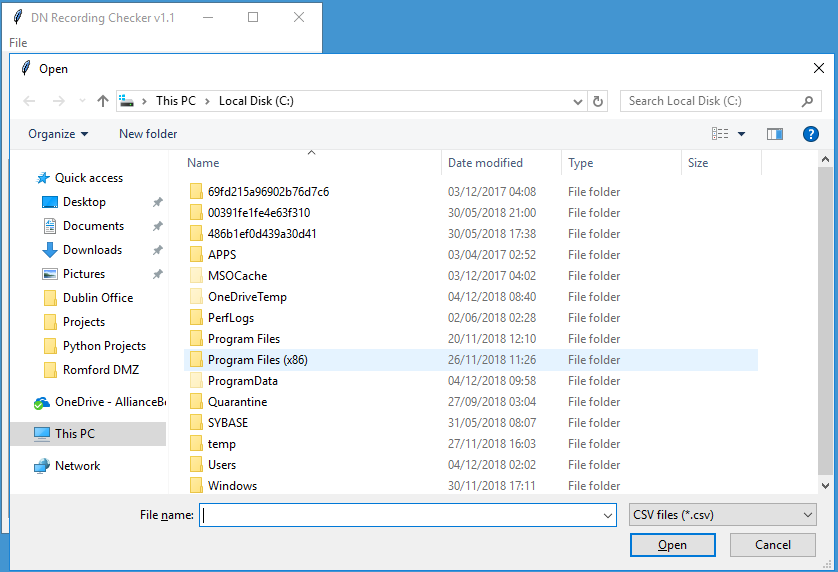
Different CUCM servers are defined in JSON formatted files, allowing for multiple CUCM clusters running different versions (and thus different AXL schemas). Load the CSV file via **File > Load AXL**:



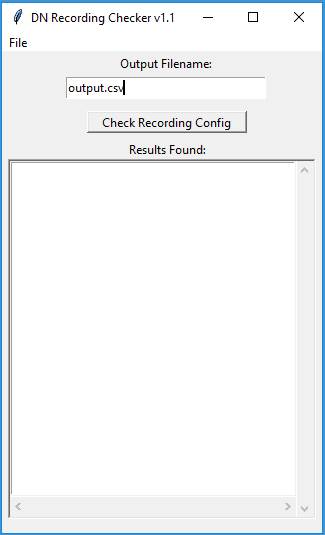
It will then prompt for the password:



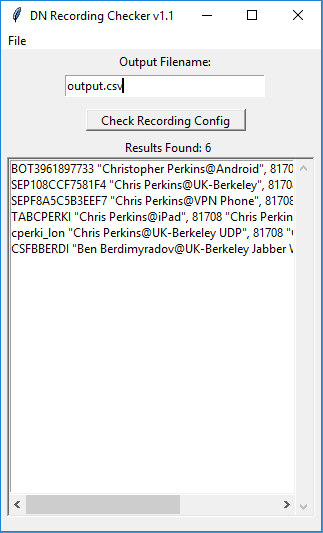
It will then prompt for the input CSV file:



If you wish to save the output in a CSV file, enter the filename into the text box:



Click **Check Recording Config**, the results will be displayed & optionally saved.



# Customising the Tool

The configuration for connecting via AXL to a CUCM cluster & what recording profile(s) to check against are stored in JSON format, for example:

[

{

"fqdn": "cucm-emea-pub.somewhere.com",

"username": "AppAdmin",

"wsdl\_file": "file://C://temp//AXLAPI.wsdl",

"recording\_profiles": ["NICE\_NTR\_RP"],

"application\_user": "NICE\_NTR\_USER"

}

]

* The JSON file starts with [ and ends with ].
* “fqdn” should be the FQDN or IP address of the target CUCM publisher.
* “username” is an application or end user with the Standard AXL API Access role.
* “wsdl\_file” points to the location of the AXL schema, note the slightly different path syntax for Windows.
* “recording\_profiles” is a list of the names of recording profiles.
* “application\_user” is the name of the application user recorded phones should be associated to, so that call information can be retrieved via JTAPI.

It is possible to simultaneously check against multiple recording profiles by adding more names to the list:

"recording\_profiles": ["NICE\_NTR\_RP", "RED\_BOX\_RP"]

If you've adjusted the CallManager service parameters so that built-in bridge is on by default & privacy is off by default, update the if statement checking the results of the SQL query as follows:

# Check for missing recording configuration, phones (tkclass=1) + device profiles (tkclass=254)

if d\_tkclass == "1":

if (d\_tkstatus\_builtinbridge == "0" or dpd\_tkstatus\_callinfoprivate == "1"

or dnmap\_fkrecordingprofile not in rp\_pkids or dnmap\_fkrecordingprofile == ""

or dnmap\_tkpreferredmediasource != "2" or rd\_tkrecordingflag != "1" or not user\_associated):

row["nice"] = dn["nice"]

self.list\_box.insert(tk.END, f'{d\_name} "{d\_description}", {n\_dnorpattern} "{n\_description}", {user\_associated}, {dn["nice"]}')

result\_list.append([d\_name, d\_description, n\_dnorpattern, n\_description, user\_associated, dn["nice"]])

cntr += 1

elif d\_tkclass == "254":

if (dpd\_tkstatus\_callinfoprivate == "1" or dnmap\_fkrecordingprofile not in rp\_pkids

or dnmap\_fkrecordingprofile == "" or dnmap\_tkpreferredmediasource != "2"

or rd\_tkrecordingflag != "1" or not user\_associated):

row["nice"] = dn["nice"]

self.list\_box.insert(tk.END, f'{d\_name} "{d\_description}", {n\_dnorpattern} "{n\_description}", {user\_associated}, {dn["nice"]}')

result\_list.append([d\_name, d\_description, n\_dnorpattern, n\_description, user\_associated, dn["nice"]])

cntr += 1