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 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 - 2019, Chris Perkins

v1.0 – initial release.

v1.1 – fixes some edge cases.

v1.2 – code tidying.

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

# Using the Tool

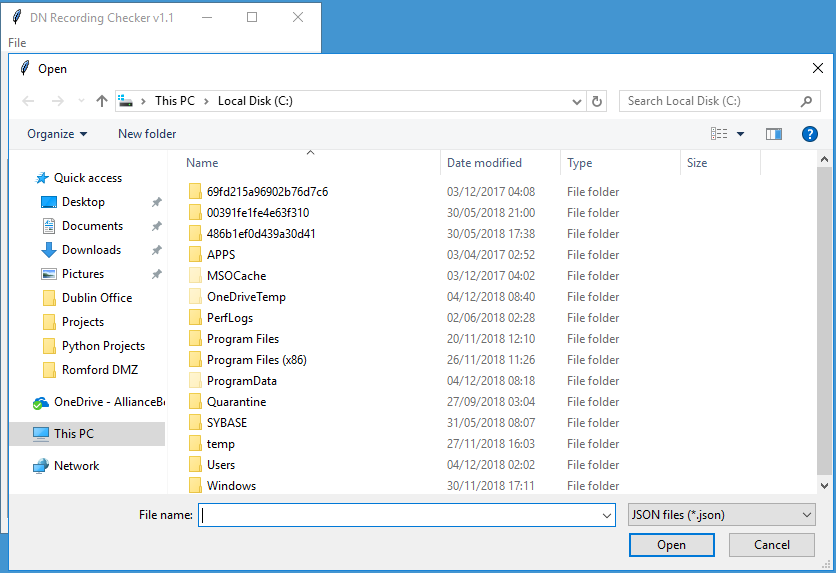
For a list of DNs in a CSV file, the tool finds phones (tkclass=1) & device profiles (tkclass=254) where the built-in bridge isn’t default or privacy isn’t default, automatic call recording isn't enabled, the recording profile doesn't match & recording media source isn't phone preferred. It can optionally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

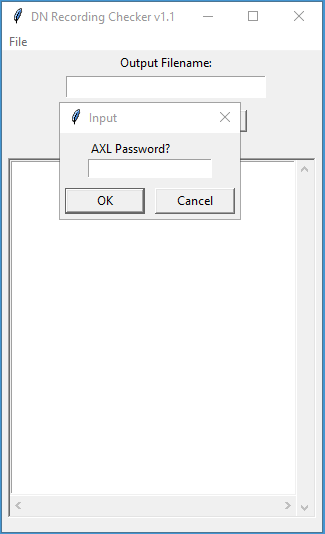


It connects to CUCM via the AXL API, so the AXL schema for the version of CUCM in use is required, this is downloaded from CUCM via **Application > Plugins > Cisco AXL Toolkit**. The required files contained within the .zip file are *AXLAPI.wsdl, AXLEnums.xsd* and *AXLSoap.xsd*.

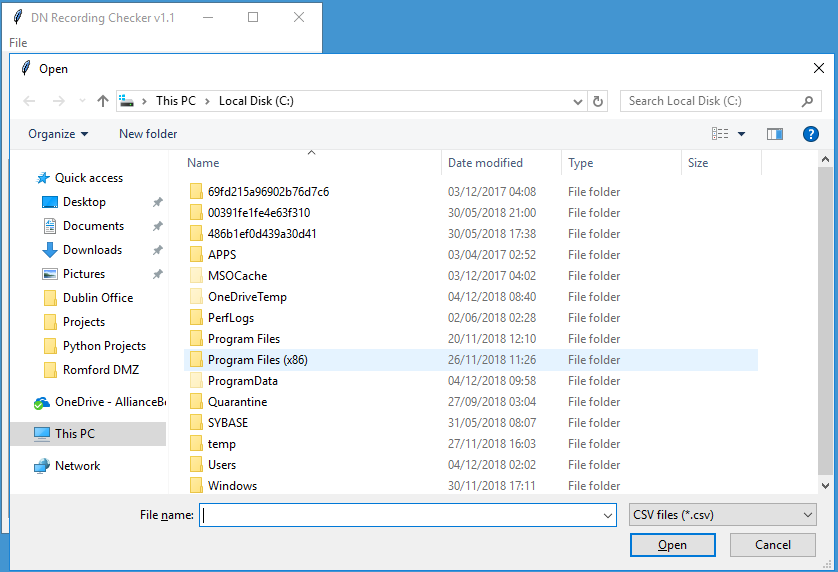
Different CUCM servers are defined in JSON formatted files, allowing for multiple CUCM clusters running different versions (and thus different AXL schemas). Load the CSV file via **File > Load AXL**:



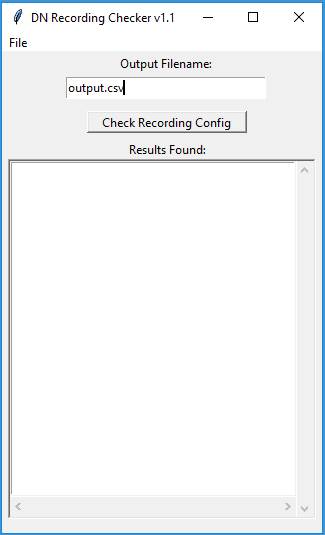
It will then prompt for the password:



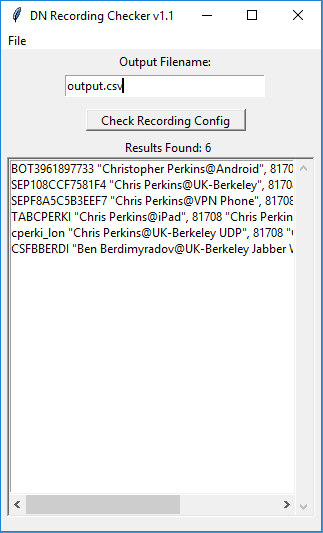
It will then prompt for the input CSV file:



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",

"subquery": "(dnmap.fkrecordingprofile!=(SELECT rp.pkid FROM recordingprofile rp WHERE rp.name LIKE 'NICE\_NTR\_ABITL\_RP') AND dnmap.fkrecordingprofile!=(SELECT rp.pkid FROM recordingprofile rp WHERE rp.name LIKE 'NICE\_NTR\_RP'))"

}

]

* 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.
* “subquery” is an SQL query that specifies the name of the recording profile, simply paste it into the quotes after LIKE.

It is possible to simultaneously check against multiple recording profiles by joining 2 queries via AND:

"subquery": "(dnmap.fkrecordingprofile!=(SELECT rp.pkid FROM recordingprofile rp WHERE rp.name LIKE 'NICE\_NTR\_RP') AND dnmap.fkrecordingprofile!=(SELECT rp.pkid FROM recordingprofile rp WHERE rp.name LIKE '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, change the SQL queries as follows:

# Check for phones (tkclass=1)

sql\_statement = f"SELECT d.name, d.description, n.dnorpattern, n.description AS ndescription " \

f"FROM device d INNER JOIN devicenumplanmap dnmap ON dnmap.fkdevice=d.pkid INNER JOIN numplan n " \

f"ON dnmap.fknumplan=n.pkid INNER JOIN deviceprivacydynamic dpd ON dpd.fkdevice=d.pkid " \

f"INNER JOIN recordingdynamic rd ON rd.fkdevicenumplanmap=dnmap.pkid WHERE (d.tkclass=1 " \

f"AND n.dnorpattern='{dn}') AND (d.tkstatus\_builtinbridge=0 OR dpd.tkstatus\_callinfoprivate=1 " \

f"OR {axl\_json['subquery']} OR dnmap.fkrecordingprofile IS NULL OR dnmap.tkpreferredmediasource!=2 " \

f"OR rd.tkrecordingflag!=1) ORDER BY d.name"

# Check for device profiles (tkclass=254)

sql\_statement = f"SELECT d.name, d.description, n.dnorpattern, n.description AS ndescription " \

f"FROM device d INNER JOIN devicenumplanmap dnmap ON dnmap.fkdevice=d.pkid INNER JOIN numplan n " \

f"ON dnmap.fknumplan=n.pkid INNER JOIN deviceprivacydynamic dpd ON dpd.fkdevice=d.pkid " \

f"INNER JOIN recordingdynamic rd ON rd.fkdevicenumplanmap=dnmap.pkid WHERE (d.tkclass=254 " \

f"AND n.dnorpattern='{dn}') AND (d.tkstatus\_builtinbridge=0 OR dpd.tkstatus\_callinfoprivate=1 " \

f"OR {axl\_json['subquery']} OR dnmap.fkrecordingprofile IS NULL OR dnmap.tkpreferredmediasource!=2 " \

f"OR rd.tkrecordingflag!=1) ORDER BY d.name"