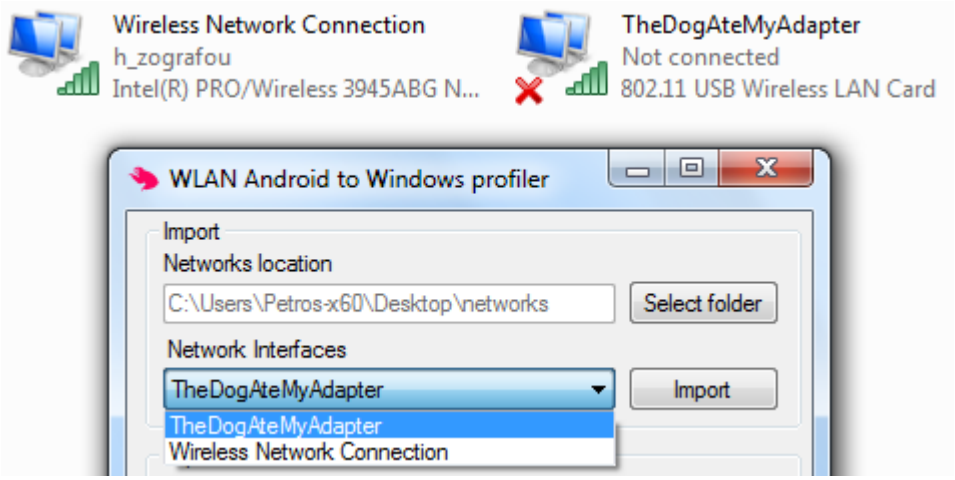
WLAN Android to Windows profiler documentation

Main functions:

- Import windows profiles
- Export (backup) windows profiles
- Convert profiles (from android)
- Dynamic Wireless interface selector
- Error resistant

In all the above functions you have to select the desire wireless network interface to complete the action. The list with the available interfaces is dynamic and identifies whenever an interface is wireless even if you rename it. The profiler is designed to be error resistant and provides a debugging log.



By default the program is looking for the folder “networks”, if the folder exists the folder is selected as the working folder. All the functions require a working folder in order to complete the import/export/convert.

Common faults:

Import faults:
During testing I discover that windows may fail to import a duplicate profile, so make sure you have in the working folder the networks you want to import.

Export faults:
If one of the networks contains special characters on the SSID like for an example “This is my | house“ the special character “|” is causing the export command to fail.

Convert faults:
Currently the profiler is not supporting older protocols like 802.1x so if you have a saved

network which uses 802.1x the profiler will ignore it and will move to next one. The supported networks are WPA – WPA2.

Import:

The import as mentioned above requires two things a working folder and a wireless interface. After the folder is selected the profiler will scan all the files and only recognize the .xml files for importing. A list with the full path of the files will be then created and an import function will execute in a hidden state a cmd instance that imports the profiles
<netsh wlan add profile filename = " _filename" Interface = " _interface" user = " + _user >

Export:

The export will scan the computer for the already installed profiles before continues the export, we could just use the command
<netsh wlan export profile interface="Wireless Network Connection">
that is exporting all the profiles in the given adapter but instead an alternative method that the profiler uses is,
<netsh wlan show profiles interface= _interface > (show all the profiles in the given adapter) and using regular expresions saves the SSID of the installed profiles, and then executes
</c netsh wlan export profile _profilename folder =_path key=clear interface=_interface>
The reason I choose to go with the second approach is if a network contains a special character the command will stop and any profile below it's not going to be exported, also this gives the profiler the error resistance and point to the next profile.

Convert:

For the converting a class AndNetworks is been used. The class contains only one method export(). The purpose of the class is to save the network information from the wpa_supplicant.conf after the parsing from the main class. The export() method is using the XML LinQ library from .NET framework and exports a windows compatible xml that the netsh command will recognize and import. The generated xml is completely built from scratch based on exported windows profiles. Before the conversion we use a list to strip and save the filename from any special characters, that list will become handy during importing as we will know the filenames that have been converted and saved in the 'networks' folder.

Not currently working

In the export we can specify if the xml will store the passphrashe (password) as a plain text or encrypted. The encryption feature is not yet be implemented.