AR8x application note

This guide providels details on how to remotely debug your Python -projects on the AR8x debian based OS using Microsoft Visual Studio.

1. PREREQUISITIES

Microsoft Visual Studio. In this documentation we use V2015 Professional, but other versions including Web Express and Desktop Express will probably work as well.

Get also Python installed to your PC in case you don’t have it yet installed. You can download the version you prefer from here: <https://www.python.org/downloads/> The version installed in to the AR8x is 2.7.9, so prefer to install that your PC as well.

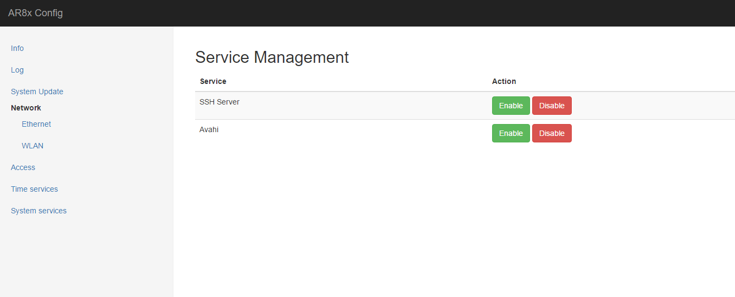
In addition to the above, you will also need install Python Tools. You can acquire them from <https://microsoft.github.io/PTVS/>

Once you have installed the Python Tools, you can start your Visual Studio.

The AR8x has packages for python-pip and PTVS debugger already installed.

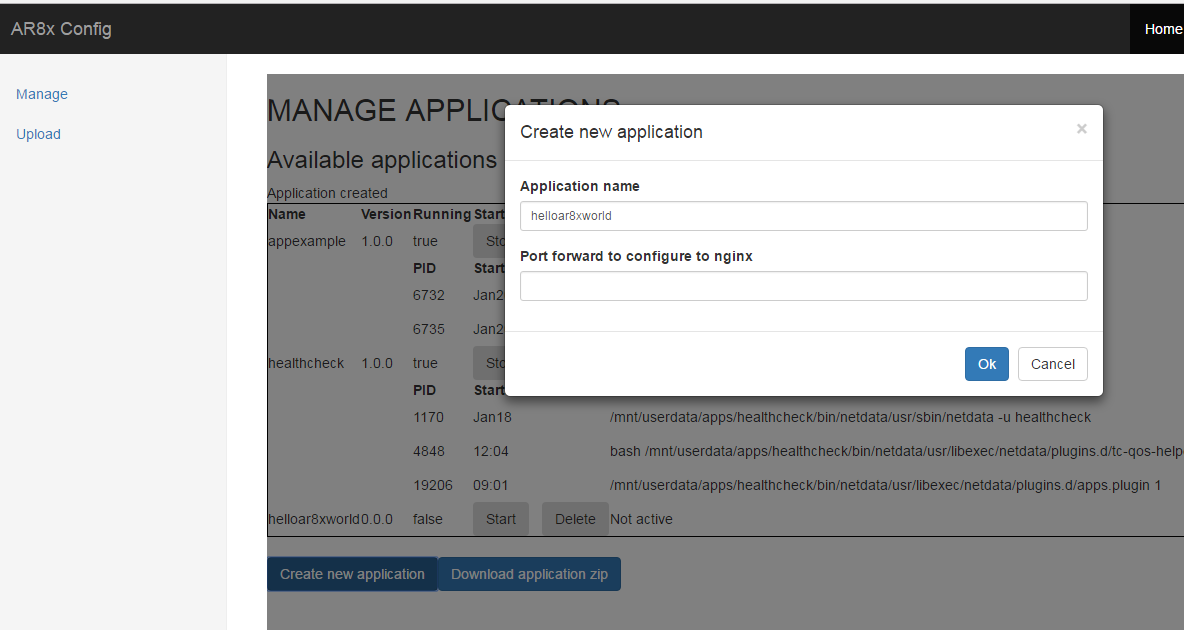
1. AR8x CONFIG
   1. SSH

SSH enabled on your AR8x device. If you have successfully powered your device up and connected to it as shown on **AN001\_AR8x\_GettingStarted.docx**, open up your web browser and the Web Config UI of your AR8x device. On the WebConfig, open “System” and click “System Services” from the menu on the left. From this menu you need to enable SSH.

****

* 1. BLANK PROJECT

After the above, create “blank” project from Applications->Manage->Add blank project. Once you have given the project a name, it will automatically create an username and a password for the application. The password & username combination is applicationname:applicationname. *\*Note that the platform does not allow SSH for any users by default, so these login details are meant only for debugging purposes. The SSH will be disabled once the device is restarted again.*

**

1. VS Python project

On Visual Studio, click “File-New-Project..” and select “Python-Python Application”. Then add a name for your project, we use “helloar8xpython” in this documentation. After the project has been created, right click on the name of the project in the Solution explorer and add empty python file to with the name “helloar8xpython.py” to it.

Add these lines to your python file:

#import the ptvsd library

import ptvsd

print ('helloar8xpython starting')

#Enable remote debug

ptvsd.enable\_attach('secret',address = ('0.0.0.0',5678))

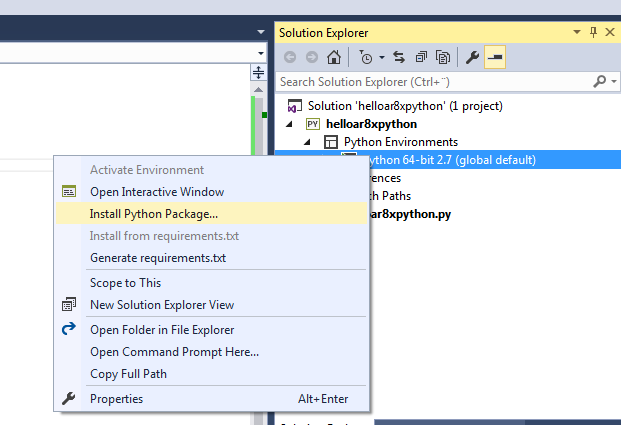
#Wait for VS to attach

ptvsd.wait\_for\_attach()

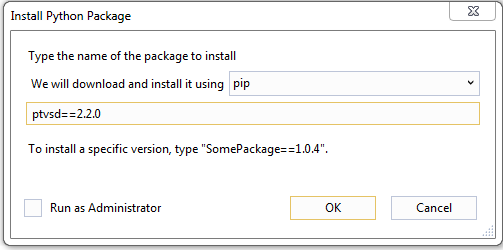
print ('hello ar8x world from python')

#tcp://secret@ar8x81d39b.local:5678

On Solution Explorer, expand “Python environments” and right-click “Python 64-bit 2.7 (global default)”(or similar) and select “Install Python package...”.



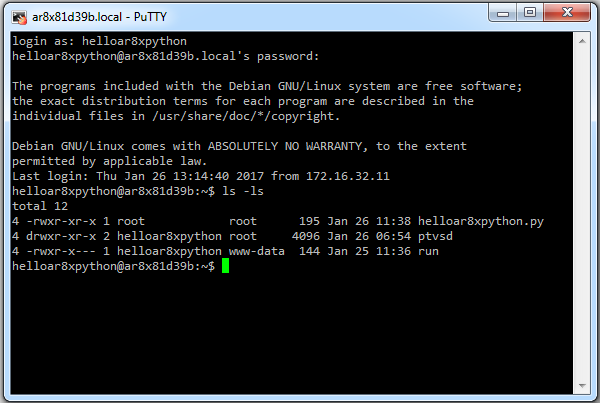
On the window that opens, type in “ptvsd==2.2.0” and select “Run as Administrator” and click OK.



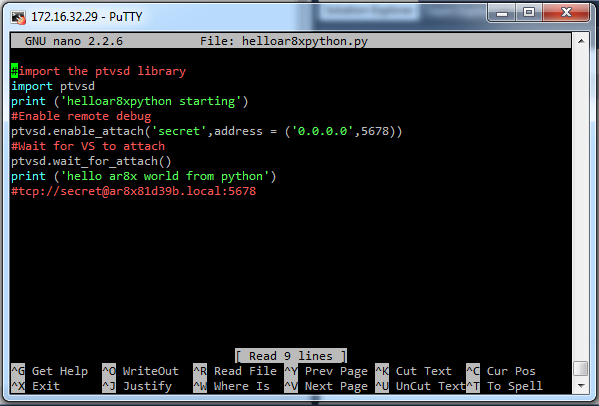
Now that the ptvsd-package is installed, you should be able to debug the project locally. Try it out, but note that since no debug-connections, the code inputted will not do much. If the project starts as expected, you can continue.

At this point, open also a SSH connection to your AR8x device via Putty or other SSH client. You should be able to connect to your device using the ar8x123456.local-address described in AN001. For the username and password, use the name of the blank project you created on chapter 2.2. Open also a SFTP-connection using the same username & password combination. Find the installation folder your Python Tools, should be something like: “**C:\Program Files (x86)\VS2015\Common7\IDE\Extensions\Microsoft\Python Tools for Visual Studio\2.2\ptvsd**”. Copy the ptvsd folder to the home-folder of the user you used for logging in to the device.

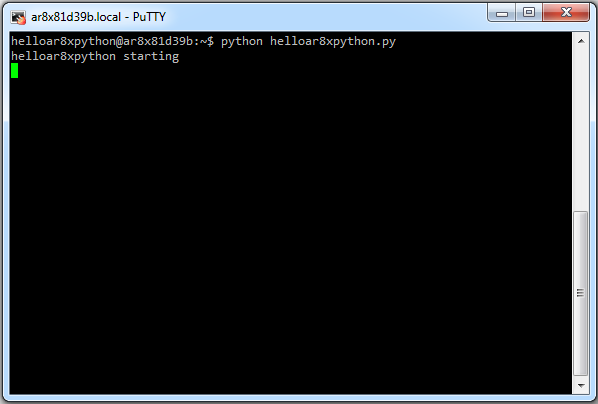
After you have copied the folder in question to the device, open your SSH client and create a file to the home-folder with the same name as your Visual Studio project contains, i.e. helloar8xpython.py. At this point the folder should contain something like this:



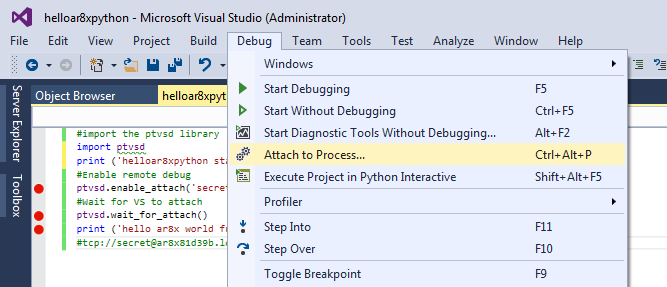
Now edit the helloar8xpython.py to contain the same data as your Visual Studio file has, i.e:



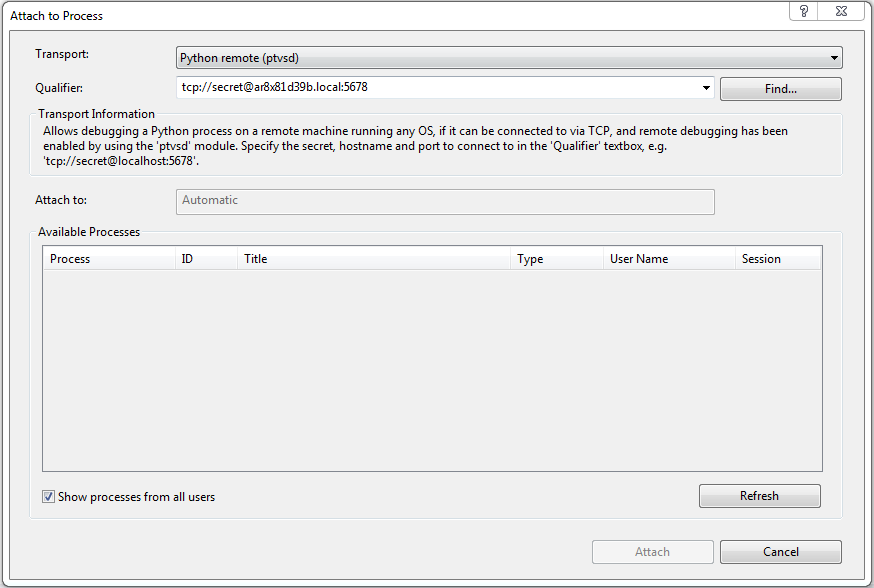
Save the file and start the script by typing in: python helloar8xpython.py:



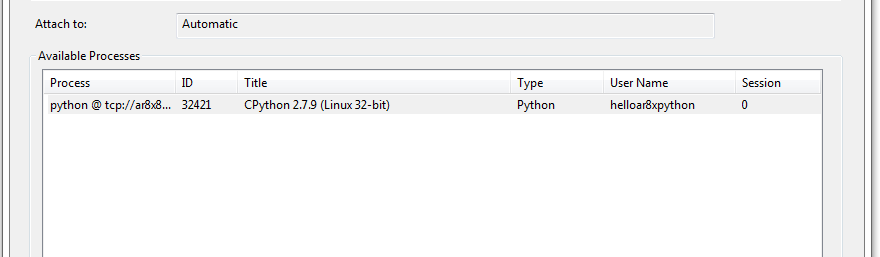
Now move back to Visual Studio and click “Debug-Attach to Process...”



Select “Python remote (ptvsd)” as the Transport and type in to the Qualifier the secret@address.local:port-combination as follows: *tcp://secret@ar8x81d39b.local:5678.* Replace the address with your AR8x device address and port if you used something else than defined in this example. ***secret***should be replaced with the string used when calling ptvsd.enable\_attach in helloar8xpython.py-file.



Then click “Refresh” from the bottom and you should see the python-process appearing there.



Click Attach and you the app should execute. If you have placed breakpoints to the code, you should also see it breaking there.

