# City Scape Test Suite – User Manual

## Overview

City Scape Test Suite application enables an end-to-end offline validation of spectrum station setup based on the CityScape Spectrum Observatory software. The test suite allows users to configure RF Sensor parameters offline and thereby captures the desired spectrum data locally without having to connect to the Azure backhaul. Also, it allows users to bypass a few restrictions on the sensor parameters for more flexible testing. To plot the PSD estimate data, users may use the Python script provided (CityScapePSDFilePlotter, described in Section 1.4).

## 

Fig. 1: Interface of the CityScape Test Suite.

### System Design

The test suite is built upon the observatory station service, and shares many common components.

The host PC will use a USRP to collect I-Q data and generate PSD estimates, using exactly the same software routine used when the station is operating in normal operational mode. The host PC can be accessed using TeamViewer. If you do not have a remote access to the host PC (due to the lack of the network connectivity), you can either disassemble the station and attach a monitor and a keyboard to locally control the station host PC or temporarily replace the host PC with your own computer (which has the station software, UHD, and the test suite installed).

Assuming that everything works fine, following is how the data flow within the system (\*re-tune and stitching process omitted):

1. The USRP receives I-Q data, and passes that to the host PC.
2. The host PC passes the I-Q data to UHD (‘USRP driver’). UHD calibrates the data (DC Spike and I-Q imbalance), and passes the data to the station software.
3. The station software re-calibrates the data, so as the periodogram of the data is in dBm scale. From there, the station software calculates the PSD estimates.
4. The station software writes the PSD estimate data into the local disk.
5. Using “CityScapePSDFilePlotter”, the user plots out the PSD estimates stored in the data files.

The GUI of the test suite lets the tester to configure necessary parameters for the step #3 and step #4, including the start / stop scan frequency, USRP address, etc. CityScapePSDFilePlotter, which is a separate tool but bundled together with this test suite, lets the tester to plot PSD estimates from the generated files. These tools do not, however, let the tester to manipulate the calibration files: **the data calibration routines are automatically applied by UHD and the station software, and is not controlled by the test suite. The tester must manually verify if the calibration files are correctly generated and installed in the correct paths before running the program (exact steps described in Section 1.2).**

Following are the diagrams of the system, in device-level (Fig.2) and data flow level (Fig.3), as discussed above:

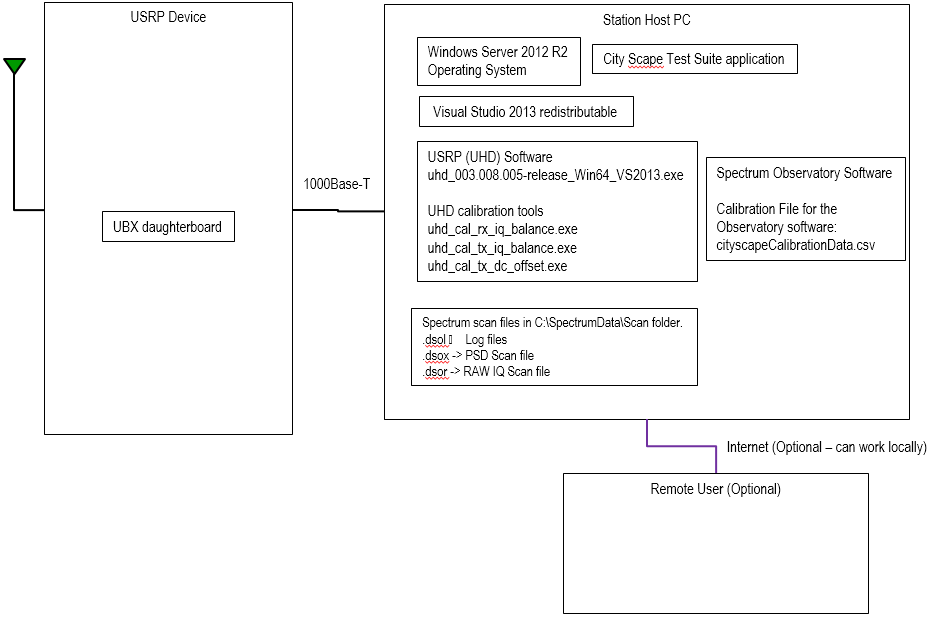


Fig. 2: System Diagram, in Device Level.

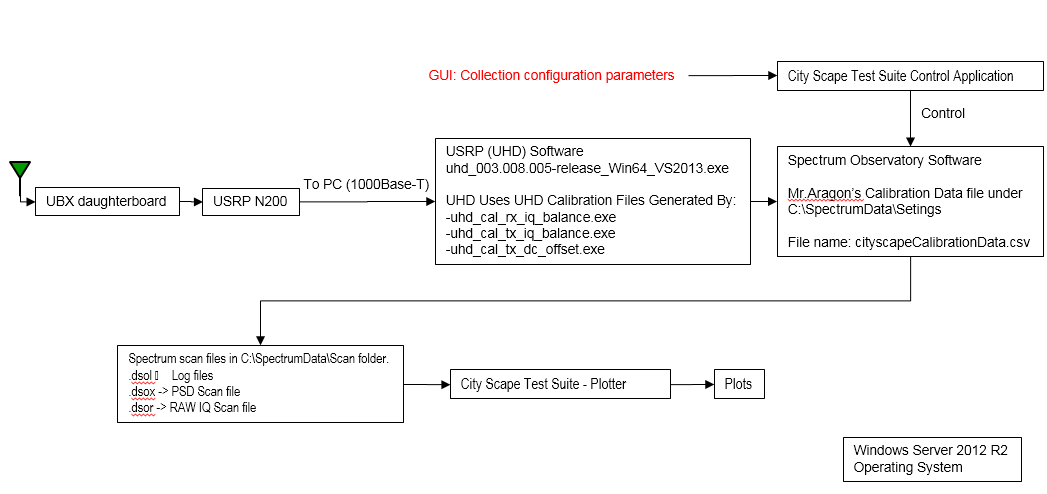


Fig. 3: System Diagram, in Data Flow Level.

## Getting Started

To start using City Scape Test Suite application, please follow the below instructions on the test PC:

### Pre-Requisites

1. Visual Studio 2013 redistributable is required to run the Spectrum Observatory software on the test PC. Download the software from here: <http://www.microsoft.com/en-us/download/details.aspx?id=40784>
2. Download the City Scape Spectrum Software from here: <https://www.dropbox.com/s/41hu3etnxcnr5yg/ImporterSetup.msi?dl=0>
3. Open the ImporterSetup.MSI and follow the onscreen instructions to complete the installation.
4. Download City Scape Test Suite Application from here: <https://www.dropbox.com/s/fhfi304ea1c2gpn/CiytScapeTestSuite.zip?dl=0>
5. Unzip the City Scape Test Suite Application to a known folder on the local drive.
6. Install Ettus UHD software on the test PC from here:

<http://files.ettus.com/binaries/uhd_stable/uhd_003.008.005-release/uhd_003.008.005-release_Win64_VS2013.exe>

1. Make sure the USRP device is configured and calibrated in UHD-level correctly as suggested in section 3 of City Scape Station Deployment Guide (<https://csmaster20161201t0759.blob.core.windows.net/msidownload/CityScape%20Station%20Deployment%20Guide.pdf>).
   * It is recommended to setup the device with IP address 192.168.10.2
2. Plug the test PC to the USRP device using an Ethernet cable and verify the connection using ping command. For eg: ping <USRP device IP address>
3. Run USRP commands as below: <install-path> normally C:\Program Files (x86)\UHD

<install-path>\bin\uhd\_find\_devices.exe

<install-path>\bin\uhd\_usrp\_probe.exe

* + If USRP commands are successfully completed, the test PC is set up correctly and ready for tests.

1. **Observatory Software-level Calibration**: As mentioned in the Dropbox share document, Calibration-Cityscape-USRP-20161009.pdf [ <https://www.dropbox.com/s/1m0j96h2x7cuaw2/Calibration-Cityscape-USRP-20161009.pdf?dl=0> ], place the calibration data file under folder C:\SpectrumData\Settings with file name ***cityscapeCalibrationData.csv***. **Please note**, scanner service looks for calibration data with the same exact name. Do not change the file structure.

### Launching the City Scape Test Suite Application

1. Browse to the City Scape Test Suite folder location and locate CityScapeTestSuite.exe application.
2. Open the application to see the user interface.
3. To load the values, click on the ‘Load From File…” button and choose stationConfiguration.dsos file from City Scape Test Suite folder.

### Save Scanner Configuration Changes (dsos)

1. Launch the City Scape Test Suite Application (by following the section 1.2.2).
2. Edit necessary fields. For eg: Edit ‘Start Frequency MHz’ and ‘Stop Frequency MHz’ values in Scanner Configuration.
   * If you want to generate PSD estimate files, instead of RAW I-Q data files, check ‘Output PSD Data’ in ‘Experiment Type 2 – Time Averaged PSD’. Note that CityScapePSDFilePlotter.py can only handle PSD estimate files (.dsox), so you have to use a different parser software (example: <https://github.com/city-scape/CityScape_Raw_Data_Decoder>) if you want to process the Raw I-Q data files.
3. Click on ‘Save As…” button
4. Browse to C:\SpectrumData\Settings folder and save with file name ‘stationConfiguration.dsos’. Please note, file name of scanner configuration file to be exactly as given. If there is already file present with same name, it can be overwritten or rename the existing file to a different one.

## Begin Tests

1. Stop the Scanner Service. To stop the scanner service, use either task manager to kill the service or use the command below from an elevated cmd window:

taskkill /F /IM Microsoft.Spectrum.Scanning.Service.exe

1. Launch City Scape Test Suite Application, as in 1.2.2.
2. Make necessary changes to the settings.
3. Save Scanner Configuration Changes to dsos file, as in 1.2.3.
4. Start the Scanner Service. Start it from task manager, as shown in the below screenshot.
5. Spectrum scan files will be captured in C:\SpectrumData\Scan folder. Below are the files with extensions:

.dsol 🡪 Log files

.dsox -> PSD Scan file

.dsor -> RAW IQ Scan file

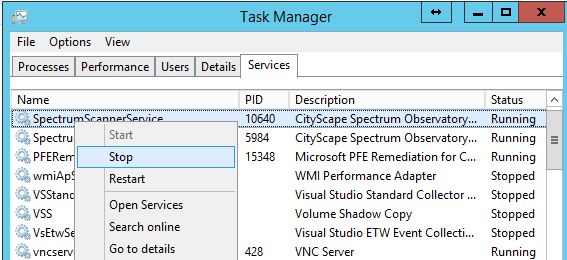
****

Fig. 4: Starting / Stopping Station Services

At any given point, scanner service can be stopped to start another scan with new configuration settings. The tests can be repeated as many as needed. It is highly recommended to stop and start the scanner in between the configuration changes. Whenever the Scanner Service is stopped, a new scan file is generated under C:\SpectrumData\Scan folder.

## Plotting of Data

CityScapePSDFilePlotter is a simple Python script to help you plotting the PSD estimate files generated using City Scape Test Suite. Download links:

**Distribution for Windows (if you don't want to figure out Python library dependencies):**[**https://www.dropbox.com/s/26gr5uh0f8t3adh/Windows-CityScapePSDFilePlotter.zip?dl=0**](https://www.dropbox.com/s/26gr5uh0f8t3adh/Windows-CityScapePSDFilePlotter.zip?dl=0)

**Source codes (If you want to modify the program or to run it on non-WIndows machines):**[**https://github.com/city-scape/CityScape\_Raw\_Data\_Decoder/blob/master/python/GUI\_Example/CityScapePSDPlotter.py**](https://github.com/city-scape/CityScape_Raw_Data_Decoder/blob/master/python/GUI_Example/CityScapePSDPlotter.py)[**https://github.com/city-scape/CityScape\_Raw\_Data\_Decoder/blob/master/python/psdFile\_pb2.py**](https://github.com/city-scape/CityScape_Raw_Data_Decoder/blob/master/python/psdFile_pb2.py)

(You need to download both CityScapePSDPlotter.py and psdFile\_pb2.py and place them in a same directory.)

### Installation

* Distribution for Windows (if you don't want to figure out dependencies):

Download and extract the zip file. Run CityScapePSDPlotter.exe to execute the program.

* **Using Python Source codes:**

1. Install the following dependencies.

* Python 2.7
* Protobuf Python binding (python-protobuf)
* MatPlotLib
* NumPy
* SciPy

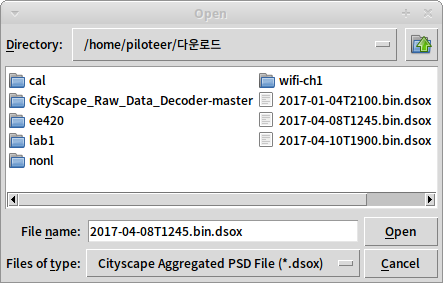
2. Place both CityScapePSDPlotter.py and psdFile\_pb2.py in a same directory.

3. Using CLI (bash, tcsh, cmd.exe, etc), go to the directory where the python files are located.

4. Run "python CityScapePSDPlotter.py" to execute the program. (Alternatively, you can use chmod +x CityScapePSDPlotter.py to make it an executable file if you are using Linux of Mac OS X.)

### Usage

1. When you launch the program, it will open a file dialogue. Navigate to where the aggregate PSD file is located (default: **C:\SpectrumData\Scan**), and open the file (fig.6).



​Fig. 6 : File Open Dialogue.

1. Give the program some time to process the file. It is somewhat slow on Windows, but if the file is small enough (5 - 10 min), it won't take too long (fig .7).

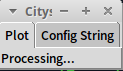
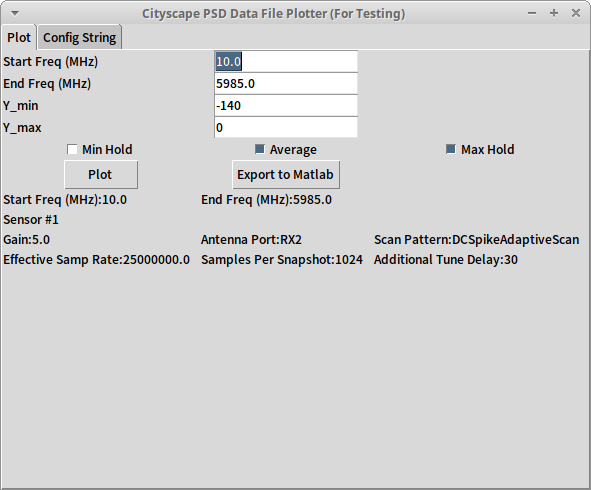
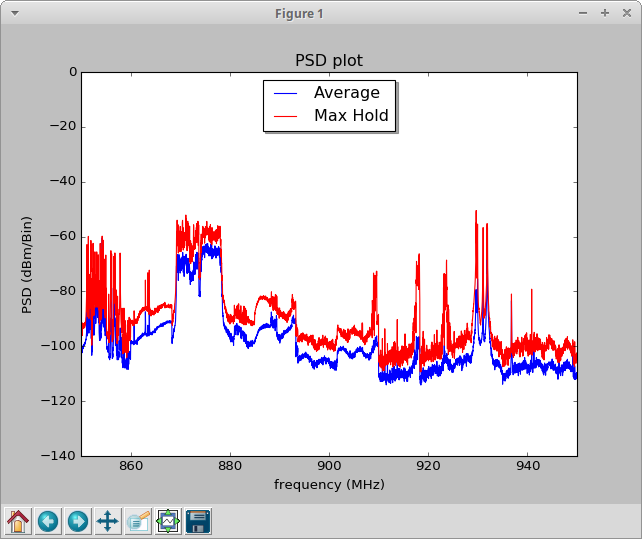


Fig.7: PSD Plotter application, processing the data.

1. Once the program is ready, it will display a summary of the station configuration and prompt you to input start & stop frequency, and y-axis range of the plot (default: -140 to 0 dBm/bin). Set the values, and click the "Plot" button below to generate a plot (Fig.8 / 9).  
     
     
   ​ Fig.8: PSD Plotter application, ready to plot.



​Fig. 9: A PSD Estimate plot, generated using the plotter.

## Limitations

Current limitations of the test suite include:

* PSD estimate files generated using the test suite will suffer from the same data quality issues that are currently suffered by the City Scape station software (including the aliasing problem).
* CityScapePSDPlotter.py only supports PSD estimate data files for now (can’t plot Raw I-Q data files, even though they can be generated using the test suite).

## Appndix A – Recommanded Test Approach

### I-Q Imbalance Calibration (UHD-level Calibration)

1. Open a file explorer, and go to ‘%windir%\system32\config\systemprofile\AppData\Roaming\.uhd\cal’. If you see \*.csv files, the calibration files are installed.
2. To verify that the calibration files that are installed in the directory are the correct files, open a command prompt or a PowerShell window and type “uhd\_usrp\_probe”. Locate the serial number of the daughterboard, and compare that against the serial number of the daughterboard in the calibration file. If they match, the installed files are the correct files.

Alternatively, you can try injecting a signal to the radio and observe its mirror image in the periodogram (due to the I-Q imbalance). However, as UBX-40 is a high-quality daughterboard, this is not always obvious.

### Amplitude Calibration (Station Software-level Calibration)

1. Open a file explorer, and go to ‘C:\SpectrumData\settings’. If you see cityscapeCalibrationData.csv, the calibration file is installed.
2. To verify that the calibration file that is installed in the directory is the correct file, open a command prompt or a PowerShell window and type “uhd\_usrp\_probe”. Locate the serial number of the daughterboard, and compare that against the serial number of the daughterboard stored in the calibration file. If they match, the installed files are the correct file.

**Alternative Way (using a signal generator):**

1. Using a calibrated signal generator, inject a CW (with known power level and frequency) to the sensor.
   * Maximum allowable level of the tone is -20 dBm. Anything above may permanently damage the board. Recommended level is about -50 dBm.
   * Use of an attenuator is recommended if the generator can generate more than -20dBm of power.
2. Using the test suite, generate PSD estimates and plot the results.
   * Make sure that the injected CW falls between the start and the stop scan frequency.
3. From the plot, locate the injected CW and check whether the power level of the injected CW matches your expectation.
4. Repeat Step 1-3 with different power level and frequency settings until you are confident that the station is calibrated as intended.

### On Site Testing

1. Temporarily install the sensor to the desired location.
2. Using the test suite, generate PSD estimates between 10 MHz to 3000 MHz with the Rx Gain knob set to 10 dB and 0 dB. Also, with Rx Gain knob set to 0 dB, attach a 10dB attenuator to the sensor and generate PSD estimates.
   * Label the generated PSD estimate files carefully, so as you wouldn’t end up mixing them accidently.
3. Plot the PSD estimates, and cross-compare the results.
   * If the observed power level in any FFT bin increases significantly more (or significantly less) than the increase in the configured gain level, and if the observed signal is not supposed to be bursty (example of non-bursty transmission: FM, ATSC, etc), nonlinearity is detected.
     + To avoid a false-positive case, you may want to use enough averages and FFT points to make the final conclusion.
4. If you are still not sure, try calculating the observed noise power level of the sensor and compare that against the expected noise power level. This can be done by calculating:

, where the noise figure (\*Gain, frequency dependent) of the board is available at <http://files.ettus.com/performance_data/ubx/UBX-without-UHD-corrections.pdf>

## Appendix B - PSD File Size Estimate

The sensor is tasked to scan for I-Q data and the time-averaged minimum, maximum and average PSD values for 1-minute durations are computed on the PC. Lossless compression of 10% is achievable for these files. The estimated file size is

File size = 3*nNltC*

where *n* is the number of snapshots, *N* is the samples per snapshot, *l* is the size of each PSD sample which is 16 bits, *t* is the time in minutes during which time-averaged FFT is computed and *C* is the compression factor. Since the minimum, maximum and average PSD values are stored, a factor of 3 is introduced.

For scanning and computing 1024-point PSD charts over 50 MHz to 6 GHz at 25 MS/s for 1-minute at 10% compression, each PSD file occupies **1.315 MB** of storage.