**User Guide for Jenkins QA Automation scripts**

July 7th 2017

**[A] Introduction:**

This script is to be used with the Jenkins server. It consists of three main scripts namely *QA\_Automation.py* (top level script and controls the whole automation), *Run\_PKG\_Installation.py* (Handles the package installation on your station) and last is *Run\_QA.py* script (Handles the QA part).

**[B] Software required:**

1. Python version 2.7

Important note: After you install Python 2.7 on your machine, you need to add the python path to environmental variable to make use of command prompt to run python scripts.

Go to **Computer->Properties->Advanced System Properties-> Environmental variables->System variables->Path->Edit**, and then add the path of your python directory all the way at the end of this path variable. Typically this path should be **C:\Python27** if you have installed the same version of python as mentioned above.

**[C] Structure and script details:**

QA Automation consists of the following scripts. The package will be downloaded to the station through the Jenkins server

1. ***QA\_Automation.py*:** This is top level script and calls the other two scripts.
2. Determines the package to be installed
3. Parse the QA\_config file. Details can be found in Section F.
4. Error handling: Handles error thrown while the script is running using try-catch block.
5. ***RUN\_PKG\_Installation.py:*** This script is called from the top level script. It handles following functions:
6. Extract package on hard disk (i.e. QA-station)
7. Installing the package in background.
8. ***Run\_QA.py:*** This script handles the QA specific to each chipset.
9. This script runs the QA based on the QA level (sanity/full)
10. Moves the contents of the Setup\_files folder to the
11. Save the log/csv files and create the report using reportbuildr.
12. Exit once the QA is finished to the top level scrip.

**[D] Setup Steps:**

i. Connect the QA Station to the Jenkins server.

## ii. Move the QA\_config.txt and Setup\_files folder into the job folder on the SVN. The Setup\_files folder should contain the needed additional files, such as the test flow files and IQramp template files. It will contain another folder called Setup that will contain the path loss file and setup file. Check the Verify\_IQfact+\_MRVL\_8801 folder as an example.

iii. Change the QA\_config.txt file to fit the station.

iv. Jenkins will automatically pull the relevant files from the SVN when running the QA.

**Before you run the scripts on any machine**, please make sure following points are done:

1. Add python path (C:\Python27) to environmental variable as mentioned in Section [B] above.
2. Change the security settings on your machine to minimum level. For this follow the path:

Go to **Control panel-> User Accounts-> Again user accounts** ( you should be here by now Control Panel\User Accounts\User Accounts) then -> **Change user account control settings ->** Change the slider level to **Never notify**.

For more details with pictures, follow this link : <http://www.sevenforums.com/tutorials/299-user-account-control-uac-change-notification-settings.html>

**[E] Running the Program:**

**The scripts will be automatically called by Jenkins** through the **Verify\_PROJECTNAME job**. This job will be automatically called after the build is finished, but can also be kicked off manually through Jenkins.

**[F] Config File format:**

1. Project name – The name of the project. Ex: IQfact+\_MRVL\_8801

2. Test\_mode – Determines which test list to use. Can be sanity or full.

3. New\_Pkg\_Installation – Determines if the package is old or new. Should be set as 1.

4. Install\_Redistributables – Determines if the package redistributables need to be installed.

5. Recipient\_Email\_ID – Where emails should be sent out to. Deprecated since Jenkins now sends emails.

6. Enable\_Emails – Enable email notifications. Set to 0 since Jenkins now sends emails.

7. Test list – The name of the test flow files to be used in testing.

8. Number\_of\_DUTs – How many DUTs that are to be tested simultaneously

9. Com\_Ports\_List – List of the comm ports corresponding to each DUT.