Wireless Innovation Forum Contribution

All blanks ([…]) must be completed for this Submission to be given consideration. In making this submission, the Submitters agree that they are bound by the Policies and Procedures of the Software Defined Radio Forum Inc. doing Business as the Wireless Innovation Forum ("The Forum"), including but not limited to the Intellectual Property Rights Policy (Policy 007) and the Restricted and Controlled Information Policy (009).

Committee: SSC WG4 CBSD Task Group
Title: WInnForum SAS Test Harness for CBSD UUT Tutorial
Short Title: WInnForum SAS Test Harness for CBSD UUT Tutorial
Source:

Idan Raz
Airspan

Chris Williams-
Ericsson

Douglas Goedken
Nokia

Date: [2 Mar 2018]
Distribution: [Members]

Document Summary: Tutorial for the WInnForum SAS Test Harness for CBSD UUT

Notes of Importance: [Optional. Short statement; please limit to 50 words or less.]

Impacts/Effects: [Optional. Short statement; please limit to 50 words or less.]

Action Desired: [Optional]
Action Required for Closure: [Optional]
Desired Disposition Date: [Day Month Year]

IF THIS IS A CODE CONTRIBUTION, THE SECOND PAGE MUST BE COMPLETED AND SIGNED

1. Additional Copyright License

In addition to the rights and licenses granted by the undersigned pursuant to Section 4 of the IPR Policy, the undersigned hereby agrees as follows: if this Code Contribution is included in whole or in part in the Specification or Other Work Product of the Committee named above, the undersigned hereby grants the Forum and its members a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the undersigned's copyrights in its Code Contribution right to sublicense the right to implementers or users, as appropriate, of such Code Contribution to copy, modify, and redistribute such Code Contribution or included portion thereof. THIS CODE CONTRIBUTION IS PROVIDED TO THE FORUM BY THE UNDERSIGNED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE DISCLAIMED. IN NO EVENT SHALL THE UNDERSIGNED BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS CODE CONTRIBUTION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

2. Future Code Contributions (initial if appropriate)

___ The rights and licenses granted above apply to this and all future Code Contributions made by the undersigned relating to this Specification or Other Work Product.

IN WITNESS WHEREOF, the Code Contributor has executed this Contribution Agreement through its duly authorized Representative.

Member: _____

By:_____

Name:_____

Title:_____

Date:_____

WInnForum SAS Test Harness for CBSD UUT Tutorial

Version V1.0.0.1

2 March 2018

TERMS, CONDITIONS & NOTICES

This document has been prepared by the SSC Work Group 4 to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter "the Forum"). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SSC Work Group 4.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter's copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum's participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

Table of Contents

List of Figures

List of Tables

No table of figures entries found.

Contributors

The following individuals made significant contributions to this document:

Idan Raz (Airspan)

Chris Williams (Ericsson)

Douglas Goedken (Nokia)

WInnForum SAS Test Harness for CBSD UUT Tutorial

1 Introduction

WInnForum SAS Test Harness is developed for test and certification purposes of CBSD UUT. The WInnForum SAS Test Harness is available for download from the GitHub repository https://github.com/Wireless-Innovation-Forum/Citizens-Broadband-Radio-Service-Device

Note: It is recommended to check the GitHub repository as the Test Harness code may have periodic updates to address reported items.

2 Scope

This document is the tutorial of WInnForum SAS Test Harness.

3 References

3.1 Normative references

The following referenced documents are necessary for the application of the present document.

[n.1] WINNF-TS-0065 Version V1.1.0, “CBRS Communications Security Technical Specification”, 26 July 2017

[n.2] WINNF-TS-0022 Version V1.1.2, “WInnforum CBRS Certificate Policy Specification”, 6 February 2018

[n.3] WINNF-TS-0016 Version V1.2.1, “Spectrum Access System (SAS) - Citizens Broadband Radio Service Device (CBSD) Interface Technical Specification”, 3 January 1018

[n.4] WINNF-17-SSC-0002 Version V2.0.2, “Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): WInnForum recognized CBRS Air Interfaces and Measurements”, 6 September 2017

[n.5] WINNF-TS-0122 Version V1.0.0, “Conformance and Performance Test Technical Specification; CBSD/DP as Unit Under Test (UUT)”, 19 December 2017

4 Definitions and abbreviations

4.1 Abbreviations

CBSD	Citizens Broadband Radio Service Device
CBRS	Citizens Broadband Radio Service.
CFR	Code of Federal Regulation.

CPI	Certified Professional Installer.
DP	Domain Proxy
EIRP	Effective Isotropic Radiated Power
GAA	General Authorized Access.
HTTP	Hypertext Transfer Protocol.
HTTPS	HTTP over TLS.
JSON	JavaScript Object Notation.
PAL	Priority Access License
PPA	PAL Protection Area
SAS	Spectrum Access System
TLS	Transport Layer Security.

4.2 Definitions

Blacklist: A list of *CBSDs* that are to be denied service.

CBRS band: The 3550-3700 MHz Citizens Broadband Radio Service band.

CBSD Antenna: The radiating element(s) of the CBSD. Each CBSD has one CBSD Antenna. Note that the CBSD's antenna may be instantiated with multiple physical antennas (e.g., an antenna array for MIMO operation), but those antennas must be transmitting one aggregate waveform collectively from a single geolocation, and with a total transmit power that conforms to all the CBSD's registration parameters and authorized transmit power levels provided by the SAS in its active Grants (e.g., maximum allowable EIRP).

CBSD Registration: The procedure by which a CBSD indicates to a SAS its intention to operate. Successful registration implies a validation by the SAS that the CBSD has been FCC certified and confers on the CBSD the right to be authorized by the SAS to operate in accordance with a Grant. During the registration process, each CBSD provides a fixed location, unique identifiers (e.g., owner information, device information), *Group* membership, and radio-related capabilities. A successful registration procedure concludes with the SAS providing a unique identifier for that CBSD.

CBSD User: The registered entity that has operational responsibility for the CBSD.

Channel: the contiguous frequency range between lower and upper frequency limits.

Citizens Broadband Radio Service Device (CBSD): Fixed Stations, or networks of such stations, that operate on a Priority Access or General Authorized Access basis in the Citizens Broadband Radio Service consistent with Title 47 CFR Part 96. For CBSDs which comprise multiple nodes or networks of nodes, CBSD requirements apply to each node even if network management and communication with the SAS is accomplished via a single network interface.

Domain Proxy (DP): An entity engaging in communications with the SAS on behalf of multiple individual CBSDs or networks of CBSDs. The Domain Proxy can also provide a translational

capability to interface legacy radio equipment in the 3650-3700 MHz band with a SAS to ensure compliance with Part 96 rules.

Grant: The authorization provided by a SAS to a CBSD, subject to a Heartbeat exchange, to transmit using specified operating parameters. Grants are identified by a unique Grant identifier. Once issued, a Grant's operating parameters are never changed; if new or modified operating parameters are required, then a new Grant must be obtained. The Grant's operating parameters are maximum EIRP and Channel.

Group: A collection of *CBSDs* which are provided a special, common form of management by the SAS. The nature of the special management is dependent on Group type.

Interference Coordination Group: A group of CBSDs that does not require intra-group, inter-CBSD interference coordination from the SAS.

PAL reserved channel: A 10 MHz channel in the range of 3550-3650 that a SAS may establish for exclusive use of a set of one or more CBSDs that are registered as belonging to a PPA based upon acquired PAL rights.

PAL Protection Area (PPA): An area within a PAL established by a PAL owner for protecting exclusive use of channels based upon the acquisition of PAL rights. The area is based upon the coverage area of the set of CBSDs that are members of the PPA. The SAS maintains a list of CBSDs that are members of the PPA.

REG-Conditional: A parameter in the *RegistrationRequest* object that may be provided by the CBSD or may be provided through other means.

Spectrum Access System (SAS): A system that authorizes and manages use of spectrum for the Citizens Broadband Radio Service.

5 Prerequisites for WInnForum SAS Test Harness

The WInnForum SAS Test Harness is designed to be operating-system independent. It has been tested on Windows (Windows 7 and Windows 10) and Linux platforms.

5.1 Python Installation

5.1.1 Windows Operating System

5.1.1.1 Python 2.7 installation (Python version 2.7.13)

- Go to https://www.python.org/downloads and download for Python 2.7.13 the file "Windows x86 MSI Installer"
- Run the file "python-2.7.13.msi" which will install Python 2.7 and create a directory c:\python27.
- Open Windows Command Prompt using Administrator privileges.
- From the Windows Command Prompt go to directory python27: "cd c:\python27"
- Verify the python version by executing following (without quotes): "python --version". The output should be Python 2.7.13.

5.1.1.2 Python2.7 “Site Packages” directory

When installing python 2.7, the directory c:\python27\Lib\site-packages is empty. Go to https://github.com/Wireless-Innovation-Forum/Citizens-Broadband-Radio-Service-Device and download “site-packages_23_Nov_2017.zip” file. Unzip the file and copy the entire content of the “site-packages” directory to c:\python27\Lib\site-packages.

5.1.2 Linux Operating System

Install Python 2.7 (Python 2.7.13 or later) as per normal Linux installation procedures.

Python package requirements can be found on GitHub in the requirements.txt file. These may be installed via Python pip for package installation.

5.2 WInnForum SAS Test Harness installation

Go to https://github.com/Wireless-Innovation-Forum/Citizens-Broadband-Radio-Service-Device/releases and download the source code (zip and tar.gz versions) for WInnForum software Test Harness for CBSD.

Unzip the contents of the zipped file to a folder of user’s choosing, though minimize total path length.

5.3 allure-cli

allure-cli is required to run the test summary in a graphical display via web GUI [refer to section 14.4]. It is included with the WInnForum SAS Test Harness installation. The allure-cli directory is located under the cbrsPython-master directory.

5.4 WInnForum SAS Test Harness directory structure

After performing the installation steps detailed in this section, the following directory structure exists for WInnForum SAS Test Harness:

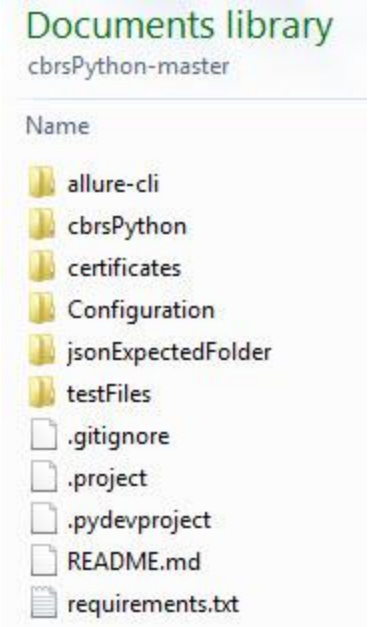

Figure 1: WInnForum SAS Test Harness directory structure

5.5 WInnForum SAS Test Harness Time Synchronization

In order for the SAS Test Harness to operate properly, the server it runs on must be synchronized to UTC time. The WInnForm SAS-CBSD protocol contains messaging which utilizes UTC time, and assumes that both the SAS and CBSD are synchronized to the same time base.

Synchronization should be on the order of 1-second accuracy, which might be achieved via SNTP, for example.

6 Running the WInnForum SAS Test Harness

6.1 Configuration of the WInnForum SAS Test Harness

After installation, the following steps must be completed to configure the SAS Test Harness for operation. Details may be found in later sections.

1. Verify the pre-requisites of the WInnForum SAS Test Harness installation listed in Section 5 are in place.

2. Add test certificates to be used by the SAS Test Harness in the pre-existing cbrsPython-master/certificates directory. See Section 7 for details. This should include:

    - Root CA Public Certificate/Chain of Root CA and Sub-CA

- SAS Public Certificate
- SAS Private Key
- CPI (Certified Professional Installer) Public Certificate (if UUT supports CPI-signed-data)

3. Modify Configuration/conf.xml (see Section 10) as follows:
   - Configure the server IP address and port number to be used for SAS-CBSD messaging. The IP address should match the IP address of the server on which the test harness is running.
   - Modify the certificate names to match those to be used for testing (see step 2, above).
4. Modify cbrsPython-master/Configuration/cbsdc1.xml (and cbsd2.xml) so that the cbsdSerialNumber in the file reflects the CBSD serial number of the UUT. The CBSD serial number should be formatted exactly as the CBSD reports in its registration message. For single-CBSD test cases (WINNF.FT.C.xxx.y), the cbsdSerialNumber within file cbsdc1.xml is expected. For dual-CBSD test cases (WINNF.FT.D.xxx.y), the second CBSD's cbsdSerialNumber must be added to cbsdc2.xml. See Section 10.

6.2 Running the Test Harness

Under Windows, the Test Harness may be run from the Windows Command Prompt using Administrator privileges. Change directory to the cbrsPython-master\cbrsPython\controllers directory and use the command: python .\StartOfProject.py

Under Linux, the Test Harness may be run from the cbrsPython-master/cbrsPython/controllers directory using the command python ./StartOfProject.py

Each test case starts with the UUT in an unregistered state.

The WInnForum SAS Test Harness will prompt the user to input a test name (i.e. 'WINNF.FT.C.REG.1', which corresponds to a CSV test file name (i.e. WINNF.FT.C.REG.1.csv) which resides in the testFiles directory. Enter only the CSV test case file name (without the '.csv extension) and not its full path.

Test case starts → WInnForum SAS Test Harness is waiting for CBSD/Domain Proxy UUT to send the first Request message according to line #2 in the CSV test case file.

At this time, the operator should perform any procedure required to initiate registration, grant, etc. on the UUT.

The Request/Response messages between CBSD/Domain Proxy UUT and WInnForum SAS Test Harness are seen in the Command Prompt window as the test sequence progresses. WInnForum SAS Test Harness validates the CBSD/Domain Proxy UUT Request messages according to the

respective JSON files. If WInnForum SAS Test Harness fails the CBSD/Domain Proxy Request message validation, WInnForum SAS Test Harness replies to CBSD with HTTP Error 400 and fails the test.

After the last SAS-CBSD Request/Response step in the CSV test case file is performed, the questions are invoked. The questions are based upon monitoring of the RF output of the UUT during the test case, so it is important for the operator to understand the monitoring required for each test case prior to running it. Answer the questions based on CBSD/Domain Proxy state and radio conditions.

WInnForum SAS Test Harness then asks for additional free text comment to be included in the log file.

WInnForum SAS Test Harness will present in Command Prompt window the final PASS/FAIL result of the test.

After presenting the "PASS/FAIL" result of the test, WInnForum SAS Test Harness replies to any additional incoming Request messages from CBSD/Domain Proxy UUT certain fixed responses, until the next test case is started. Those responses are described in Section 13.12.

WInnForum SAS Test Harness asks for the next test case file name to be executed.

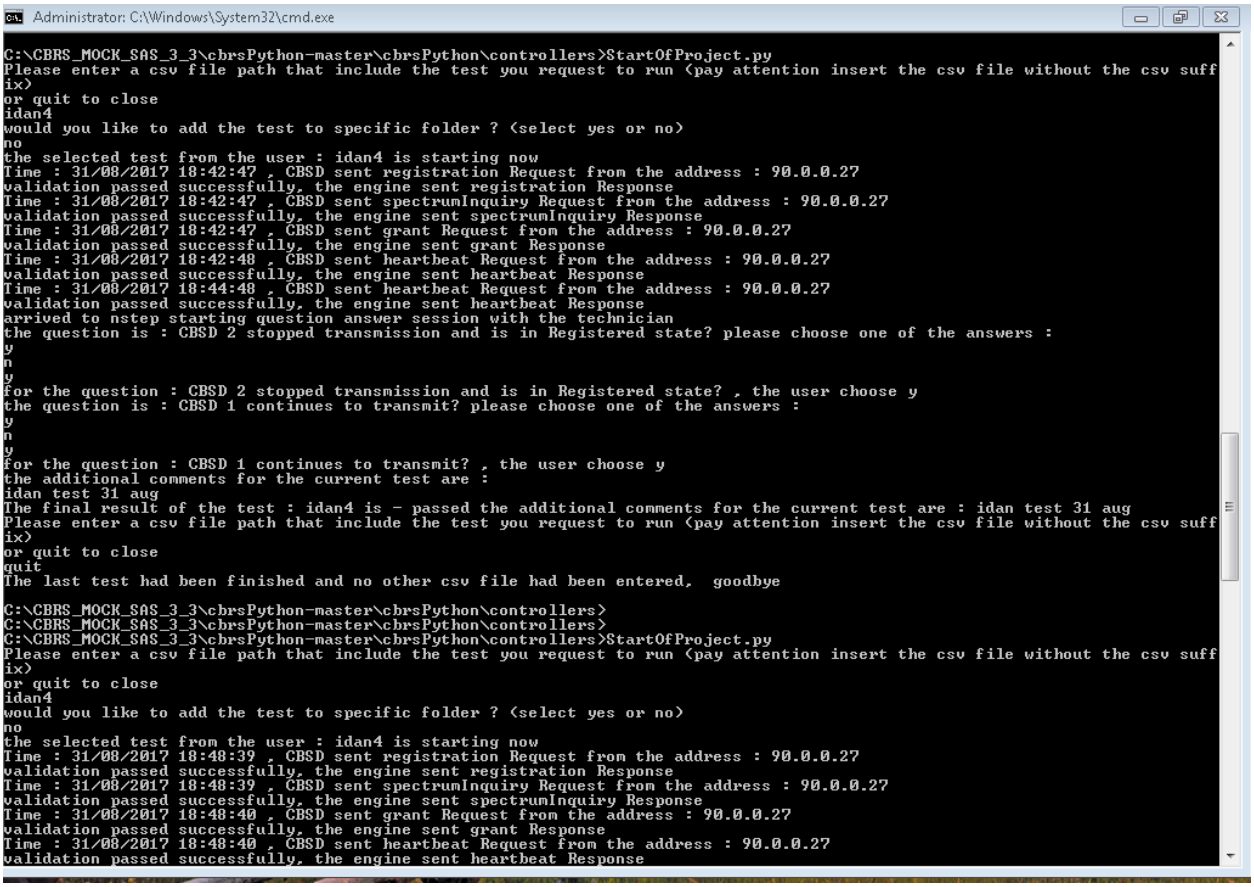

Figure 2: Windows Command Prompt Running WInnForum SAS Test Harness

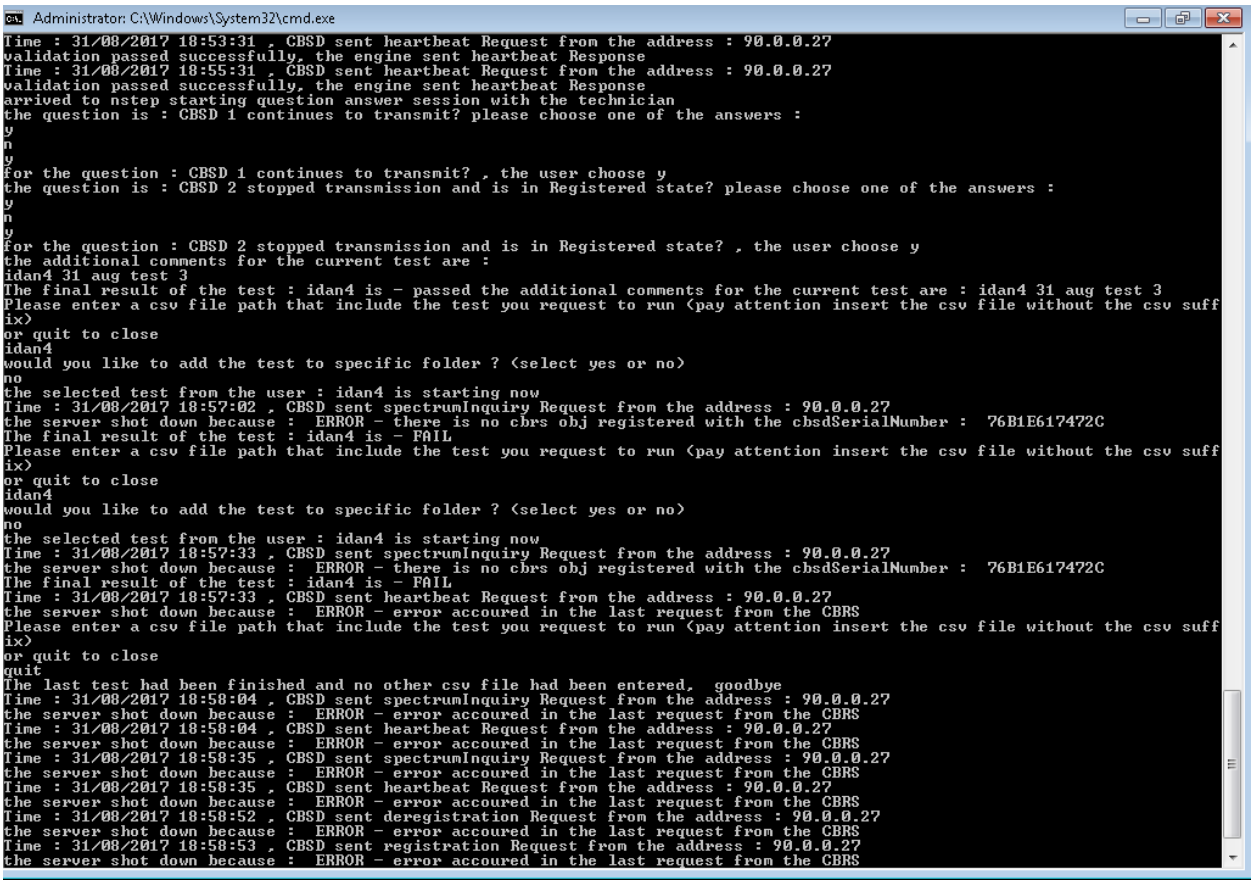

Figure 3: WInnForum SAS Test Harness failing the validation of HTTP 400 error

6.3 Test Flow

To run the test cases, it is important to keep in mind:

- The UUT should be placed in an unregistered mode prior to running each test case, since each test case starts with registration as its first step

- The test harness should be configured to start a test case, and after it has started, the UUT should be triggered to start its transmission process (i.e. registration, grant, heartbeat, etc.).

- After a test case has been completed, the UUT must be placed in UN-Registered mode prior to starting the next test case. This may be accomplished by whatever means are appropriate for the UUT, including resetting the UUT, resetting the state machine of the UUT, or deregistering the UUT with the Test Harness (see Section 13.12 for details on Test Harness message handling between test cases).

- For single-CBSD test cases (WINNF.FT.C.xxx.y), only a single CBSD should attempt to register, and only a single grant will be requested. Attempting to register more than one CBSD or requesting more than one grant is not supported by these test cases.

- For Domain Proxy-CBSD test cases (WINNF.FT.D.xxx.y), two CBSD should attempt to register, and each CBSD should request a single grant only. Attempting to register one, or more than two CBSD, or requesting more than one grant per CBSD is not supported by these test cases.

6.4 Restarting the WInnForum SAS Test Harness

Current implementation of WInnForum SAS Test Harness has an issue in clean closure of the running WInnForum SAS Test Harness threads/processes. Pressing "CTRL ^C" in the Windows Command Prompt may not properly stop the WInnForum SAS Test Harness

If the WInnForum SAS Test Harness needs to restart for any reason in Windows, then close the Windows Command Prompt running the WInnForum SAS Test Harness. Open a new Windows Command Prompt using Administrator privileges and initiate again the WInnForum SAS Test Harness.

For Linux, it may also require closing the terminal window, or alternately "CTRL ^Z" to put the process in background mode, and subsequently killing the process prior to starting the Test Harness again.

6.5 Python Script to Implement WINNF.PT.C.HBT.1 Power Measurement Test Case

The Transmit power measurement test case, WINNF.PT.C.HBT.1, requires a separate script to be executed to support the unique heartbeat required for this test case

Use "StartPowerMeas.py" to start this test case. User is prompted for:

- start frequency (fs) [MHz],
- bandwidth (BW) [MHz], and
- maxEirp (P) [dBm/MHz] for the test case.

Once this has started, UUT may start registration.

The program will only allow a grant for:

- lowFrequency = fs [MHz]
- highFrequency = fs + BW [MHz]
- grant request maxEirp <= P [dBm/MHz]

The Python program is threaded, so after test starts, user may enter "get" to print-back the test parameters, or "stop" to quit the program at any time. The user can also Ctrl-C, as usual, to end the program.

Known issue for this program: “stop” or “get” command inputs do not always work. The user may need to enter command a second time.

7 X.509 certificates

The WInnForum SAS Test Harness comes with an example set of X.509 certificates for TLS authentication with the CBSD/DP UUT. Certificates located in directory cbrsPython-master\certificates.

All certificates are in PEM format.

The WInnForum SAS Test Harness X.509 certificates files are:

- sasharnesspriv.key: this is the X.509 private key of the WInnForum SAS Test Harness
- sasharnesscert.pem: this is the X.509 device certificate of the WInnForum SAS Test Harness
- cbsdpkichain.pem: this is the PKI chain of trust which signed the X.509 device certificate of the CBSD UUT. This needs to be the <cacerts> in the conf.XML file if CBSD is UUT.
- dppkichain.pem: this is the PKI chain of trust which signed the X.509 device certificate of the Domain Proxy UUT. This needs to be the <cacerts> in the conf.XML file if Domain Proxy is UUT.

WInnForum SAS Test Harness also has a set of X.509 certificates to be loaded on the CBSD/Domain Proxy UUT, located in the \certificates\UUT directory.

For CBSD as UUT certificates files are:

- UUTCBSDprivkey.key: this is the X.509 private key of the CBSD UUT
- UUTCBSD.pem: this is the X.509 device certificate of the CBSD UUT
- TestLabSASPKIchain.pem: this is the PKI chain of trust which signed the X.509 SAS device certificate of the SAS Test Harness

For Domain Proxy as UUT certificates files are:

- UUTDomainProxyprivkey.key: this is the X.509 private key of the Domain Proxy UUT
- UUTDomainProxy.pem: this is the X.509 device certificate of the Domain Proxy UUT
- TestLabSASPKIchain.pem: this is the PKI chain of trust which signed the X.509 SAS device certificate of the SAS Test Harness.

The \certificates directory also contains certificates for use when CPI information is supplied.

- cpicert.cer
- cpicert.pem
- UUT\CPI\cpicertprivkey.key

The \certificates\negative_security_test_cases directory contains sub-directories for SAS Test Harness certificates for use with the "Unsuccessful TLS connection" tests described in [n.5]. Only SAS Test Harness private key and SAS Test Harness device certificate need to be replaced in conf.XML file. CBSD/Domain Proxy UUT do not need any replacement in their certificates.

- WINNF.FT.C.SCS.2_TLS failure due to revoked certificate\sascricert.pem
- WINNF.FT.C.SCS.2_TLS failure due to revoked certificate\sascricertprivkey.key
- WINNF.FT.C.SCS.3_TLS failure due to expired server certificate\sasexpiredcert.pem
- WINNF.FT.C.SCS.3_TLS failure due to expired server certificate\sasexpiredcertprivkey.key
- WINNF.FT.C.SCS.4_TLS failure certificate is issued by an unknown CA\unknownsascert.pem
- WINNF.FT.C.SCS.4_TLS failure certificate is issued by an unknown CA\unknownsascertprivkey.key
- WINNF.FT.C.SCS.5_TLS failure certificate is corrupted\sasharnesscertnotvalid.pem
- WINNF.FT.C.SCS.5_TLS failure certificate is corrupted\sasharnesspriv.key

The X.509 certificates can be replaced with the tester's set of X.509 certificates. The X.509 certificates need to be in PEM format.

The PKI chain of trust PEM file contains the Sub-CA and the Root CA.

Turn off the Firewall in order not to block the TLS and HTTPs transaction between WInnForum SAS Test Harness and CBSD/Domain Proxy UUT.

7.1 Creating New PKI and X.509 certificates for SAS Harness Testing

In order to create a new full set of X.509 certificates required for SAS Test Harness, please refer to the instructions detailed in "readme_file_x509_RSA_certs_test_labs.txt" along with the opensslcbrs1.cnf file. These files are located in directory \certificates\How_To_Generate_Certificates. This is intended to be created with a Linux OpenSSL machine using the opensslcbrs1.cnf file instead of the default openssl.cnf file.

Test Lab running the SAS Test Harness generates a Root CA and PKI chain for itself (each test lab effectively has a different PKI chain). The "readme_file_x509_RSA_certs_test_labs.txt" details the certificate generation process in "step 1)" to "step 59)"

CBRS PKI X.509 RSA certificates to be generated by Test Lab running the SAS Test Harness:

1) CBRS Root CA (the test lab creates a Root CA for itself)
2) SAS Provider CA
3) SAS Provider Unknown CA (for negative security test cases)
4) Domain Proxy CA
5) CBSD Manufacturer CA
6) SAS Provider certificate (for SAS Test Harness regular testing)
7) SAS Provider certificate of Unknown CA (for negative security test cases)
8) SAS Provider certificate with expired validity time (for negative security test cases)
9) SAS Provider certificate with CRL extensions (for negative security test cases)

CBRS PKI X.509 RSA certificates to be signed by Test Lab and send to UUT vendor:

1) CBSD UUT (for vendor of CBSD as UUT)
2) Domain Proxy UUT (for vendor of Domain Proxy

8 TLS Mutual Authentication between WInnForum SAS Test Harness and CBSD/Domain Proxy

8.1 TLS Version

WInnForum SAS Test Harness complies with TLS V1.2 according to [n.1]. If CBSD/Domain Proxy UUT sends TLS "Client Hello" message which states only TLS V1.0, then WInnForum SAS Test Harness will reject the TLS connection with a TLS "Alert" message with "level: FATAL, Description: Protocol Version".

8.2 X.509 Mutual Authentication Certificate Validation

WInnForum SAS Test Harness sends to CBSD/Domain Proxy UUT TLS "Server Hello" message which includes "Certificate Request". This is so CBSD/Domain Proxy UUT will send its X.509 certificate to the WInnForum SAS Test Harness.

If CBSD/Domain Proxy UUT does not send its certificate to WInnForum SAS Test Harness, then WInnForum SAS Test Harness will reject the TLS connection with a TLS "Alert" message with "level: FATAL, Description: Handshake Failure".

If CBSD/Domain Proxy UUT sends to WInnForum SAS Test Harness its certificate which is not signed by a Root CA (or PKI chain of trust) that exists on the WInnForum SAS Test Harness,

then WInnForum SAS Test Harness will reject the TLS connection with a TLS “Alert” message with “level: FATAL, Description: Unknown CA”.

8.3 Identification of Failed TLS Connection

When WInnForum SAS Test Harness rejects TLS connection establishment with CBSD/Domain Proxy UUT, WInnForum SAS Test Harness will not report this in its logs. In this case the WInnForum SAS Test Harness will not show any message received from CBSD/Domain Proxy UUT in its CLI window.

9 WInnForum SAS Test Harness Software Description

The figure below provides a high-level view of the Test Harness Software. The main components are:

- Test Engine which implements communications with UUT and runs each test case
- Test Case Definition. Each test case is defined as a series of steps in a .csv file. Each step corresponds to a .json file which indicates the expected request/response parameters for that json message step.
- Configuration files for Test Harness setup of lab-specific or UUT-specific information.
- Log files, which provide detailed logging of each test case run.

Important: For normal test lab operation, only the .xml configuration files should be altered. Altering other files (i.e. .csv, .json, or .py files) may adversely impact operation of the test cases or test engine.

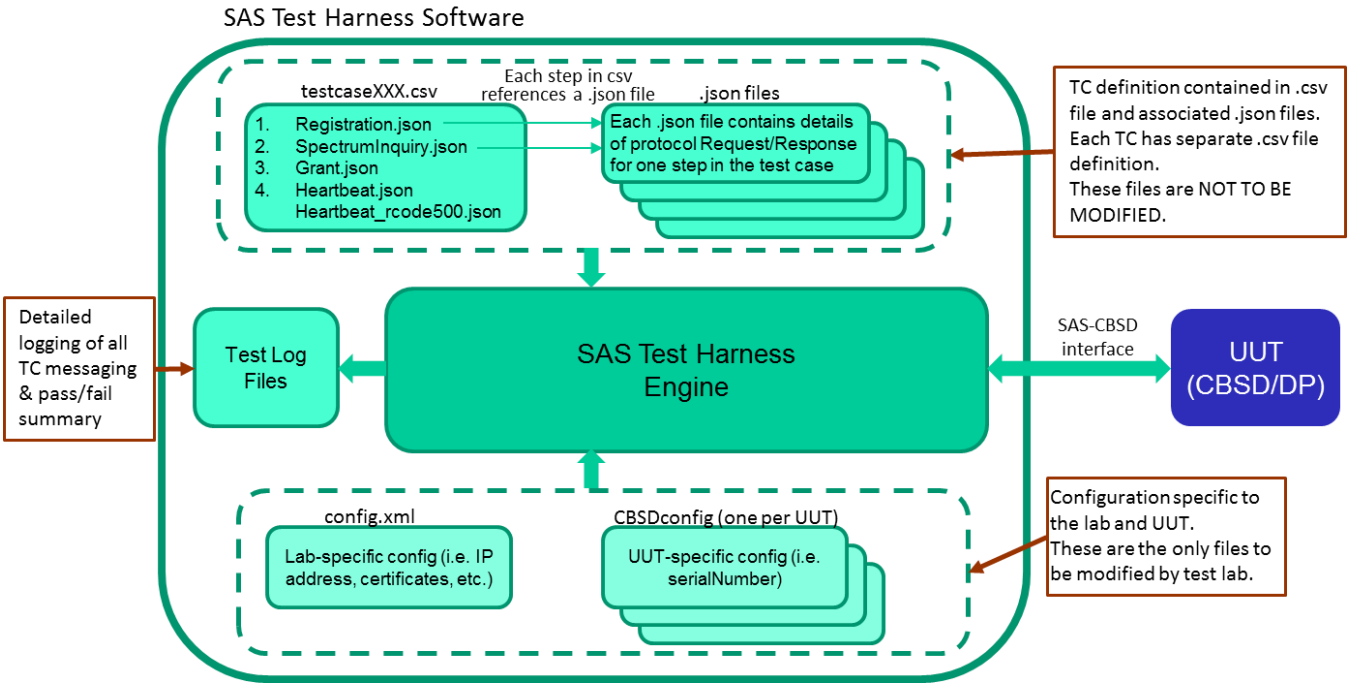

Figure 4: WInnForum SAS Test Harness Components

The following sections provide more details on aspects of the test harness.

10 XML files

10.1 General

WInnForum SAS Test Harness has several XML files for input parameters of test environment configuration and CBSD/Domain Proxy UUT parameters

10.2 XML Configuration File for Test Environment

This "conf.xml" file is located in directory cbrsPython-master\Configuration. It includes the general parameters for the test environment: IP address, port number, certificate file names, etc. The parameter values may need to be changed to reflect the test setup. For instance, the IP address (hostIP) of the PC running the SAS Test Harness.

```xml
<?xml version="1.0"?>
<cbrsParams>
        <jsonsRepoPath>jsonExpectedFolder</jsonsRepoPath>
        <testRepoPath>testFiles</testRepoPath>
        <hostIp>127.0.0.1</hostIp>
        <port>5000</port>
        <heartbeatLimit>30</heartbeatLimit>
        <pemFilePath>certificates/sas_certificate.pem</pemFilePath>
        <keyFilePath>certificates/sas_private_key.key</keyFilePath>
        <caCerts>certificates/rootCA_certificate.pem</caCerts>
        <cpiCert>certificates/cpi_certificate.pem</cpiCert>
</cbrsParams>
```

Figure 5: XML Configuration File for Test Environment

10.3 XML Configuration File for CBSD UUT Data

Each CBSD UUT requires its own XML file located in directory "cbrsPython-master\Configuration\CBSDconfig". It includes the cbsdSerialNumber for the CBSD UUT that WInnForum SAS Test Harness will verify when CBSD UUT sends the Registration Request message. For common practice, the XML file names are cbsdc1.xml (for CBSD #1 UUT), cbsdc2.xml (for CBSD #2 UUT), etc. The cbsdSerialNumber will need to be changed to reflect the UUT's serial number.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cbrsParams>
        <registrationParams>
                <cbsdSerialNumber>D823956047</cbsdSerialNumber>
        </registrationParams>
</cbrsParams>
```

Figure 6: XML Configuration File for CBSD UUT Data

10.4 XML Configuration File for CBSD Air Interface Options

CBSD/Domain Proxy UUT is required to specify its Air Interface in the Registration Request message. WInnForum SAS Test Harness validates the Air Interface parameter received in the Registration Request from the CBSD/Domain Proxy UUT according to the valid entries listed in the file "cbrsPython-master\Configuration\ObjectListConfig\laterDefindedOptionaljson". The radioTechnology field can be updated according to the latest version of [n.4].

```
"airInterface": {
   "type": "object",
   "description": "Air interface parameter for a CBSD",
   "required": [ "radioTechnology" ],
   "properties": {
     "radioTechnology": { "enum": [ "E_UTRA", "CAMBIUM_NETWORKS" ],
                          "description": "Which PHY radio technology is used by the device in the 3.5GHz band" }
               }
          }
```

Figure 7: XML Configuration File for Air Interface Options

10.5 XML Configuration File for CBSD Grouping Options

CBSD/Domain Proxy UUT may optionally specify its Grouping in the Registration Request message. WInnForum SAS Test Harness validates the Grouping parameter received in the Registration Request from CBSD/Domain Proxy UUT according to the valid entries listed in the file “cbrsPython-master\Configuration\ObjectListConfig\laterDefindedOptionaljson”. The groupingParam fields can be updated according to the latest version of [n.4].

```
  "groupingParam": {
    "description": "Container for information about a CBSD membership group.",
    "type": "array",
    "items": {
        "type": "object",
        "required": [ "groupType", "groupId" ],
        "properties": {
                "groupType": {
                        "enum": [ "INTERFERENCE_COORDINATION" ],
                    "description": "Type of the group"
                   },
                "groupId": {
                        "type": "string",
                                "maxLength" : 128,
                    "description": "ID of the group"
                    }
         }
      }
   },
```

Figure 8: XML Configuration File for groupingParam Options

11 CSV Test Case files

11.1 General

The CSV test case file is running the SAS-CBSD desired sequence of Request/Response messages for the desired test case according to [n.5]. The CSV test case files are located at directory “cbrsPython-master\testFiles”.

Each line in the CSV test case file corresponds to a specific JSON file name located at “cbrsPython-master\jsonExpectedFolder”. The last line in the CSV test case file is considered by the WInnForum SAS Test Harness to be the last step in the test sequence. WInnForum SAS Test Harness will invoke the questions written in the JSON file appearing in the last line of CSV test case file.

Line 2 in the CSV test case file is the test sequence first SAS-CBSD Request/Response message. Line 3 in the CSV test case file is the test sequence second SAS-CBSD Request/Response message. Line #N+1 in the CSV test case file is the test sequence #N SAS-CBSD Request/Response message.

The CSV test case file contains all the test sequence steps required for the CBSD to reach the last step. The last step in the CSV file is actually the desired test case.

If a Domain Proxy is UUT then each column in the CSV test case file corresponds to a specific CBSD controlled by the Domain Proxy. There is no limit in WInnForum SAS Test Harness for number of columns in CSV test case files (number of CBSDs controlled by Domain Proxy). If the Request message from Domain Proxy UUT is for individual CBSD, WInnForum SAS Test Harness will reply with individual Response. If the Request message from Domain Proxy UUT is a combined JSON array for several CBSDs, WInnForum SAS Test Harness will reply with a combined JSON Array Response.

For common practice, the CSV test case file name corresponds to a specific section number (specific test case) according to [n.5].

	A
1	jsonFileName
2	HBT_registration_Cbsd1.json
3	HBT_spectrumInquiry_Cbsd.json
4	HBT_grant_Cbsd.json
5	HBT_heartbeat_GrantedState.json
6	HBT_heartbeat_AuthState.json
7	HBT_heartbeat_AuthState.json
8	HBT_heartbeat_AuthState.json
9	HBT_heartbeat_AuthState.json
10	HBT_heartbeat_AuthState.json

Figure 9: CSV test case file example for single CBSD

	A	B
1	jsonFileName	jsonFileName2
2	HBT_registration_Cbsd1.json	HBT_registration_Cbsd2.json
3	HBT_spectrumInquiry_Cbsd.json	HBT_spectrumInquiry_Cbsd.json
4	HBT_grant_Cbsd.json	HBT_grant_Cbsd.json
5	HBT_heartbeat_GrantedState.json	HBT_heartbeat_GrantedState.json
6	HBT_heartbeat_AuthState.json	HBT_heartbeat_AuthState2.json
7	HBT_heartbeat_AuthState.json	HBT_heartbeat_AuthState2.json
8	HBT_heartbeat_AuthState.json	HBT_heartbeat_AuthState2.json
9	HBT_heartbeat_AuthState.json	HBT_heartbeat_AuthState2.json
10	HBT_heartbeat_AuthState.json	HBT_heartbeat_AuthState2.json

Figure 10: CSV test case file example for Domain Proxy UUT controlling 2 CBSDs

12 JSON files

12.1 General

Each JSON file contains only a single SAS-CBSD protocol Request message and a single SAS-CBSD protocol Response message. For example, there is a JSON file for SAS-CBSD Registration Request/Response and another JSON file for SAS-CBSD Grant Request/Response.

The JSON files are located in directory “cbrsPython-master\jsonExpectedFolder”.

If a Domain Proxy is UUT, then each CBSD controlled by the Domain Proxy requires a separate JSON file containing only a single SAS-CBSD protocol Request message and a single SAS-CBSD protocol Response message for that CBSD. If the Domain Proxy UUT controls N CBSDs, then the “jsonExpectedFolder” needs to have N individual JSON files for each Request/Response message for each CBSD.

If the Request message from Domain Proxy UUT is for individual CBSD, WInnForum SAS Test Harness will reply with an individual Response. If the Request message from Domain Proxy UUT is a combined JSON array for several CBSDs, WInnForum SAS Test Harness will reply with a combined JSON Array Response.

{CBSD as UUT} or {Domain Proxy UUT controlling just one CBSD} are the same for WInnForum SAS Test Harness.

12.2 Full Request URI format in CBSD Request messages HTTP Header

WInnForum SAS Test Harness expects the following Full Request URI format according to [n.3]: https://<WInnForum SAS Test Harness IP address:port>/v1.2/<message>

For example: https://70.14.0.72:5000/v1.2/grant https://70.14.0.72:5000/v1.2/spectrumInquiry

12.3 Optional Parameters in Request message from CBSD/Domain Proxy UUT

The Request message from the CBSD/Domain Proxy UUT may or may not include parameters according to [n.3]. WInnForum SAS Test Harness validates the Request message from CBSD/Domain Proxy UUT in the following order:

- Priority #1 (highest): WInnForum SAS Test Harness attempt to match parameters in the Request message according to the JSON file which is placed directly in the "jsonExpectedFolder"
- Priority #2: If a parameter existing in the Request message was not found according to priority #1 then the WInnForum SAS Test Harness shall attempt to match the additional parameter in the Request message according to the JSON file which is placed in the "jsonExpectedFolder\OptionalParams" or in the "cbrsPython-master\Configuration\ObjectListConfig\laterDefindedOptional.json" file
- If a parameter existing in the Request message was not found according to priority #1 and priority #2 above, then WInnForum SAS Test Harness will fail the validation of the Request message from CBSD/Domain Proxy UUT.
- If a parameter in the Request message appears both in the JSON file placed in the "jsonExpectedFolder", and in the JSON file which is placed in the "jsonExpectedFolder\ OptionalParams" or "laterDefindedOptional.json" file, then WInnForum SAS Test Harness will validate the parameter based on the JSON file placed directly in the "jsonExpectedFolder".
- If a parameter in the Registration Request message appears both in the JSON file placed directly in the "jsonExpectedFolder", and in the XML configuration file for CBSD/Domain Proxy UUT data, then WInnForum SAS Test Harness will validate the parameter based on the JSON file placed directly in the "jsonExpectedFolder".

For each SAS-CBSD protocol Request message there exists a single JSON file in the "jsonExpectedFolder\OptionalParams" directory with a specific name.

12.4 Range of Values for Parameters in Request message from CBSD/Domain Proxy UUT

The parameters in the Request messages can have a range of values according to [n.3]. JSON files for the WInnForum SAS Test Harness can also be configured with a range of values. A range of values is applicable both for the JSON file which is placed directly in the "jsonExpectedFolder", and for the JSON file which is placed in the "jsonExpectedFolder\OptioanlParams" and the "laterDefindedOptional.json" file.

12.5 JSON File Name

The JSON file name is written at the first line of the JSON file as “name”:”<JSON file information>”.

12.6 JSON File Questions for PASS/FAIL criteria

WInnForum SAS Test Harness will apply the questions written in the JSON file for the final PASS/FAIL criteria of the test. The questions must be in format of y/n (yes/no) response for a user to enter in the CLI of the WInnForum SAS Test Harness.

The WInnForum SAS Test Harness will display the questions written in the JSON file only if this JSON file is the last SAS-CBSD message in the test sequence.

The questions correspond to the last step in the specific test section of [n.5] which describes the required radio and CBSD conditions.

13 WInnForum SAS Test Harness Handling of SAS-CBSD Procedures

13.1 General

WInnForum SAS Test Harness appears to the CBSD/Domain Proxy UUT as a SAS and is required to support several procedures according to [n.3].

13.2 Handling of cbsdId and grantId

For all the JSON files located in the “jsonExpectedFolder”, there is no need to specify the parameters of cbsdId and grantId in the Request and Response messages appearing in the JSON file. The JSON file for Registration Request/Response must contain the relevant file name of the XML configuration file for CBSD UUT data using the line “xmlFilelLinked”: “cbsdc1” (without the .xml suffix of the file name). The line “xmlFilelLinked”: “cbsdc1” needs to appear only in the Registration JSON and not in the JSON files for other messages.

If WInnForum SAS Test Harness successfully validated the Registration Request from CBSD/Domain Proxy UUT then WInnForum SAS Test Harness will automatically generate a cbsdId in the format <fccId>Mock-SAS<cbsdSerialNumber> from the parameters received in the Registration Request message. WInnForum SAS Test Harness will use this cbsdId for subsequent Request/Response messages in the test sequence for this CBSD/Domain Proxy UUT.

If WInnForum SAS Test Harness successfully validates the Grant Request from CBSD/Domain Proxy UUT then WInnForum SAS Test Harness will automatically generate a grantId which is a random selected number. WInnForum SAS Test Harness will use this grantId for subsequent Request/Response messages in the test sequence for this CBSD/Domain Proxy UUT.

```
{
    "name":"REG_MultiStep_Success_cbsd1",
        "xmlFilelLinked": "cbsdc1",
    "questions":
                                [
                                        {
                                                "question" : "Were there RF transmissions from the CBSD1 during the test?",
                                                "answers" : ["y", "n"],
                                                "expectedAnswer" : "n"
                                        }
                                ],
                        "registrationRequest":
                [
                {
                                  "userId":"${maximumLength:128}",
                                  "fccId":"${maximumLength:128}"
                                }
                        ],
                        "response":{
                        "registrationResponse" :
                                [
                                        {
                                          "response": {
                                                 "responseCode": 0
                                          }
                                        }
                                ]
                                }

}
```

Figure 11: Registration JSON file in "jsonExpectedFolder"

13.3 Repeated Successful Heartbeat Request/Response

For some test cases in [n.5], the CBSD will be in "Authorized" state sending repeatedly Heartbeat Requests. The CBSD expects successful Heartbeat Responses from the WInnForum SAS Test Harness to remain in "Authorized" state. To execute the JSON file of Heartbeat Request/Responses repeatedly, the line "repeatsAllowed": "True" needs to be present in the Heartbeat JSON file.

Current WInnForum SAS Test Harness implementation requires that Heartbeat Repeats JSON file in "jsonExpectedFolder" will include all the parameters expected in the Heartbeat Request message from CBSD/Domain Proxy UUT either with specific value or with range values.

```
{
    "name":"HBT_heartbeat_AuthorizedState_Repeats",
    "repeatsAllowed":"True",
        "heartbeatRequest":
            [
            {
                "operationState": "AUTHORIZED"
            }
            ],
    "response":
            {
            "heartbeatResponse":
                [
                {
                    "response" : {
                        "responseCode" : 0
                        }
                }
        ]
    }


}
```

Figure 12: Heartbeat Repeats JSON file in "jsonExpectedFolder"

13.4 Heartbeat Absent functionality

HBT Absent functionality is supported by delaying all test harness responses, once triggered in the appropriate step, until the end of the test. There can be multiple responses sent after the test is complete, but no responses will be send during the test case.

HBT absent test cases are indicated in the HBT jsonExpected file by adding the flag "isAbsentResponse": "True". This adds a large delay to the HBT response message, to simulate HBT response absent.

Known issue: operates properly for one CBSD only. Does not operate properly for two or more CBSD. Since the HBT Absent test cases in v1.1 are single-CBSD only, this functionality should be sufficient, although the multiple-CBSD case should be considered in a future enhancement.

13.5 Repeated Successful Spectrum Inquiry Request/Response

A Spectrum Inquiry Request can come from the CBSD/Domain Proxy UUT at any time after CBSD is in "Registered" state. The CBSD/Domain Proxy UUT may send several Spectrum Inquiry requests during its normal operation. CBSD expects successful Spectrum Inquiry Responses from the WInnForum SAS Test Harness. To execute the JSON file of Spectrum Inquiry Request/Responses repeatedly, the line "repeatsAllowed": "True" needs to be present in the Spectrum Inquiry JSON file.

Current WInnForum SAS Test Harness implementation requires that Spectrum Inquiry Repeats JSON file in "jsonExpectedFolder" will include all the parameters expected in the Spectrum

Inquiry Request message from CBSD/Domain Proxy UUT either with specific value or with range values.

13.6 First Spectrum Inquiry Request as an Optional Procedure in the Test Sequence

Spectrum Inquiry may be an optional procedure not mandated at all by the specific test case sequence of [n.5]. In this case it is not mandated that the first Spectrum Inquiry Request will come at a specific place in the test sequence (for example the first Spectrum Inquiry Request from CBSD/Domain Proxy UUT may come after Grant Request/Response and not necessarily after Registration Request/Response). To execute the JSON file of first Spectrum Inquiry Request as an optional procedure that can come at any time, the line “appearAlways”: “True” needs to be present in the Spectrum Inquiry JSON file.

This JSON file must appear in the CSV file for the CBSD/Domain Proxy UUT. For common practice, it can appear as a step after the “Registration” step in the CSV file. If the line “appearAlways”: “True” is present in the Spectrum Inquiry JSON file then WInnForum SAS Test Harness will accept the first Spectrum Inquiry Request at any point during the test sequence. If the line “appearAlways”: “True” is not present then WInnForum SAS Test Harness will accept the first Spectrum Inquiry Request at the specific point according to the step sequence in the CSV file.

Current WInnForum SAS Test Harness implementation requires that Spectrum Inquiry JSON file with “appearAlways”: “True” in the “jsonExpectedFolder” will include all the parameters expected in the Spectrum Inquiry Request message from CBSD/Domain Proxy UUT either with specific value or with range values.

```
{
    "name":"HBT_spectrumInquiry_FullMeasRep",
    "questions": {},
    "appearAlways":"True",
    "measReportRequested" : "True",
    "fullBandReport" : "True",
    "spectrumInquiryRequest":
          [
              {
                "measReport": {
                     "rcvdPowerMeasReports": [
                       {
                          "measBandwidth": "${range:1To10000000}",
                          "measRcvdPower":  "${range:-100To-25}",
                          "measFrequency": "${range:3550000000To3699999999}"
                       }
                       ]
                  }
              }
          ],
    "response":
          {
           "spectrumInquiryResponse":[
           {
           "response" : {
               "responseCode" : 0
                }
          }
          ]
    }
}
```

Figure 13: Spectrum Inquiry JSON file in "jsonExpectedFolder"

13.7 Relinquishment Request as an Optional Procedure in the Test Sequence

Relinquishment Request may be an optional procedure for CBSD/Domain Proxy UUT prior to Deregistration Request. In this case a test case of [n.5] for Deregistration Request/Response may not mandate that CBSD/Domain Proxy UUT will send Relinquishment Request prior to Deregistration Request, however the CBSD/Domain Proxy UUT does send Relinquishment Request. If the next step in the CSV file is "Deregistration" and "Relinquishment" does not appear as a step in the CSV file, WInnForum SAS Test Harness will accept a Relinquishment Request from CBSD/Domain Proxy UUT only if the Relinquishment Request is followed by the Deregistration Request.

13.8 Handling of Grant Expiry Time, Transmission Expiry Time and Grant Renew

For all the JSON files located in the "jsonExpectedFolder", there is no need to put the parameters of grantExpiryTime and transmitExpiryTime in the Response messages appearing in the JSON file.

For the Heartbeat JSON files located in the "jsonExpectedFolder", there is a need to put the parameters of grantRenew in the Request message appearing in the JSON file.

WInnForum SAS Test Harness automatically calculates the grantExpiryTime for the Grant Response as <current UTC time of test execution> + <SECONDS_TO_ADD_FOR_GRANT_EXPIRE_TIME> appearing in the Consts.py file at cbrsPython-master\cbrsPython\model\Utils

WInnForum SAS Test Harness automatically calculates the transmitExpiryTime for the Heartbeat Response as on <current UTC time of test execution> + <SECONDS_TO_ADD_FOR_TX_EXPIRE_TIME > appearing in the Consts.py file.

If CBSD/Domain Proxy UUT includes “grantRenew”: “true” in the Heartbeat Request, then WInnForum SAS Test Harness will include in the subsequent Heartbeat Response the updated grantExpiryTime as <current UTC time of test execution> + <SECONDS_TO_ADD_FOR_GRANT_EXPIRE_TIME>.

WInnForum SAS Test Harness will support a short grantExpireTime for grant expiry test case. In GRA jsonExpected file, the flag “shorterGrantTime”: “True” is used to indicate the constant SHORTER_GRANT_EXPIRE_TIME be used instead of the normal grantExpireTime used for typical test cases.

If the Heartbeat JSON file located in the “jsonExpectedFolder”, is for an unsuccessful scenario according to [n.5], meaning the JSON for Heartbeat Response has a non-zero value for responseCode, WInnForum SAS Test Harness automatically gives the transmitExpiryTime for the Heartbeat Response as <current UTC time of test execution>.

Certain HBT non-zero responseCodes test cases will trigger the transmitExpireTime to be returned as current time (i.e. stop transmitting NOW). The list of non-zero responseCodes includes [500, 501, 502, 105].

For Grant Renew handling, current WInnForum SAS Test Harness implementation requires that “grantRenew” parameter will appear in the Heartbeat JSON located in the “jsonExpectedFolder” and not in the “optionalParams” folder. The “grantRenew” parameter can be with a specific value or range of values.

Current WInnForum SAS Test Harness implementation does not handle range of values for “grantRenew” in the Heartbeat JSON file if the “repeatsAllowed”: “True” exists in the Heartbeat JSON file. In this case only a unique value for “grantRenew” parameter can be specified in the JSON file.

Following the SAS-CBSD protocol, transmitExpireTime must not exceed grantExpireTime. Therefore, notwithstanding the information above, the transmitExpireTime provided in the heartbeatResponse is always limited to be no more than the grantExpireTime if the grantExpireTime is close to expiring.

```
{
    "name":"NMS Test: Initial Heartbeat",
    "questions":
    [
        {
            "question" : "OK?",
            "answers" : ["y", "n"],
            "expectedAnswer" : "y"
        }
    ],
    "heartbeatRequest":
    [
        {
            "grantRenew":{"$or":["true","false"]}

        }
    ],
    "response":
    {
        "heartbeatResponse":
        [
            {
                "heartbeatInterval":120,
                "response": {"responseCode": 0}

            }
        ]
    }
}
```

Figure 14: Heartbeat JSON file in "jsonExpectedFolder" with "grantRenew" parameter

13.9 Handling of airInterface Parameter in Registration Request

CBSD/Domain Proxy UUT sends the airInterface parameter in Registration Request. The possible values for airInterface parameters are listed in [n.4]. For all the Registration JSON files located in the "jsonExpectedFolder" and "jsonExpectedFolder\OptionalParams" directory, there is no need to put the parameters of airInterface in the Registration Request messages. WInnForum SAS Test Harness will validate the values of airInterface parameter in Registration Request based on the list appearing in the "laterDefindedOptional.json" file for CBSD air interface options.

13.10 Handling of groupingParam in Registration Request

CBSD/Domain Proxy UUT sends the groupingParam in Registration Request. The possible values for groupingParam are listed in [n.4]. For all the Registration JSON files located in the "jsonExpectedFolder" and "jsonExpectedFolder\OptionalParams" directory, there is no need to put the parameters of groupingParam in the Registration Request messages. WInnForum SAS Test Harness will validate the values of groupingParam in Registration Request based on the list appearing in the "laterDefindedOptional.json" file for CBSD Grouping Parameter options.

13.11 Handling of measReport in Request Messages

CBSD/Domain Proxy UUT may include measReport parameter in various Request messages. For all the JSON files located in the "jsonExpectedFolder", there is no need to put the parameter of measReport in the Request messages appearing in the JSON file. WInnForum SAS Test Harness will validate all the measReport appearing in the various Request messages with the specific JSON file located in the "jsonExpectedFolder\OptionalParams" directory. The WInnForum SAS Test Harness will validate each object if the measReport parameter in the Request message contains an array of several objects.

```
"measReport": {
    "description": "Measurement report provided by a CBSD.",
    "type": "object",
    "additionalproperties": true,
    "properties": {
        "rcvdPowerMeasReports": {
        "type": "array",
        "description": "Report structured used by a CBSD when a measurement report type is EutraCarrierRssiNonTx or EutraCarrierRssiAl
        "items": {
            "type": "object",
            "description": "Report for a particular Carrier RSSI measurement.",
            "required": [ "measFrequency", "measBandwidth", "measRcvdPower" ],
            "properties": {
                "measFrequency": {
                    "type": "number",
                    "description": "Frequency of the low end of the measured channel in Hz.",
                    "minimum": 3550000000,
                    "maximum": 3700000000
                   },
                "measBandwidth": {
                    "type": "number",
                    "description": "Measurement bandwidth of the channel in Hz.",
                    "minimum": 0,
                    "maximum": 10000000
                  },
                "measRcvdPower": {
                     "type": "number",
                     "description": "Carrier RSSI measurement of the channel in dBm as per 3GPP TS 32.592.",
                     "minimum": -100,
                     "maximum": -25
                       }
                }
```

Figure 15: MeasReport information in a JSON file in "jsonExpectedFolder\OptionalParam"

To request a measurement report, the "measReportConfig" parameter needs to be in the response of the appropriate jsonExpected file.

To check measReport in the request message:

- include measReport object with single rcvdPowerMeasReports object defining the measReport ranges
- Add flag "measReportRequested":" True" to jsonExpected file where measReport is expected, to activate pass/fail check of measurement report for that message
- For response in HBT message, code checks for up to 5 HBT requests for the measReport.

An additional check is made for measReport type: RECEIVE_POWER_WITHOUT_GRANT. The specification calls for the measReport to cover the entire CBRS band. Therefore, all the elements of the rcvdPowerMeasReports are checked to ensure full-band coverage. This additional check is triggered by adding the flag "fullBandReport": "True" to the jsonExpected file where the measReport is expected (i.e. SIQ and GRA message types).

13.12 Handling of Request Messages after Test Sequence Concluded

WInnForum SAS Test Harness HTTPs server cannot be stopped while the WInnForum SAS Test Harness is running, hence the TLS connection with CBSD/Domain Proxy UUT continues to be successful and CBSD/Domain Proxy UUT can continue sending Request messages after the steps listed in the CSV file are finished.

Each test case consists of a set of protocol exchanges as set out within the relevant test *.csv and *.json files. From the point at which the last protocol step in the test is complete (the point at which the questions prompt appears) until the next test case as started, the Test Harness will respond to any request messages from the UUT with pre-conditioned responses. These responses are intended to implement the test requirements that, for instance, no further successful registration or grants may be obtained by the CBSD immediately after a test has completed, but prior to start of the next test case.

This behavior also allows a UUT to relinquish/deregister after a test case has completed, in order for it to be prepared for the next test case, although this may also be accomplished by other means.

The exact behavior of the Test Harness to request messages in the period between active test cases being executed is outlined as follows:

Request Msg	**Response Msg**	**Description**
registrationRequest	registrationResponse, responseCode = 200 (REGISTRATION_PENDING)	Test Harness does not allow new registration.
spectrumInquiryRequest	spectrumInquiryResponse responseCode = 0, availableChannel 3550-3555 MHz	
grantRequest	grantResponse, responseCode = 400 (INTERFERENCE)	Test Harness does not allow new grants.
heartbeatRequest	heartbeatResponse: if request operationState = "GRANTED", responseCode = 501 (SUSPENDED_GRANT)	Test Harness allows AUTHORIZED UUT to continue heartbeating in AUTHORIZED state, or GRANTED UUT to continue heartbeating in

	if request operationState = "AUTHORIZED", responseCode = 0	GRANTED state
relinquishmentRequest	relinquishmentResponse: responseCode = 0	Test Harness allows UUT to relinquish a grant.
deregistrationRequest	deregistrationResponse: responseCode = 0	Test Harness allows UUT to deregister

Figure 16: Test Harness Behavior Between Test Cases

13.13 Handling of Request Message Validation Failure

During the test sequence, if WInnForum SAS Test Harness fails the validation of a Request message from CBSD/Domain Proxy UUT, then WInnForum SAS Test Harness replies with HTTP 400 message, fails the test and waits for a new CSV file to run a new test case. Request message validation failure can occur in WInnForum SAS Test Harness if the Request message format and parameter values are not according to the WInnForum SAS Test Harness JSON files, or if the Request message does not match the next step in the CSV file.

13.14 CPI signature checking in registration message

For registration messages containing cpiSignatureData, the code supports checking as follows:

- Regular checking of jsonExpected and registrationOptional.json of the cpiSignatureData object (i.e. checks for protectedHeader, encodedCpiSignedData, and digitalSignature parameters).
- Check signature algorithm used (as listed in protectedHeader) is "RS256" or "ES256"
- Verify signature of cpiSignatureData
- Decode the payload, and check it against the cpiSignatureDataSchema.json schema (in jsonExpectedFolder/OptionalParams).

Keep in mind the following notes:

- cpiSignatureData should be signed by CPI private key.
- CPI certificate is required to verify signature. It's file path/name is added to parameters in conf.xml. The Test Harness derives the CPI public key from the CPI certificate file and uses that to verify the signature.

14 Log files

14.1 General

WInnForum SAS Test Harness generates and stores 2 types of log files:

- Log file of Windows Command Prompt running WInnForum SAS Test Harness
- Log file of individual test cases

14.2 Log file of Windows Command Prompt running WInnForum SAS Test Harness

This log file saves all the printouts of the Windows Command Prompt for as long as the WInnForum SAS Test Harness is running. The log file is saved with timestamps taken from the local time of the PC running the WInnForum SAS Test Harness.

This log file is stored at directory c:\...\cbrsPython-master\Logs\CMDSessions. Log file name includes the start time.

14.3 Log file of Individual Test Cases

This log file is generated and stored for each individual test case. It shows full JSON parsing of the SAS-CBSD messages. The log file is saved with timestamps taken from the local time of the PC running the WInnForum SAS Test Harness.

This log file is stored at directory c:\...\cbrsPython-master\Logs\LogsPerTest.

14.4 Web GUI summary display of test cases using allure-cli utility

It is possible to view a graphical web GUI summary display of the test cases which is updated on the fly as the test cases progress in WInnForum SAS Test Harness. This is done using the allure-cli utility.

For the web GUI display, a dedicated directory name is required in c:\...\cbrsPython-master\Logs\SpecificFolderOfTests.

At the beginning of the test, WInnForum SAS Test Harness asks if a dedicated directory name is required. The user needs to input only the name, not the full path. WInnForum SAS Test Harness will create under the dedicated directory an additional 3 directories: “chrome”, “firefox”, “xml”.

The web GUI summary is supported only by firefox and chrome web browsers. To view the web GUI summary in chrome web browser double click the “chromeReport” file located in c:\...\cbrsPython-master\Logs\SpecificFolderOfTests\<name>\chrome

It is possible to view the web GUI summary report on the fly as test execution progresses in WInnForum SAS Test Harness, and after test executions have finished.

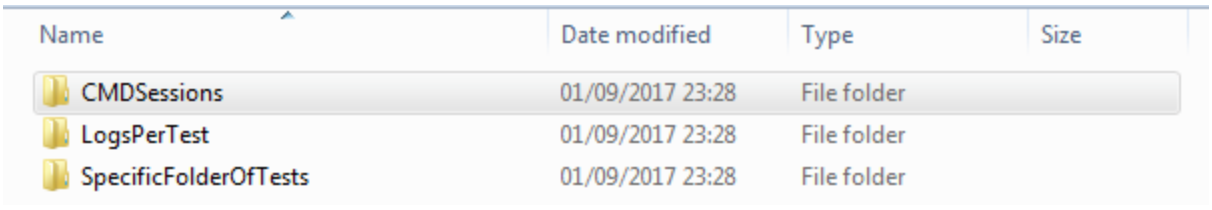

Figure 17: Content of "c:\...\cbrsPython-master\Logs" directory

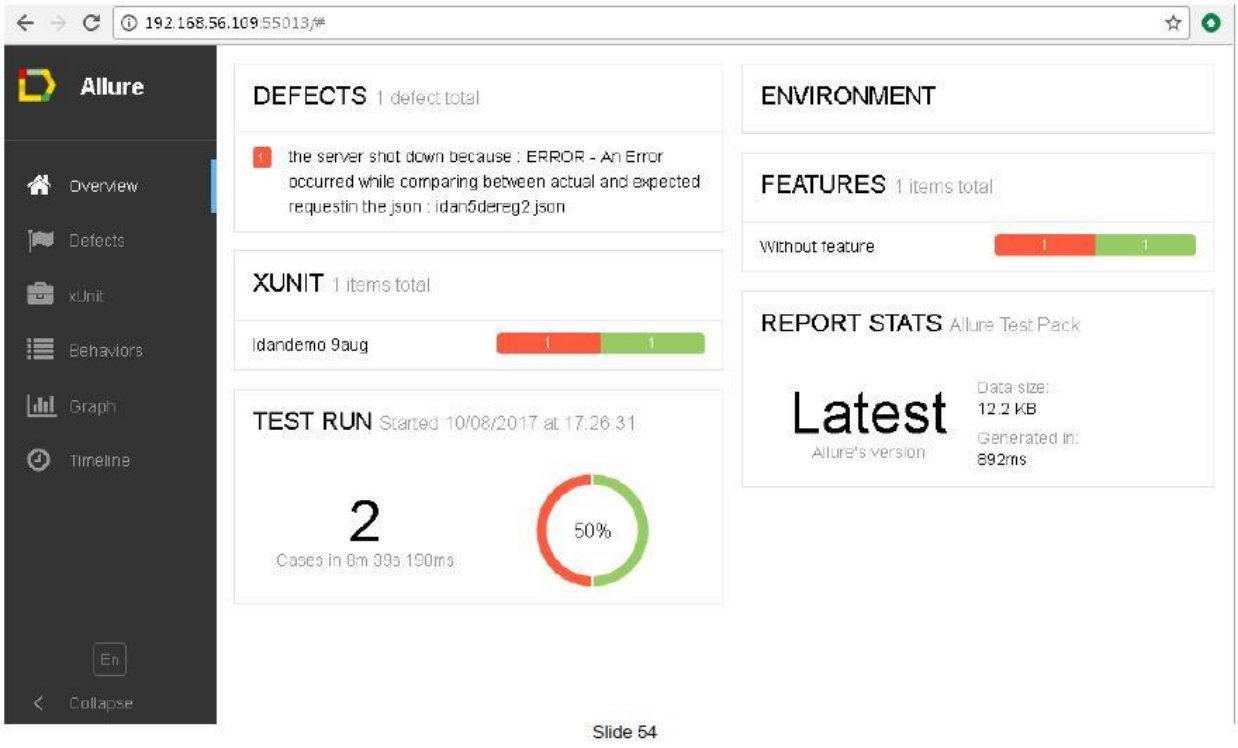

Figure 18: Web GUI summary of WInnForum SAS Test Harness test cases