Wireless Innovation Forum Contribution

[Note: This page is removed once the document has been balloted and approved]

Committee: Spectrum Sharing Committee, Working Group 4

Title: Test and Certification for Citizens Broadband Radio Service (CBRS); Conformance and Performance Test Technical Specification; CBSD as Unit Under Test (UUT) – DRAFT, Request for Comment

Short Title: DRAFT: CBRS Test and Certification TS Request for Comment

Rapporteur: Masoud Olfat, PhD
Federated Wireless
Masoud.Olfat@FederatedWireless.com

Awaiz Ahmad Khan
Ruckus Wireless
Awaiz.khan@ruckuswireless.com

Date: 23 August 2017

Distribution: Unrestricted, Members

Document Summary: The present document contains the Protocol Implementation Conformance Statement (PICS), test cases to ensure conformance of entities for Citizens Broadband Radio Service (CBRS) which implement signaling protocols and procedures specified in [n.5] and the requirements mandated by 47 C.F.R Part 96 [n.1], [n.2] for CBSD as Unit Under Test (UUT).

Notes of Importance: Required for uniform test and certification of the components of CBSD Functional Architecture

Action Desired: Request for Comment

Action Required for Closure: Present for Consideration to FCC OET. Comments to be provided via the WInnForum supplied form (attached).

Desired Disposition Date: 14 September 2017

Test and Certification for Citizens Broadband Radio Service (CBRS); Conformance and Performance Test Technical Specification; CBSD/DP as Unit Under Test (UUT)

Working Document WINNF-TS-0122

Version V0.0.0-r4.0

23 August 2017

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 It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the SSC Work Group 4.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.

Wireless Innovation Forum ™, WInnForum Standards™ and SDR Forum ™ are trademarks of the Software Defined Radio Forum Inc.

Table of Contents

List of Figures

List of Tables

Contributors

Editor and Task Group Chair: Awaiz Khan, Ruckus Wireless

Other Member Representatives

- Doug Goedken, Nokia
- Idan Raz, AirSpan
- Chris Williams, Ericsson

Test and Certification for CBRS; Conformance and Performance Test Technical Specification; CBSD/DP as UUT

1 Introduction

The present document contains the Protocol Implementation Conformance Statement (PICS), test cases to ensure conformance of the components of a three-tiered Spectrum Sharing Architecture to the specifications and Requirements defined by Federal Communications Commission (FCC) and Winn Forum.

All terms determined must be defined before completing and submitting this document for final approval.

2 Scope

The present document specifies the measurement procedures for the conformance test of some of the components included in the CBRS Architecture, detailed in Section 5. These procedures contain transmitting characteristics, receiving characteristics and performance requirements as part of the Winn Forum, Spectrum Shared Committee.

Not all components and interfaces in [n.3] are covered by the certification and test cases defined in this document. Development of some of the interfaces and components are out of the scope of Winn Forum, and therefore no test and certification process are provided for them. The scope of Winn Forum test and certification activities includes:

- Enforcing the impact of potentially harmful interference on incumbent Federal DoD systems.
- Conformance of SAS, Domain Proxy and potentially other components and interfaces, whose functionalities are standardized by Winn Forum.
- Enforcing the impact of potentially harmful interference on non-federal incumbents and PAL protection.

The functionalities of RAN or radio device operations and functions are outside the scope of this document.

More generally, tests are only applicable to those components that are intended to support the appropriate functionality. To indicate the circumstances in which tests apply, this is noted in the "*definition and applicability*" part of the test.

This document only covers the test cases required for certification of CBRS entities, and does not include the proprietary tests performed by equipment vendors.

Moreover, this document only covers the test specifications and test cases for the CBRS architecture components, and does not include the test code. The test code is stored in a repository maintained by Winn Forum Working Group 4 [i.1].

3 References

3.1 Normative references

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

[n.1] FCC Report and Order 15-47A1: "Amendment of the Commission’s Rules with Regard to Commercial Operations in the 3550-3650 MHz Band", FCC, April 17 2015, https://apps.fcc.gov/edocs_public/attachmatch/FCC-15-47A1.pdf

[n.2] FCC Report and Order 16-55A1: "Amendment of the Commission’s Rules with Regard to Commercial Operations in the 3550-3650 MHz Band", FCC, May 2 2016, https://apps.fcc.gov/edocs_public/attachmatch/FCC-16-55A1.pdf

[n.3] SSC-Wireless Innovation Forum, WG1&3 Task Group: "SAS Functional Architecture ", Working Document WINNF-15-P-0047 Version V0.3.6, 12 June 2015.

[n.4] SSC-Wireless Innovation Forum, “CBRS Communications Security Technical Specification”, WINNF-TS-0065-V1.1.0

[n.5] SSC-Wireless Innovation Forum, “Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - Citizens Broadband Radio Service Device (CBSD) Interface Technical Specification”, WINNF-TS-0016-V1.1.0

[n.6] SSC-Wireless Innovation Forum, “Requirements for Commercial Operation in the U.S. 3550-3700 MHz Citizens Broadband Radio Service Band”, WINNF-TS-0112-V1.1.0

[n.7] SSC-Wireless Innovation Forum, “WInnForum Recognized CBRS Air Interfaces and Measurements”, WINNF-17-SSC-002-V2.0.1

[n.8] SSC-Wireless Innovation Forum, “WInnForum CBRS Certificate Policy Specification”, WINNF-TS-0022-V1.0.0

3.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] WG4 GitHub Repositories,

- https://github.com/Wireless-Innovation-Forum/Citizens-Broadband-Radio-Service-Device
- https://github.com/Wireless-Innovation-Forum/Spectrum-Access-System
- https://github.com/Wireless-Innovation-Forum/Spectrum-Access-System/tree/master/cert

4 Definitions and Abbreviations

4.1 Abbreviations

CBRS: *Citizens Broadband Radio Services*

CBSD: *Citizens Broadband Radio Service Device*

DOD: *Department of Defense*

DP: *Domain Proxy*

ECC: Elliptic Curve Cryptography (algorithm)

EMS: *Element Management System*

ESC: *Environment Sensing Capability*

FCC: *Federal Communications Commission*

IOT: *Inter-Operability Test*

NTIA: *National Telecommunications and Information Administrations*

RAN: *Radio Access Network*

RSA: Rivest, Shamir, Adleman (cryptography algorithm)

SAS: *Spectrum Access System*

4.2 Definitions

Test Harness: Software hosted in [i.1] that interfaces with a CBSD Under Test and used to perform test cases in this document.

CBSD Under Test: A CBSD to which the sequence of steps listed in the test specifications in this document are applied. Via the Test Harness, the CBSD Under Test exchanges sequences of simulated messages with a simulated SAS according to the test specifications in this document. CBSD Under Test is applied generically within this document to include, where appropriate, the combination of a Domain Proxy and CBSD under test.

5 Test and Certification Process

Figure 1 depicts the approved test and certification process.

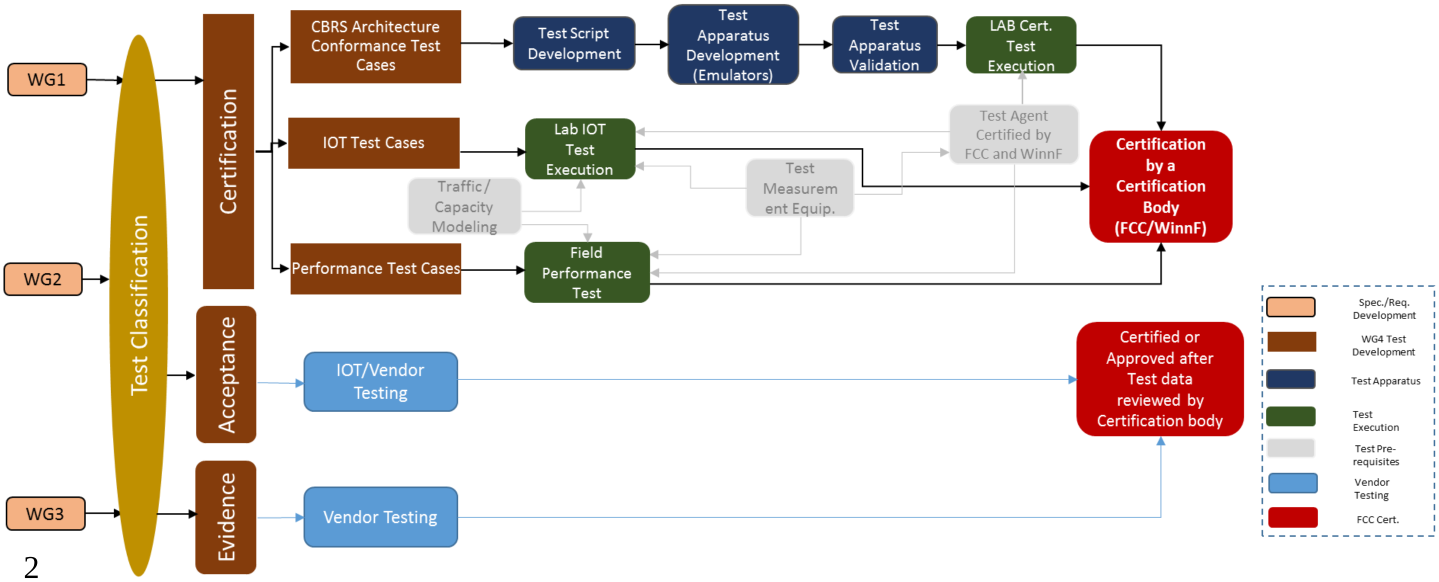


Figure 1: Working Group 4 Test and Certification Process

The requirements, protocols, specifications, and interfaces are defined by SSC-Winn Forum Work Groups 1, 2, and 3. The specifications are derived from FCC, NTIA, and DOD requirements. According to requirements and specifications defined by other work groups, Work Group 4 develops the test cases. The certification test cases can be classified in three classes as follows:

- Functional Test cases
- Interoperability test cases
- Field/Performance test cases

The functional test cases are converted to test scripts to facilitate the development of test apparatus (emulator), which must be validated through a process defined by Winn Forum and FCC. The lab and performance testing require traffic/capacity modeling and measurement equipment. The Interoperability test cases ensure the components developed by different vendors can interconnect each other and provide the functionalities defined by Working Groups 1, 2, and 3.

A test agent is certified by Winn Forum, and ideally FCC/DoD, to perform the certification of ecosystem components. In addition, the test agent publishes the test report

Vendor testing could be either considered as a pre-requisite for certification process, or, by discretion of the certification management entity, they could be partially or fully considered as part of certification plan

Certification is governed either directly by, or through a certification body designated by, the FCC, DOD, and Winn Forum.

5.1 Test Case Classification

In addition to certification specifications and requirements, Winn Forum test cases should include verification that error conditions and fault management protection to support incumbent interference management, conforming to Winn Forum requirements, and meeting required performances performed by vendors prior to or during official certification process. To this end the requirements could be reviewed initially by the certification body and test cases could be classified in three classes: Certification, Acceptance, and Evidence

5.1.1 Class 1: Certification

Testing takes place in an independent, secure and supervised test center or by "Certification partners" where selected requirements are tested and officially approved as having met a standard.

5.1.2 Class 2: Acceptance

Testing conducted to determine if the requirements or specifications (e.g. WG2/WG3 specifications) are met. This testing can be done in lab inter-operability testing similar to how telecommunications equipment is currently tested. This would be focused on black-box system level testing. These tests could be either functional, performance, or inter-operability test cases.

5.1.3 Class 3: Evidence

Material that is presented that furnishes proof of compliance or operation that will satisfy outside regulators that all necessary tests have been executed and passed

5.2 Test ID Definition

Each test case specified in this document has an associated test ID. A test ID shall be defined in the following format.

{*TestRequirement*}.{*TestCategory*}.{*UnitUnderTest*}.{*TestFunction*}.{*SubTestNumber*}

TestRequirement indicates whether a test is to verify if the Unit Under Test meets FCC requirements or Technical Specifications provided by Wireless Innovation Forum. The category of a test, which can be functional, interoperability, or performance, is shown in *TestCategory*. *UnitUnderTest* represents the entity under test, which can be SAS, CBSD, Domain Proxy, ESC or a combination of those entities. *TestFunction* indicates a particular function or requirement a test intends to verify. *SubTestNumber* is an integer larger than 0 to number different test cases in a group of tests performing similar test functions.

In the above Test ID format, the strings in the curly braces are replaced by values in the following tables depending on the characteristics of each test.

Table 5-1 **The values of TestRequirement in Test ID**

Value	Description
FCC	This test is to verify an FCC requirement
WINNF	This test is to verify a Technical Specifications provided by Wireless Innovation Forum

Table 5-2 **The values of TestCategory in Test ID**

Value	Description
FT	This test is a functional test
IT	This test is an interoperability test
PT	This test is a performance test
DT	Database Interface Test

Table 5-3 **The values of UnitUnderTest in Test ID**

Value	Unit under test
S	SAS
C	CBSD
D	Domain Proxy (with CBSD(s))
E	ESC
SC	SAS and CBSD
SD	SAS and Domain Proxy
SS	SAS and SAS

1 Table 5-4 **The values of TestFunction in Test ID**

Value	Description
EXZ	SAS exclusion zone protection
REG	CBSD Registration procedure
SIQ	CBSD Spectrum inquiry procedure
GRA	CBSD Grant procedure
HBT	CBSD Heartbeat procedure
MES	CBSD Measurement report
RLQ	CBSD Grant Relinquishment procedure
DRG	CBSD Deregistration procedure
SCS	SAS-CBSD Security validation
ESP	ESC Protection
INP	ESC Response / DoD Incumbent Protection
FSP	FSS Protection
GWP	GWBL Protection
PPR	PPA Protection
PCR	PPA Creation
FDB	FSS Database
GDB	GWBL Database
FPD	FCC PAL Database
WPD	WINNF PAL Database
PRO	Propagation Model Verification
SSS	SAS-SAS Security, Authentication and Encryption Protocols
ARE	SAS-SAS Administrator Record Exchange

SIR	SAS Implementation record
ESM	ESC Sensor Message
CDT	CBSD Device Type Record
CRD	CBSD Registration Data Message
IPD	Incumbent Protection Data Message
ZRE	Zone Record Exchange
CEM	Coordination Event Message
FAD	Full Activity Dump Message

5.3 Equipment Requirements for this Test Plan

5.3.1 Required Vendor-Supplied Equipment for Test Process

The following equipment is required to perform test cases in this document:

- For stand-alone CBSD, vendor shall supply one CBSD
- For CBSD under control of Domain Proxy, vendor shall supply a Domain Proxy with two CBSD
- Vendor shall supply any ancillary equipment required to ensure CBSD will transmit when it obtains a grant in AUTHORIZED state. This may include additional equipment such as End-User device(s), if required.

5.3.2 Test Equipment Requirements

The following test equipment are required to test the UUT:

- Mock-SAS test harness
- RF measurement equipment: Equipment (such as spectrum analyzer) capable of measuring RF interface of UUT, to determine:
  - Whether UUT is transmitting or not, including ability to measure time when UUT RF transmission starts or ends
  - Whether UUT is transmitting within granted frequency range

  This includes any ancillary RF components (attenuators, cables, couplers, etc.) which may be required to perform those measurements.

The following are outside the scope of this document:

- Choice of specific RF test equipment, and exact configuration or operation of that equipment
- Specific test setup to allow monitoring of the UUT by the RF measurement equipment

5.3.3 UUT Test Interface Requirements

The unit-under-test shall provide functionality via a vendor-defined test interface, in order to support completion of all test cases in this document. The interface is outside the scope of this test document, but shall provide a minimum set of functionalities as described below:

1. Method to return CBSD to an unregistered state after the conclusion of a test case. Test cases conclude with the CBSD in various states, such as Authorized grant state and actively transmitting, or may end with the test harness providing an error condition (such as a non-zero error code, or no response). Therefore, it is necessary for the test interface to provide a method to return the CBSD to an unregistered state, prior to start of the next test case.
2. Method to trigger CBSD to perform the following sequence in its entirety, automatically, starting from an unregistered state, and provided each step results in a successful Response message (response code = 0):
    a. Register with SAS
    b. Perform Spectrum Inquiry Request (optional, if UUT supports this message)
    c. Perform Grant Request, where UUT requests a pre-defined frequency range of operation, as required by the particular test case
    d. Heartbeat the CBSD for the *grantId* obtained
    e. Begin transmission within the Granted frequency range
3. Method to trigger CBSD to perform a Spectrum Relinquishment, if it has a valid grant in AUTHORIZED or GRANTED state.
4. Method to trigger CBSD to Deregister, if it is in the registered state.

    Method to load test certificates into UUT, as required, for use in authentication procedures with the test harness.
5. If provided by the CBSD, access to the CBSD CPI interface for entering of CPI-related registration information into the CBSD management interface.

5.4 Validity of CBSD Messages

The following table (taken from [n.5]), shows the valid and invalid messages for a given CBSD state. This table will be used to test the compliance of CBSD with respect to state transition.

Table 5-**5 Validity of CBSD Messages for Different States in SAS-CBSD Protocol**

	CBSD Message Validity					
State	R1 (Registration)	G1 (Grant Request)	H1 (Heartbeat)	L1 (Relinquish)	D1 Deregister	Spectrum Inquiry
Unknown	Yes	No	No	No	No	No

Registered	No	Yes	No	No	Yes	Yes
Registration Pending	Yes	No	No	No	Yes	No
Granted	No	Yes	Yes	Yes	No	Yes
Transmission	No	Yes	Yes	Yes	No	Yes
Decommissioned	Yes	No	No	No	No	No

6 SAS-CBSD/DP Interface Conformance Test Specifications

This section includes all test cases required for ensuring the SAS-User interface conforms by the specifications defined by Winn Forum and directed by the requirements established by the FCC and DOD. The table shown in this section identifies and categorizes the test cases for conformance testing.

Table 5-6 Look Up table for Test Case List

C1	Applies to any CBSD that allows multi-step registration
C2	Applies to any CATA CBSD that determines its own location
C3	CBSDs that report CPI signed data
C4	Applies to CBSD that reports received power without grant
C5	Applies to CBSD that reports received power with grant
C	Conditional[Mandatory if CBSD supports relavant functionality]
O	Optional
M	Mandatory

Table 5-7 Test Case List

00122 Section	CAT A	CAT A: CPI	CAT B	DP	Required for Certification	Test Case ID	Test Case Title
6.1.4.1.1	X	X	X		C1*	WINNF.FT.C.REG.1	Multi-Step registration for CBSD
6.1.4.1.2	X	X	X	X	C1*	WINNF.FT.D.REG.2	Multi-Step registration for DP/CBSD
6.1.4.1.3	X				C2*	WINNF.FT.C.REG.3	Single-Step registration for Category A CBSD
6.1.4.1.4	X			X	C2*	WINNF.FT.D.REG.4	Single-Step registration for DP/ CBSD Cat A
6.1.4.1.5		X	X		C3*	WINNF.FT.C.REG.5	Single-Step registration for CBSD with CPI signed data
6.1.4.1.6		X	X	X	C3*	WINNF.FT.D.REG.6	Single-Step registration for DP/CBSD with CPI signed data
6.1.4.1.7	X	X	X	X	O	WINNF.FT.C.REG.7	Registration includes CBSD Group identifier
6.1.4.1.8	X	X	X	X	M	WINNF.FT.C.REG.8	Registration due to change of an installation parameter
6.1.4.2.1	X	X	X		M	WINNF.FT.C.REG.9	Missing Required parameters (responseCode 102)
6.1.4.2.2	X	X	X	X	M	WINNF.FT.D.REG.10	Missing Required parameters in DP/CBSD request (responseCode 102)
6.1.4.2.3	X	X	X		M	WINNF.FT.C.REG.11	Pending registration (responseCode 200)
6.1.4.2.4	X	X	X	X	M	WINNF.FT.D.REG.12	Pending registration in DP/CBSD request (responseCode 200)
6.1.4.2.5	X	X	X		M	WINNF.FT.C.REG.13	Invalid parameter (responseCode 103)

6.1.4.2.6	X	X	X	X	M	WINNF.FT.D.REG.14	Invalid parameters in DP/CBSD request (responseCode 103)
6.1.4.2.7	X	X	X		M	WINNF.FT.C.REG.15	Blacklisted CBSD (responseCode 101)
6.1.4.2.8	X	X	X	X	M	WINNF.FT.D.REG.16	Blacklisted CBSD in DP/CBSD request (responseCode 101)
6.1.4.2.9	X	X	X		M	WINNF.FT.C.REG.17	Unsupported SAS protocol version (responseCode 100)
6.1.4.2.10	X	X	X	X	M	WINNF.FT.D.REG.18	Unsupported SAS protocol version in DP/CBSD request (responseCode 100)
6.1.4.2.11	X	X	X		O	WINNF.FT.C.REG.19	Group Error (responseCode 201)
6.1.4.2.12	X	X	X	X	O	WINNF.FT.D.REG.20	Group Error in DP/CBSD request (responseCode 201)
6.1.4.3.1	X			X	M	WINNF.FT.C.REG.21	Category A device has moved greater than 50m horizontally
6.1.4.3.2	X			X	M	WINNF.FT.C.REG.22	Category A device has moved greater than 3m vertically
6.2.4.1.1	X	X	X		O	WINNF.FT.C.SIQ.1	Spectrum Inquiry (First time after registration) with Conditional
6.2.4.1.2	X	X	X	X	O	WINNF.FT.D.SIQ.2	Spectrum Inquiry (First time after registration) with Conditional Error parameters for all requests – Domain Proxy
6.3.4.1.1	X	X	X		M	WINNF.FT.C.GRA.1	Grant (First time after registration) with Conditional parameters
6.3.4.1.2	X	X	X	X	M	WINNF.FT.D.GRA.2	Grant (First time after registration) with Conditional parameters for all

							requests – Domain Proxy
6.3.4.2.1	X	X	X	X	M	WINNF.FT.D.GRA.3	Unsuccessful Grant response for requests with various responseCodes – Domain Proxy
6.4.4.1.1	X	X	X		M	WINNF.FT.C.HBT.1	Heartbeat Success Case (first Heartbeat Response)
6.4.4.1.2	X	X	X	X	M	WINNF.FT.D.HBT.2	Domain Proxy Heartbeat Success Case (first Heartbeat Response)
6.4.4.2.1	X	X	X	X	M	WINNF.FT.C.HBT.3	Heartbeat responseCode=105 (DEREGISTER)
6.4.4.2.2	X	X	X		M	WINNF.FT.C.HBT.4	Heartbeat responseCode=500 (TERMINATED_GRANT)
6.4.4.2.3	X	X	X	X	M	WINNF.FT.C.HBT.5	Heartbeat responseCode=501 (SUSPENDED_GRANT)
6.4.4.2.4	X	X	X	X	M	WINNF.FT.C.HBT.6	Heartbeat responseCode=502 (UNSYNC_OP_PARAM)
6.4.4.2.5	X	X	X	X	M	WINNF.FT.D.HBT.7	Domain Proxy Heartbeat responseCode=500
6.4.4.3.1	X	X	X	X	M	WINNF.FT.C.HBT.8	Heartbeat Response Absent (First Heartbeat)
6.4.4.3.2	X	X	X	X	M	WINNF.FT.C.HBT.9	Heartbeat Response Absent (Subsequent Heartbeat)
6.4.4.4.1	X	X	X	X	O	WINNF.FT.C.HBT.10	Successful Grant Renewal in Heartbeat Test Case
6.4.4.4.2	X	X	X	X	M	WINNF.FT.C.HBT.11	Grant Expiry in Heartbeat Test Case
6.5.4.1.1	X	X	X	X	M	WINNF.FT.C.MES.1	Measurement Capability in

							Registration Request Message
6.5.4.2.1	X	X	X		C4*	WINNF.FT.C.MES.2	Registration Response contains measReportConfig
6.5.4.2.2	X	X	X	X	C4*	WINNF.FT.D.MES.3	Domain Proxy: Registration Response contains measReportConfig
6.5.4.2.3	X	X	X	X	C5*	WINNF.FT.C.MES.4	Grant Response contains measReportConfig
6.5.4.2.4	X	X	X		C5*	WINNF.FT.C.MES.5	Heartbeat Response contains measReportConfig
6.5.4.2.5	X	X	X	X	C5*	WINNF.FT.D.MES.6	Domain Proxy: Heartbeat Response contains measReportConfig
6.6.4.1.1	X	X	X	X	M	WINNF.FT.C.RLQ.1	Successful relinquishment response for CBSD in Granted state
6.6.4.2.1	X	X	X	X	M	WINNF.FT.C.RLQ.2	Successful relinquishment response for CBSD in Authorized state
6.6.4.3.1	X	X	X	X	O	WINNF.FT.C.RLQ.3	Unsuccessful relinquishment response due to missing CBSD ID parameter
6.6.4.3.2	X	X	X	X	O	WINNF.FT.C.RLQ.4	Unsuccessful relinquishment response due to missing Grant ID parameter
6.6.4.4.1	X	X	X	X	O	WINNF.FT.C.RLQ.5	Unsuccessful relinquishment response due to invalid CBSD ID value
6.6.4.4.2	X	X	X	X	O	WINNF.FT.C.RLQ.6	Unsuccessful relinquishment response due to

							invalid Grant ID parameter
6.7.4.1.1	X	X	X		M	WINNF.FT.C.DRG.1	Valid and correct cbsdId
6.7.4.1.2	X	X	X	X	M	WINNF.FT.C.DRG.2	Valid and correct cbsdId: Domain Proxy serving two CBSDs
6.7.4.2.1	X	X	X		O	WINNF.FT.C.DRG.3	Missing cbsdId: single object in the DeregistrationRequest
6.7.4.2.2	X	X	X	X	O	WINNF.FT.C.DRG.4	Missing cbsdId: Domain Proxy serving two CBSDs
6.7.4.3.1	X	X	X	X	O	WINNF.FT.C.DRG.5	cbsdId Does Not Exist in the SAS: single request object
6.8.4.1.1	X	X	X		M	WINNF.FT.C.SCS.1	Successful TLS connection between CBSD and mock-SAS
6.8.4.1.2	X	X	X		M	WINNF.FT.C.SCS.2	Successful TLS connection between CBSD and mock-SAS (No critical extension in certificate)
6.8.4.1.3	X	X	X	X	M	WINNF.FT.D.SCS.3	Successful TLS connection between Domain-proxy and mock-SAS – Domain Proxy
6.8.4.1.4	X	X	X	X	M	WINNF.FT.D.SCS.4	Successful TLS connection between DomainProxy and mock-SAS (No critical extension in certificate) – Domain Proxy
6.8.4.2.1	X	X	X		O	WINNF.FT.C.SCS.5	TLS failure due to blacklist certificate
6.8.4.2.2	X	X	X		O	WINNF.FT.C.SCS.6	TLS failure due to incorrect critical extension in server certificate

6.8.4.2.3	X	X	X		O	WINNF.FT.C.SCS.7	TLS failure due to expired server certificate
6.8.4.2.4	X	X	X		O	WINNF.FT.C.SCS.8	TLS failure when Critical extension in server certificate contains inapplicable fields
6.8.4.2.5	X	X	X		O	WINNF.FT.C.SCS.9	TLS failure when mock-SAS certificate is issue by unknown CA
6.8.4.2.6	X	X	X	X	O	WINNF.FT.D.SCS.10	TLS failure due to blacklist certificate – Domain Proxy
6.8.4.2.7	X	X	X	X	O	WINNF.FT.D.SCS.11	TLS failure due to incorrect critical extension in server certificate – Domain Proxy
6.8.4.2.8	X	X	X	X	O	WINNF.FT.D.SCS.12	TLS failure due to expired server certificate – Domain Proxy
6.8.4.2.9	X	X	X	X	O	WINNF.FT.D.SCS.13	TLS failure when Critical extension in server certificate contains inapplicable fields – Domain Proxy
6.8.4.2.10	X	X	X	X	O	WINNF.FT.D.SCS.14	TLS failure when mock-SAS certificate is issue by unknown CA – Domain Proxy

6.1 CBSD Registration Process

6.1.1 Definition and applicability and Scope of Test Case

This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Registration Procedure. It assumes as a precondition the CBSD has successfully discovered the SAS it wants to register with.

The main approach is for each test to generate a CBSD registration request and to validate the CBSD takes the appropriate action following the SAS registration response covering all the defined responseCodes available that pertain to the CBSD registration process in [n.5]. This includes successful registration as well, which is signified by responseCode 0.

6.1.2 Test Characteristics

Table 6-1 CBSD Registration Process Test Characteristics

1	**Test ID**	WINNF.FT.C.REG
2	**Title**	CBSD Registration Process
3	**Working Group / Entity**	WG3
4	**Test Type**	Functionality
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification**	[n.5], sections 8.3, 10.1, and 10.2

6.1.3 Method of test

6.1.3.1 Initial Conditions / Test Pre-conditions

- The pre-conditions of the test case are:
    - CBSD has gone through SAS discovery process and can authenticate with the SAS. The exact condition of the CBSD after the discovery process are detailed in each test case.

- The applicable structure of the RegistrationRequest parameter and RegistrationRequest object are defined in [n.5].

In summary, the CBSD parameters used for the registration process are categorized into the following [n.5]:

- Required
- Conditional
- Optional

Two cases are considered:

- When the location and other installation parameters are already uploaded in the CBSD, and could be included in the registration message.
- When the location and other installation parameters are uploaded offline by a professional installer.

Note: In this Section, "Multi-Step Registration” refers to the Registration process where the REG-conditional parameters are not included in the "RegistrationRequest Object". “Single-Step

Registration” refers to the Registration process where the conditional parameters are included in the "RegistrationRequest Object". A CBSD vendor may support one or more of these registration methods. Test cases apply according to the type of registration process(es) supported by the CBSD under test.

6.1.4 Test Procedure

6.1.4.1 Successful registration (responseCode 0)

Upon a successful response from the SAS (responseCode = 0), the CBSD will generate its next message to the SAS. This can be a SpectrumInquiry or Grant request. Since these test cases are only validating the Registration period, these subsequent requests will not be handled by the SAS test harness.

6.1.4.1.1 [WINNF.FT.C.REG.1] Multi-Step registration for CBSD

This test validates that each of the required parameters appear within the registration request message. The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	CBSD sends correct Registration request to SAS Test Harness: valid userId, fccId, and cbsdSerialNumber Note: It is outside the scope of this document to test the Registration information that is supplied via another means	PASS	FAIL
2	• The registration parameters sent from CBSD are received and conform to proper format and acceptable ranges. • SAS test harness sends a CBSD Registration Response as follows: – SAS response includes a valid cbsdId. – measReportConfig shall not be included – The responseCode in Response Object shall be 0	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 0 and cbsdId. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.1.2 [WINNF.FT.D.REG.2] Multi-Step registration for DP/CBSD

This test validates that each of the required parameters appear within the registration request message. This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps:

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	DP with two CBSD sends correct Registration requests in the form of one 2-element Array or as individual messages to SAS Test Harness: valid userId, fccId, and cbsdSerialNumber. Note: It is outside the scope of this document to test the Registration information that is supplied via another means.	PASS	FAIL
2	• The registration parameters sent for each CBSD are received and conform to proper format and acceptable ranges. • SAS test harness sends a CBSD Registration Response in the form of one 2-element Array or individual messages as follows: – SAS response includes a valid cbsdId for each CBSD. – measReportConfig shall not be included – The responseCode in Response Object shall be 0 for each CBSD	-	-
3	Each CBSD receives CBSD Registration Response and obtains the responseCode = 0 and cbsdId. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.1.3 [WINNF.FT.C.REG.3] Single-Step registration for Category A CBSD

This test validates that each of the required and REG-Conditional parameters appear within the registration request message.

For a Category A CBSD which determine own location, the test lab and vendor must agree on the lab setup under which device can determine its own location successfully.

For a Category A CBSD which does not determine its own location, refer to the test case for the CPI signed data.

The following are the test execution steps:

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state • Coordinates (latitude and longitude) and antenna height are obtained, it is beyond the scope of this case to specify the method		

1	CBSD sends proper Registration request to SAS Test Harness: all required and REG-Conditional parameter included (userId, fccId, cbsdSerialNumber, cbsdCategory, airInterface, installationParam, measCapability) for a Category A CBSD. The InstallationParam object shall include the latitude, longitude, height and heightType. Verify all of the required and REG-conditional parameters are provided Verify the latitude, longitude, and elevation are within +-50m horizontally and +/-3m vertically	PASS	FAIL
2	• The registration parameters sent from CBSD are received and conform to proper format and acceptable ranges. • SAS test harness sends a CBSD Registration Response as follows: – SAS response includes a valid cbsdId. – measReportConfig shall not be included. – The responseCode in Response Object shall be 0	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 0 and cbsdId. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.1.4 [WINNF.FT.D.REG.4] Single-Step registration for DP/ CBSD Cat A

This test validates that each of the required and REG-Conditional parameters appear within the registration request message. This test case applies to Domain Proxy supervising two CBSDs.

For a Category A CBSD which determine own location, the test lab and vendor must agree on the lab setup under which device can determine its own location successfully.

For a Category A CBSD which does not determine its own location, refer to the test case for the CPI signed data.

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state • Coordinates (latitude and longitude) and antenna height are obtained for each CBSD, it is beyond the scope of this case to specify the method		
1	The DP with two CBSDs sends proper Registration requests in the form of one 2-element Array or as individual messages to SAS Test	PASS	FAIL

	Harness: all required and REG-Conditional parameter included (userId, fccId, cbsdSerialNumber, cbsdCategory, airInterface, installationParam, measCapability) for Category A CBSDs. The InstallationParam object shall include the latitude, longitude, height and heightType. Verify all of the required and REG-conditional parameters are provided Verify the latitude, longitude, and elevation are within +-50m horizontally and +/-3m vertically		
2	• The registration parameters sent for each CBSD within the array are received and conform to proper format and acceptable ranges. • SAS test harness sends a CBSD Registration Response in the form of one 2-element Array or individual messages as follows: – SAS response includes a valid cbsdId for each CBSD. – measReportConfig for each CBSD shall not be included. – The responseCode in Response Object shall be 0 for each CBSD	-	-
3	Each CBSD receives CBSD Registration Response and obtains the responseCode = 0 and cbsdId. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.1.5 [WINNF.FT.C.REG.5] Single-Step registration for CBSD with CPI signed data

This test validates that each of the required and REG-Conditional parameters appear within the registration request message.

All Category B devices, and Category A devices not able to determine its own location require installation by a CPI. This test is for devices where the CPI enters data into the CBSD and this information along with the CPI signature are sent in the request message. Devices which require the CPI to enter the information into a SAS interface would follow the multiple step registration test.

The following are the test execution steps:

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state • Coordinates (latitude and longitude) and antenna height are obtained, it is beyond the scope of this case to specify the method		

1	CBSD sends proper Registration request to SAS Test Harness: all required and REG-Conditional parameter included (userId, fccId, cbsdSerialNumber, cbsdCategory, airInterface, installationParam, measCapability, cpiSignatureData) The InstallationParam object shall include the latitude, longitude, height and heightType. Verify all of the required and REG-conditional parameters are provided Verify the latitude, longitude, and elevation are within +-50m horizontally and +/-3m vertically	PASS	FAIL
2	• The registration parameters sent from CBSD are received and conform to proper format and acceptable ranges. • SAS test harness sends a CBSD Registration Response as follows: – SAS response includes a valid cbsdId. – measReportConfig shall not be included. – The responseCode in Response Object shall be 0	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 0 and cbsdId. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.1.6 [WINNF.FT.D.REG.6] Single-Step registration for DP/CBSD with CPI signed data

This test validates that each of the required and REG-Conditional parameters appear within the registration request message. This test case applies to Domain Proxy supervising two CBSDs.

All Category B devices, and Category A devices not able to determine its own location require installation by a CPI. This test is for devices where the CPI enters data into the CBSD and this information along with the CPI signature are sent in the request message. Devices which require the CPI to enter the information into a SAS interface would follow the multiple step registration test.

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state • Coordinates (latitude and longitude) and antenna height are obtained for each CBSD, it is beyond the scope of this case to specify the method		
1	The DP with two CBSDs sends correct Registration requests in the form of one 2-element Array or as individual messages to SAS Test	PASS	FAIL

	Harness: all required and REG-Conditional parameter included (userId, fccId, cbsdSerialNumber, cbsdCategory, airInterface, installationParam, measCapability, cpiSignatureData). The InstallationParam object shall include the latitude, longitude, height and heightType. Verify all of the required and REG-conditional parameters are provided Verify the latitude, longitude, and elevation are within +-50m horizontally and +/-3m vertically		
2	• The registration parameters sent from each CBSD in the array are received and conform to their format and acceptable ranges. • SAS test harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: – SAS response includes a valid cbsdId for each CBSD. – measReportConfig for each CBSD shall not be included. – The responseCode in Response Object shall be 0 for each CBSD	-	-
3	Each CBSD receives CBSD Registration Response and obtains the responseCode = 0 and cbsdId. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.1.7 [WINNFF.FT.C.REG.7] Registration includes CBSD Group identifier

The *groupingParam* is an optional parameter, if present it designates the CBSD as a member of a particular group of CBSDs.

The following are the test execution steps to validate the GroupParam object is sent to the SAS:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	UUT has successfully registered with SAS test harness	-	-
2	Using [WINNF.FT.C.REG.1] registration test procedure as the basis, include the groupingParam information in the request	-	-

3	UUT sends correct Registration request message to SAS Test Harness: All of the required parameters, plus the GroupParam object shall be sent to the SAS test harness.
	The GroupParam object shall contain:
	groupType = INTERFERENCE_COORDINATION
	groupId = userId.
		PASS	FAIL

6.1.4.1.8 [WINNFF.FT.C.REG.8] Registration due to change of an installation parameter

The purpose of this test is to verify that the CBSD sends notification to the SAS when an installation parameter has been changed.

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	UUT has successfully registered with SAS test harness	-	-
2	Change an installation parameters for the UUT (time 0)	-	-
3a	Option 1: UUT sends a Registration request to SAS Test Harness with the new value for the changed installation parameter. The registration request shall be sent time < 60 seconds from when the change was applied.	PASS	FAIL
3b	Option 2: UUT sends a deregistrationRequest prior to sending the registration request with the complete registration information (with the updated value) The deregistration request shall be sent time < 60 seconds from when the change was applied.	PASS	FAIL

6.1.4.2 Unsuccessful registration: non-zero responseCodes

CBSD under test cannot be expected to generate a message with a missing or invalid parameter. To test for responseCode not equal to 0, the SAS test harness will respond to a registrationRequest message with a registrationResponse with a non-zero responseCode.

The purpose of these tests is to ensure that the CBSD does not proceed with a SpectrumInquiry or Grant request message, and does not transmit when a responseCode other than 0 is received.

Commercial versions of CBSDs may not generate registrationRequests with an invalid parameter/value. It is left to SAS test cases to test for missing and invalid parameters.

6.1.4.2.1 [WINNF.FT.C.REG.9] Missing Required parameters (responseCode 102)

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	CBSD sends a Registration request to SAS Test Harness.	-	-
2	SAS Test Harness rejects the request by sending a CBSD Registration Response as follows:	-	-

	– SAS response does not include cbsdId – The responseCode in Response Object shall be 102.		
3	CBSD receives CBSD Registration Response and obtains the responseCode = 102. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.2 [WINNF.FT.D.REG.10] Missing Required parameters in DP/CBSD request (responseCode 102)

This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	The DP with two CBSDs sends a Registration request in the form of one 2-element Array or as individual messages to SAS Test Harness.	-	-
2	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: – SAS response does not include a cbsdId. – The responseCode in Response Object shall be 102 for CBSD1 and CBSD2	-	-
3	• CBSD1 and CBSD2 receive CBSD Registration Response and obtains the responseCode = 102. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.3 [WINNF.FT.C.REG.11] Pending registration (responseCode 200)

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	CBSD sends a Registration request to SAS Test Harness.	-	-

2	SAS Test Harness rejects the request by sending a CBSD Registration Response as follows: – SAS response does not include cbsdId – The responseCode in Response Object shall be 200.	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 200. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.4 [WINNF.FT.D.REG.12] Pending registration in DP/CBSD request (responseCode 200)

This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	The DP with two CBSDs sends a Registration request in the form of one 2-element Array or as individual messages to SAS Test Harness	-	-
2	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: – SAS response does not include a valid cbsdId. – SAS response does not include cbsdId for either CBSD – The responseCode shall be 200 for both CBSDs.	-	-
3	• Each CBSD receives the CBSD Registration Response with responseCode = 200. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.5 [WINNF.FT.C.REG.13] Invalid parameter (responseCode 103)

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		

1	CBSD sends a Registration request to SAS Test Harness.	-	-
2	SAS Test Harness rejects the request by sending a CBSD Registration Response as follows: – SAS response does not include cbsdId – The responseCode in Response Object shall be 103.	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 103. Monitor the RF output of UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

Note: Same test can be executed for a Cat A or Cat B CBSD.

6.1.4.2.6 [WINNF.FT.D.REG.14] Invalid parameters in DP/CBSD request (responseCode 103)

This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	The DP with two CBSDs sends a Single-Step Registration request in the form of one 2-element Array or as individual messages to SAS Test Harness - .	-	-
2	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: – SAS response includes a valid cbsdId for the first CBSD. – measReportConfig for each CBSD shall not be included. – The responseCode in Response Object shall be 0 for the first CBSD – The responseCode in Response Object shall be 103 for the second CBSDs	-	-
3	• The first CBSD receives the Registration Response and obtains the responseCode = 0 and cbsdId. • CBSD two receives CBSD Registration Response and obtains the responseCode = 103. A responseMessage may be included providing description of the result. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.7 [WINNF.FT.C.REG.15] Blacklisted CBSD (responseCode 101)

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	CBSD sends correct Registration request to SAS Test Harness.	-	-
2	SAS Test Harness rejects the request by sending a CBSD Registration Response as follows: – SAS response does not include CBSD ID – The responseCode in Response Object shall be 101.	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 101. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.8 [WINNF.FT.D.REG.16] Blacklisted CBSD in DP/CBSD request (responseCode 101)

This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	Two CBSDs send correct Single-step Registration requests in the form of one 2-element Array or as individual messages to SAS Test Harness	-	-
2	Assume the second CBSD is blacklisted, SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: – SAS response includes a valid cbsdId for the first CBSD. – The responseCode in Response Object shall be 0 for the first CBSD – The responseCode in Response Object shall be 101 for the second CBSD	-	-
3	• The first CBSD receives the Registration Response and obtains the responseCode = 0 and cbsdId. • CBSD two receives CBSD Registration Response and obtains the responseCode = 101.	PASS	FAIL

	Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • • UUT shall not transmit RF		

6.1.4.2.9 [WINNF.FT.C.REG.17] Unsupported SAS protocol version (responseCode 100)

The following are the test execution steps:

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	CBSD sends a Registration request to SAS Test Harness.	-	-
2	SAS Test Harness rejects the request by sending a CBSD Registration Response as follows: – SAS response does not include cbsdId – The responseCode in Response Object shall be 100. Alternatively, the SAS can return HTTP status code 404	-	-
	CBSD receives CBSD Registration Response and obtains the responseCode = 100 or HTTP status code 404. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.10 [WINNF.FT.D.REG.18] Unsupported SAS protocol version in DP/CBSD request (responseCode 100)

This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps:

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	Two CBSD send a Registration request in the form of one 2-element Array or as individual messages to SAS Test Harness.	-	-
2	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages to the CBSD test Harness as follows: – SAS response does not include cbsdId	-	-

	– The responseCode in Response Object should be 100 for each CBSD. Alternatively, the SAS can return HTTP status code 404.		
3	CBSD receives CBSD Registration Response and obtains the responseCode = 100 or HTTP status code 404. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

6.1.4.2.11 [WINNF.FT.C.REG.19] Group Error (responseCode 201)

The registrationRequest groupingParam is an optional field. If this optional parameter is provided, this test will validate that the CBSD will remain Unregistered after receiving responseCode 201.

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	CBSD sends a Registration request to SAS Test Harness. • CBSD sends proper Registration request to SAS Test Harness: all required and parameters included • CBSD request includes the optional groupingParam object containing groupType and groupId .	-	-
2	SAS Test Harness rejects the request by sending a CBSD Registration Response as follows: – SAS response does not include cbsdId – The responseCode shall be 201	-	-
3	CBSD receives CBSD Registration Response and obtains the responseCode = 201. Monitor the RF output of the UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • UUT shall not transmit RF	PASS	FAIL

Note: SAS implementation may use responseCode 103 or 201.

6.1.4.2.12 [WINNF.FT.D.REG.20] Group Error in DP/CBSD request (responseCode 201)

The registrationRequest groupingParam is an optional field. If this optional parameter is provided, this test will validate that the CBSD will remain Unregistered after receiving responseCode 201.

This test case applies to Domain Proxy supervising two CBSDs. The following steps describes the test execution steps

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	Two CBSDs send a Registration request in the form of one 2-element Array or as individual messages to SAS Test. The second CBSD includes the optional groupingParam object containing groupType and groupId.	-	-
2	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages to the CBSD as follows: – SAS response includes a valid cbsdId for the first CBSD. – measReportConfig for each CBSD shall not be included. – The responseCode in Response Object shall be 0 for the first CBSD – The responseCode in Response Object shall be 201 for the second CBSD.	-	-
3	• The first CBSD receives the Registration Response and obtain the responseCode = 0 and cbsdId. • The second CBSD receives CBSD Registration Response and obtains the responseCode = 201. Monitor the RF output of each UUT from start of test until 60 seconds after Step 2 is complete. This is the end of the test. Verify: • • UUT shall not transmit RF	PASS	FAIL

6.1.4.3 Category A CBSD location update

This section is specific to Category A CBSDs that do not require professional installation. The requirement is for the Category A (non-professionally installed) to report to the SAS any location change exceeding a distance of 50m horizontally or 3m vertically within a 60 second window. The essence of the test requirement is listed below, but is left to the CBSD vendor and certification lab to generate the required evidence showing the UUT meets the requirement.

6.1.4.3.1 WINNFF.FT.C.REG.21] Category A device has moved greater than 50m horizontally

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	UUT has successfully registered with SAS test harness	-	-
2	Move the Category A UUT a horizontal distance of 50m (time 0 at new location)	-	-
3	UUT sends a Registration request to SAS Test Harness with the new coordinates. The registration request shall be sent time < 60 seconds from when the UUT is placed at the new location.	PASS	FAIL

6.1.4.3.2 WINNFF.FT.C.REG.22] Category A device has moved greater than 3m vertically

The following are the test execution steps:

#	Test Execution Steps	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT is in the Unregistered state		
1	UUT has successfully registered with SAS test harness	-	-
2	Move the Category A UUT a vertical distance of 3m (time 0 at new location)	-	-
3	UUT sends a Registration request to SAS Test Harness with the new height. The registration request shall be sent time < 60 seconds from when the UUT is placed at the new location.	PASS	FAIL

6.2 CBSD Spectrum Inquiry Process

6.2.1 Definition and applicability and Scope of Test Case

This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Spectrum Inquiry Procedure. It assumes as a precondition the CBSD has successfully discovered the SAS it wants to communicate with.

The main approach is for each test to generate a CBSD spectrum inquiry request and to validate the CBSD takes the appropriate action following the SAS spectrum inquiry response.

6.2.2 Test Characteristics

Table 6-2 CBSD Spectrum Inquiry Process Test Characteristics

1	**Test ID**	WINNF.FT.C.SIQ
2	**Title**	CBSD Spectrum Inquiry Process
3	**Working Group / Entity**	WG4
4	**Test Type**	Functionality
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification**	[n.5], sections 8.4, 10.3, and 10.4

6.2.3 Method of test

6.2.3.1 Initial Conditions / Test Pre-conditions

- The pre-conditions of the test case are:
  - CBSD has gone through SAS discovery process and can authenticate with the SAS. The exact condition of the CBSD after the discovery process are detailed in each test case.
  - The CBSD has already registered with SAS and has obtained a valid CBSD ID.

- The applicable structure of the SpectrumInquiryRequest parameter and SpectrumInquiryResponse objects are defined in [n.5].

In summary, the CBSD parameters used for the process are categorized into the following [n.5]:

- Required
- Conditional
- Optional

6.2.4 Test Procedure

6.2.4.1 Successful response from mock-SAS server

In all test cases under this category, the SpectrumInquiry Request is correct and valid. That is, the mock-SAS harness has checked that all the "Required" parameters are present in the SpectrumInquiry Request and their values are valid and correct. This part makes sure that CBSD is able to send correct and valid SpectrumInquiry Request.

6.2.4.1.1 [WINNF.FT.C.SIQ.1] Successful spectrum Inquiry response from SAS

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	• Configure the Mock-SAS to send the successful spectrum inquiry response. • Make sure that the registration procedure has been successfully completed with the Mock-SAS • Make sure CBSD ID (C) exists in mock-SAS.	PASS	FAIL
2	Configure CBSD to send valid and correct Spectrum Inquiry with list of frequency ranges. • Make sure the list of frequencyRange objects sent by CBSD are within CBRS range. • SAS test harness is configured to send responseCode=0.	PASS	FAIL
3	SAS Harness sends a spectrumInquiry Response with • CBSD ID = C. • List of available channels with all required parameters • responseCode = 0	--	--
4	Note: As per the CBSD implementation, CBSD may go for Grant procedure followed by heartbeat procedure at this stage. If applicable, Wait for Grant Request and HeartBeat Request from the CBSD. Details for the Grant and Heartbeat process can be referred from applicable sections.	--	--
5	Verify using a spectrum analyzer that CBSD is not transmitting unless an authorized grant is available.	PASS	FAIL

6.2.4.1.2 [WINNF.FT.D.SIQ.2] Successful spectrum Inquiry response from SAS for all requests – Domain Proxy

The following steps describe the test execution.

#	Test Execution Steps	**Results**	
1	• Configure the Mock-SAS to send successful spectrum inquiry response for all the requests sent by domain proxy • Make sure that the registration procedure has been successfully completed with the Mock-SAS for all the CBSD. • Make sure CBSD ID (C1..CN) exists in mock-SAS for all the CBSD.	PASS	FAIL
2	Configure Domain-Proxy to send valid and correct Spectrum Inquiry Request containing data for multiple CBSD with list of frequency ranges. • Make sure the list of frequencyRange objects sent by Domain-Proxy are within CBRS range. • SAS test harness is configured to send responseCode=0 for all the requests.	PASS	FAIL
3	SAS Harness sends a spectrumInquiry Response for all requests with • CBSD ID = Ci. • List of available channels with all required parameters • responseCode = 0	--	--

4	As per the CBSD/DP implementation, DP may go for Grant procedure followed by heartbeat procedure. If applicable, Wait for Grant Request and HeartBeat Request from the Domain-Proxy. Details for the Grant and Heartbeat process can be referred from applicable sections.	--	--
5	Verify using a spectrum analyzer that none of the CBSD is transmitting unless an authorized grant is available.	PASS	FAIL

6.3 CBSD Spectrum Grant Process

6.3.1 Definition and applicability and Scope of Test Case

This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Spectrum Grant Procedure. It assumes as a precondition the CBSD has successfully discovered the SAS it wants to communicate with.

The main approach is for each test to generate a CBSD spectrum grant request and to validate the CBSD takes the appropriate action following the SAS spectrum grant response.

6.3.2 Test Characteristics

Table 6-3 CBSD Spectrum Grant Process Test Characteristics

1	**Test ID**	WINNF.FT.C.GRA
2	**Title**	CBSD Spectrum Grant Process
3	**Working Group / Entity**	WG4
4	**Test Type**	Functionality
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification**	[n.5], sections 8.5, 10.5, and 10.6

6.3.3 Method of test

6.3.3.1 Initial Conditions / Test Pre-conditions

- The pre-conditions of the test case are:
  - CBSD has gone through SAS discovery process and can authenticate with the SAS. The exact condition of the CBSD after the discovery process are detailed in each test case.
  - The CBSD has already registered with SAS and has obtained a valid CBSD ID.

- The applicable structure of the GrantRequest parameter and GrantResponse objects are defined in [n.5].

In summary, the CBSD parameters used for the process are categorized into the following [n.5]:

- Required
- Conditional
- Optional

6.3.4 Test Procedure

6.3.4.1 Successful responses from mock-SAS server

In all test cases under this category, the Grant Request is correct and valid. That is, the mock-SAS harness has checked that all the "Required" parameters are present in the Grant Request and their values are valid and correct. This part makes sure that CBSD is able to send correct and valid Grant Request.

6.3.4.1.1 [WINNF.FT.C.GRA.1] Successful Grant response form SAS

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	• Configure the Mock-SAS to send successful grant response. • Make sure that the registration procedure has been successfully completed with the Mock-SAS • Make sure CBSD ID (C) exists in mock-SAS.	PASS	FAIL
2	Configure CBSD to send valid and correct Grant request with operation parameters. • Make sure the operation parameters sent by CBSD contain correct values for maxEIRP and operationFrequencyRange. • Make sure the maxEIRP is according to CBSD category. It must be equal to/less than the value defined by FCC Part96. • SAS test harness is configured to send responseCode=0.	PASS	FAIL
3	SAS Harness sends a Grant Response with • CBSD ID = C. • A valid grant id • responseCode = 0	--	--
4	Note: As per specification, CBSD may start the heartbeat procedure at this stage. Details for the Grant and Heartbeat process can be referred from applicable sections.	PASS	FAIL
5	Verify using a spectrum analyzer that CBSD is not transmitting unless an authorized grant is available and first heartbeat procedure was successful.	PASS	FAIL

6.3.4.1.2 [WINNF.FT.D.GRA.2] Successful Grant response from SAS for all requests – Domain Proxy

The following steps describe the test execution.

#	Test Execution Steps	**Results**

1	• Configure the Mock-SAS to send successful gran response to all requests from domain proxy. • Make sure that the registration procedure has been successfully completed with the Mock-SAS for all the CBSD • Make sure CBSD ID (C1..CN) exists in mock-SAS.	PASS	FAIL
2	Configure Domain-Proxy to send valid and correct Grant Request containing data for multiple CBSD with list of frequency ranges. • Make sure the list of frequencyRange objects sent by Domain-Proxy are within CBRS range. SAS test harness is configured to send responseCode=0 for all the requests.	PASS	FAIL
3	SAS Harness sends a Grant Response for all requests with • CBSD ID = Ci. • Grant ID is present in all the grantResponse objects • responseCode = 0	--	--
4	Wait for Grant Request and HeartBeat Request from the Domain-Proxy.	PASS	FAIL
5	Verify using a spectrum analyzer that none of CBSD is transmitting unless an authorized grant is available and first heartbeat procedure was successful.	PASS	FAIL

6.3.4.2 Unsuccessful responses from mock-SAS

The test cases in this section are for verifying the handling of CBSD for various responseCodes in response from mock-SAS.

The actions taken in response of any responseCode are beyond the scope of this document unless mentioned in the test procedure.

6.3.4.2.1 [WINNF.FT.D.GRA.3] Unsuccessful Grant response for requests with various responseCodes – Domain Proxy

The following steps describe the test execution.

#	Test Execution Steps	**Results**	
1	• Make sure that the registration procedure has been successfully completed with the Mock-SAS for all the CBSD • Make sure CBSD ID (C1..CN) exists in mock-SAS.	PASS	FAIL
2	Configure Domain-Proxy to send valid and correct Grant Request containing data for multiple CBSD with list of frequency ranges. • Make sure the list of frequencyRange objects sent by Domain-Proxy are within CBRS range. • Make sure that measReport object is not included in the Grant request message for any record. •	PASS	FAIL
3	SAS Harness sends a Grant Response for requests with • CBSD ID = Ci.	PASS	FAIL

	• responseCode = 0 for some requests and various non-zero response codes for remaining requests Note1: How DP communicates various errors to CBSD are out of scope of this document. Note2: Verify that the next message received by mock-SAS is a valid one as per Table 1. The CBSD should have stayed in *Registered* state and hence the next message received should be one of the messages marked “Yes” in the Registered state.		
4	Verify using a spectrum analyzer that none of the CBSD is transmitting unless an authorized grant is available.	PASS	FAIL

6.4 CBSD Heart Beat Process

6.4.1 Definition and applicability and Scope of Test Case

This section provides procedures for testing CBSD behavior during the Heartbeat Process. It assumes as precondition that CBSD has successfully discovered the SAS that it wants to register with, has successfully registered, has a successful Grant request, and is in the Granted or Authorized state.

The following responseCodes are applicable to Heartbeat Response messages, and will be tested by test cases defined in this section.

Response Codes Applicable to Heartbeat Response Message

response Code	**Name**	**Description**
0	SUCCESS	CBSD request is approved by SAS
105	DEREGISTER	A CBSD receiving this responseCode is automatically deregistered by the SAS. The CBSD shall cease all transmissions, terminate all Grants, and consider itself *Unregistered*. The SAS may include this *responseCode* parameter in any message. The *responseMessage* parameter may contain a string describing the reason for deregistration.
500	TERMINATED _GRANT	The Grant is terminated. This condition occurs if incumbent status has changed permanently, causing the current Grant to terminate. The CBSD shall terminate radio operation by turning off its radio transmission associated with this Grant within 60 seconds after the value of the *transmitExpireTime* parameter expires, in accordance with part 96.39(c)(2) (ref. [n.5]). The Grant is considered terminated by the SAS, but the CBSD may relinquish the Grant. If the *operationParam* parameter is included in the *HeartbeatResponse* object, the CBSD should consider it as a recommendation by the SAS to obtain a new Grant using the included operational parameter values, and may request a new Grant using those operational parameters.

501	SUSPENDED_ GRANT	The Grant is suspended. This condition occurs if incumbent status has changed temporarily. The CBSD shall terminate radio operation by turning off its radio transmission associated with this Grant within 60 seconds after the value of the *transmitExpireTime* parameter expires, in accordance with part 96.39(c)(2) (ref. [n.5]). In such a case the CBSD may continue to send *HeartbeatRequest* objects and waiting until the Grant is re-enabled, or may relinquish the Grant and request another. If the *operationParam* parameter is included in the *HeartbeatResponse* object, the CBSD should consider it as a recommendation by the SAS to obtain a new Grant using the included operational parameter values, and may request a new Grant using those parameters.
502	UNSYNC_OP_ PARAM	Operation parameters or Grant state(s) is/are out of sync between the CBSD and the SAS. In this case, the CBSD shall immediately turn off all radio transmission and shall relinquish the Grant. The CBSD may choose to request another Grant.

6.4.2 Test Characteristics

Table 6-4 CBSD Heart Beat Process Test Characteristics

1	**Test ID**	WINNF.FT.C.HBT
2	**Title**	CBSD-SAS Heartbeat Process
3	**Working Group / Entity**	WG3
4	**Test Type**	Functional (FT)
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification**	[n.5], sections 8.6, 10.7, 10.8

6.4.3 Method of test

6.4.3.1 Initial Conditions / Test Pre-Conditions

Test case entry for all test cases includes:

1. Test harness SAS Discovery and Authentication by CBSD is complete
2. CBSD has registered successfully with test harness SAS
3. CBSD has received successful grant from test harness SAS

6.4.4 Test Procedure

6.4.4.1 Successful Heartbeat (responseCode=0)

The test cases in this section test the success path for the Heartbeat process.

6.4.4.1.1 [WINNF.FT.C.HBT.1] Heartbeat Success Case (first Heartbeat Response)

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with test harness • UUT has a valid single grant as follows:	--	--

	○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in GRANTED, but not AUTHORIZED state (i.e. has not performed its first Heartbeat Request)		
2	UUT sends a first Heartbeat Request message. Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = “GRANTED”	PASS	FAIL
3	Test harness sends a Heartbeat Response message, with the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = current UTC time + 200 seconds • *responseCode* = 0	--	--
4	For further Heartbeat Request messages sent from UUT after completion of step 3, validate: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED” and test harness responds with a Heartbeat Response message including the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = current UTC time + 200 seconds • *responseCode* = 0	PASS	FAIL
5	Monitor the RF output of the UUT from start of test until UUT transmission commences. Verify: • UUT does not transmit at any time prior to completion of step 3 • UUT transmits after step 3 is complete, and its transmission is limited to within the bandwidth range F.	PASS	FAIL

6.4.4.1.2 [WINNF.FT.D.HBT.2] Domain Proxy Heartbeat Success Case (first Heartbeat Response)

This test case applies to Domain Proxy supervising two CBSDs.

The following are the test execution steps.

#	Test Execution Steps	Results	
1	Ensure the following conditions are met for test entry:	--	--

	• DP has two CBSD registered successfully with test harness • Each CBSD {1, 2} has a valid single grant as follows: ○ valid *cbsdId* = Ci, i={1,2} ○ valid *grantId* = Gi, i={1,2} ○ grant is for frequency range Fi, power Pi • Both CBSD are in GRANTED, but not AUTHORIZED state (i.e. neither has not performed its first Heartbeat Request)		
2	Ensure DP sends first Heartbeat Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Ensure Heartbeat Request message is formatted correctly for each CBSD, including, for CBSDi i={1,2}: • *cbsdId* = Ci, i={1,2} • *grantId* = Gi, i={1,2} • *operationState* = "GRANTED"	PASS	FAIL
3	If a separate Heartbeat Request message was sent for each CBSD by the DP, the test harness shall respond to each Heartbeat Request message with a separate Heartbeat Response message. If a single Heartbeat Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Heartbeat Response message containing a 2-object array. Parameters for each CBSD within the Heartbeat Response message should be as follows, for CBSDi: • *cbsdId* = Ci • *grantId* = Gi • *transmitExpireTime* = current UTC time + 200 seconds • *responseCode* = 0	--	--
4	For further Heartbeat Request messages sent from DP after completion of step 3, validate for CBSDi: • *cbsdId* = Ci • *grantId* = Gi • *operationState* = "AUTHORIZED" and test harness responds with a Heartbeat Response message including the following parameters, for CBSDi • *cbsdId* = Ci • *grantId* = Gi • *transmitExpireTime* = current UTC time + 200 seconds • *responseCode* = 0	PASS	FAIL
5	Monitor the RF output of the UUT from start of test until UUT transmission commences. Monitor the RF output of the UUT from start	PASS	FAIL

	of test until TBD seconds after Step 3 is complete. This is the end of the test. Verify: • UUT does not transmit at any time prior to completion of step 3 • UUT transmits after step 3 is complete, and its transmission is limited to within the bandwidth range F.		

6.4.4.2 Unsuccessful Heartbeat Test Cases (responseCode != 0)

The test cases in this section cover Heartbeat Response messages with non-zero responseCodes. Part of the pass/fail criteria of these test cases is the cessation of all UUT RF transmission. Therefore, in all test cases, after the non-zero responseCode is sent, the test harness shall not allow any new Grant Request from the UUT to succeed.

6.4.4.2.1 [WINNF.FT.C.HBT.3] Heartbeat responseCode=105 (DEREGISTER)

The following are the test execution steps.

#	Test Execution Steps	Results	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with Test harness • UUT has a valid single grant as follows: o valid *cbsdId* = C o valid *grantId* = G o grant is for frequency range F, power P • UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface	--	--
2	UUT sends a Heartbeat Request message. Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED”	PASS	FAIL
3	Test harness sends a Heartbeat Response message, including the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime = Current UTC time* • *responseCode* = 105 (DEREGISTER)	--	--
4	After completion of step 3, test harness shall not allow any further grants to the UUT.		
5	Monitor the RF output of the UUT. Verify:	PASS	FAIL

	• UUT shall stop transmission within 60 seconds of completion of step 3		

6.4.4.2.2 [WINNF.FT.C.HBT.4] Heartbeat responseCode=500 (TERMINATED_GRANT)

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with test harness • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface	--	--
2	UUT sends a Heartbeat Request message. Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = "AUTHORIZED"	PASS	FAIL
3	Test harness sends a Heartbeat Response message, including the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = T = current UTC time • *responseCode* = 500 (TERMINATED_GRANT)	--	--
4	After completion of step 3, test harness shall not allow any further grants to the UUT.		
5	Monitor the RF output of the UUT. Verify: • UUT shall stop transmission within bandwidth F within (T + 60 seconds) of completion of step 3	PASS	FAIL

6.4.4.2.3 [WINNF.FT.C.HBT.5] Heartbeat responseCode=501 (SUSPENDED_GRANT)

The following are the test execution steps.

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with Test harness • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P	--	--

	• UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface		
2	UUT sends a Heartbeat Request message. Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED”	PASS	FAIL
3	Test harness sends a Heartbeat Response message, including the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = T = current UTC time • *responseCode* = 501 (SUSPENDED_GRANT)	--	--
4	After completion of step 3, test harness shall not allow any further grants to the UUT.		
5	Monitor the SAS-CBSD interface. Verify either A OR B occurs: A. UUT sends a Heartbeat Response message. Ensure message is correctly formatted with parameters: • *cbsdId* = C • *grantId* = G • *operationState* = “GRANTED” B. UUT sends a Relinquishment Request message. Ensure message is correctly formatted with parameters: • *cbdsId* = C • *grantId* = G Monitor the RF output of the UUT. Verify: • UUT shall stop transmission within bandwidth F within (*T* + 60 seconds) of completion of step 3	PASS	FAIL

6.4.4.2.4 [WINNF.FT.C.HBT.6] Heartbeat responseCode=502 (UNSYNC_OP_PARAM)

The following are the test execution steps.

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with Test harness • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface	--	--
2	UUT sends a Heartbeat Request message.	PASS	FAIL

	Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED”		
3	Test harness sends a Heartbeat Response message, including the following parameters: • *cbsdId* = C • *grantId* = G • *responseCode* = 502 (UNSYNC_OP_PARAM)	--	--
4	After completion of step 3, test harness shall not allow any further grants to the UUT.	--	--
5	Monitor the SAS-CBSD interface. Verify: • UUT sends a Grant Relinquishment Request message. Ensure message is correctly formatted with parameters: o *cbdsId* = C o *grantId* = G Monitor the RF output of the UUT. Verify: • UUT shall stop transmission within 30 seconds of completion of step 3.	PASS	FAIL

6.4.4.2.5 [WINNF.FT.D.HBT.7] Domain Proxy Heartbeat responseCode=500 (TERMINATED_GRANT)

This test case applies to Domain Proxy supervising two CBSDs.

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	Ensure the following conditions are met for test entry: • DP has two CBSD registered successfully with test harness • Each CBSD {1,2} has a valid single grant as follows: o valid *cbsdId* = Ci, i={1,2} o valid *grantId* = Gi, i={1,2} o grant is for frequency range Fi, power Pi • Both CBSD are in AUTHORIZED state and transmitting within their granted bandwidth on RF interface	--	--
2	DP sends a Heartbeat Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of size 2. Ensure Heartbeat Request message is formatted correctly for each CBSD, including, for CBSDi i={1,2}: • *cbsdId* = Ci, i = {1,2} • *grantId* = Gi, i = {1,2}	PASS	FAIL

	• *operationState* = “AUTHORIZED”		
3	If separate Heartbeat Request message was sent for each CBSD by the DP, the test harness shall respond to each Heartbeat Request message with a separate Heartbeat Response message. If a single Heartbeat Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Heartbeat Response message containing a 2-object array. Parameters for each CBSD within the Heartbeat Response message should be as follows, for CBSDi: • *cbsdId* = Ci • *grantId* = Gi • For CBSD1: o *transmitExpireTime* = current UTC time + 200 seconds o *responseCode* = 0 • For CBSD2: o *transmitExpireTime* = T = current UTC time o *responseCode* = 500 (TERMINATED_GRANT)	--	--
4	After completion of step 3, test harness shall not allow any further grants to the UUT. If CBSD sends further Heartbeat Request messages for CBSD1, test harness shall respond with a Heartbeat Response message with parameters: • *cbsdId* = C1 • *grantId* = G1 • *transmitExpireTime* = current UTC time + 200 seconds • *responseCode* = 0	--	--
5	Monitor the RF output of CBSD2. Verify: • CBSD2 shall stop transmission within bandwidth F2 within (T + 60 seconds) of completion of step 3	PASS	FAIL

6.4.4.3 Heartbeat Response Absent Test Cases

These test cases cover the case where communication is lost between the UUT and the SAS during the Heartbeat Process.

6.4.4.3.1 [WINNF.FT.C.HBT.8] Heartbeat Response Absent (First Heartbeat)

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	Ensure the following conditions are met for test entry:	--	--

	• UUT has registered successfully with Test harness • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in GRANTED, but not AUTHORIZED state (i.e. has not performed its first Heartbeat Request)		
2	UUT sends a Heartbeat Request message. Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = "GRANTED"	PASS	FAIL
3	After completion of Step 2, test harness does not respond to any further messages from UUT	--	--
4	Monitor the RF output of the UUT from start of test to 60 seconds after step 3. Verify: • At any time during the test, UUT shall not transmit on RF interface	PASS	FAIL

6.4.4.3.2 [WINNF.FT.C.HBT.9] Heartbeat Response Absent (Subsequent Heartbeat)

The following are the test execution steps.

#	Test Execution Steps	Results	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with Test harness • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface	--	--
2	UUT sends a Heartbeat Request message. Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = "AUTHORIZED"	PASS	FAIL
3	Test harness sends a Heartbeat Response message, with the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = current UTC time + 240 seconds • *responseCode* = 0	--	--

4	After completion of Step 3, test harness does not respond to any further messages from UUT	--	--
5	Monitor the RF output of the UUT. Verify: • UUT shall stop all transmission on RF interface within (*transmitExpireTime* + 60 seconds), using the transmitExpireTime sent in Step 3.	PASS	FAIL

6.4.4.4 Heartbeat Grant Renewal and Expiry Test Cases

Test cases in this section test Grant Renewal and Grant Expiry within the Heartbeat Process.

6.4.4.4.1 [WINNF.FT.C.HBT.10] Successful Grant Renewal in Heartbeat Test Case

The following are the test execution steps.

#	Test Execution Steps	Results	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with Test harness • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface. • Grant has the following parameters at the start of the test: ○ grantExpireTime =UTC time equal to time at start of test + 240 seconds = Tgrant_expire ○ transmitExpireTime = UTC time equal to time at start of test + 240 seconds ○ heartbeatInterval = 60 seconds	--	--
2	UUT sends a Heartbeat Request message. If Heartbeat Request message contains grantRenew = TRUE, go to Step 6, else go to Step 3.	--	--
3	Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED”	PASS	FAIL
4	Test harness sends a Heartbeat Response message, with the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = same as Step 1	--	--

	• *grantExpireTime* = same as Step 1 • *responseCode* = 0		
5	Go to Step 2	--	--
6	Ensure Heartbeat Request message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = "AUTHORIZED" • *grantRenew* = TRUE	PASS	FAIL
7	Test harness sends a Heartbeat Response message, with the following parameters: • *cbsdId* = C • *grantId* = G • *grantExpireTime* = UTC time of far in the future • *transmitExpireTime* = current UTC time + 240 seconds • *responseCode* = 0	--	--
8	Continue to respond to any subsquentHeartbeat Request from CBSD with Heartbeat Response with the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = same as Step 7 • *responseCode* = 0		
8	Monitor RF transmission of UUT from start of test until Tgrant_expire + 60 seconds and ensure UUT continues to transmit throughout the time period.	PASS	FAIL

6.4.4.4.2 [WINNF.FT.C.HBT.11] Grant Expiry in Heartbeat Test Case

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has registered successfully with Mock-SAS • UUT has a valid single grant as follows: ○ valid *cbsdId* = C ○ valid *grantId* = G ○ grant is for frequency range F, power P • UUT is in AUTHORIZED state and is transmitting within the grant bandwidth F on RF interface. • Grant has the following parameters at the start of the test: ○ grantExpireTime = Tgrant_expire = UTC time equal to time at start of test + TBD seconds ○ transmitExpireTime = UTC time equal to time at start of test + 240 seconds ○ heartbeatInterval = 120 seconds	--	--

	• Test harness shall not provide any other grant to UUT during test.		
2	UUT sends a Heartbeat Request message. Ensure message is formatted correctly, including: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED” • *grantRenew* may be absent or present	PASS	FAIL
3	Test harness sends a Heartbeat Response message, with the following parameters: • *cbsdId* = C • *grantId* = G • *transmitExpireTime* = min {(current UTC time + 200 seconds), Tgrant_expire } • *grantExpireTime* = Tgrant_expire • *responseCode* = 0	--	--
4	Repeat Steps 2 and 3 until time = Tgrant_expire	--	--
8	Monitor UUT RF output during test, from start time to Tgrant_expire + 60 seconds. Ensure UUT RF transmission ceases by (Tgrant_expire+60 seconds).	PASS	FAIL

6.5 CBSD Measurement Report

6.5.1 Definition and applicability and Scope of Test Case

This section explains test steps/condition/procedure for CBSD behavior for Measurement Reports.

The main test cases for Measurement Report are outlined below, in terms of Measurement Report Stimulus (in a Response message from SAS) and a Measurement Report Response (in the next Request message from the UUT).

Measurement Report Test Cases

Case	**Stimulus (from SAS)**	**Response (from CBSD)**	**CBSD Capability**	
			WITHOUT_GRANT**	**WITH_GRANT****

1	RegistrationResponse(*measReportConfig*)	Next SpectrumInquiryRequest(*measReport*)* AND GrantRequest(*measReport*)	M	n/a
2	GrantResponse(*measReportConfig*)	Next HeartbeatRequest(*measReport*)	n/a	M
3	HeartbeatResponse(*measReportConfig*)	Next HeartbeatRequest(*measReport*)	n/a	M

* *Note: SpectrumInquiryRequest is an optional message. If present, measReport shall be in both SpectrumInquiryRequest and GrantRequest*

** *Indicates test is Mandatory (M) or n/a to UUT which supports RECEIVED_POWER_WITHOUT_GRANT or RECEIVED_POWER_WITH_GRANT*

Below are the Measurement Report element enumerations and formats (from [n.7]).

Measurement Report Element Formats

Parameter	**type**	**values**
measCapability	array of string	RECEIVED_POWER_WITHOUT_GRANT RECEIVED_POWER_WITH_GRANT
measReportConfig	array of string	RECEIVED_POWER_WITHOUT_GRANT RECEIVED_POWER_WITH_GRANT
measReport	Object of *rcvdPowerMeasReports*	
rcvdPowerMeasReport	*measFrequency*	Frequency of the lowest end of the measured frequency range in Hz.
	measBandwidth	Measurement bandwidth in Hz used by CBSD to perform the Received Power measurement. The range bounded by *measFrequency* as the lower bound and (*measFrequency* + *measBandwidth*) as the upper bound expresses the frequency range used in making the measurement.
	measRcvdPower	Received Power measurement in units of dBm. The range of this parameter is -100 dBm .. -25dBm. The Received Power is measured over the frequency range from *measFrequency* as the lower bound to (*measFrequency* + *measBandwidth*) as the upper bound.

Since there is no measurement accuracy requirement, Measurement Reports are tested for proper format, but not for reporting accuracy.

6.5.2 Test Characteristics

Table 6-5 CBSD Measurement Report Process Test Characteristics

1	**Test ID**	WINNF.FT.C.MES
2	**Title**	CBSD-SAS Measurement Report
3	**Working Group / Entity**	WG3
4	**Test Type**	Functionality
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
6	**Target Specification**	[n.5], sections 8.3, 8.4, 8.5, 8.6, [n.7]

6.5.3 Method of test

6.5.3.1 Initial Conditions / Test Preconditions

Test case entry for all test cases includes:

1. Test harness SAS Discovery and Authentication by CBSD is complete

Additional entry requirements are outlined in each test case.

6.5.4 Test Procedure

6.5.4.1 Measurement Capability Test Cases

Test cases in this section test the proper reporting of Measurement Capability by the CBSD.

6.5.4.1.1 [WINNF.FT.C.MES.1] Measurement Capability in Registration Request Message

This test case is mandatory for all CBSDs.

The following steps describes the test execution steps:

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness	--	--
2	UUT sends a Registration Request message. Ensure Registration Request message is formatted correctly, including: • *userId* is present and correct • *fccId* is present and correct	PASS	FAIL

	• *cbsdSerialNumber* is present and correct • *measCapability* = one or more of the valid Measurement Capability types: {“RECEIVED_POWER_WITHOUT_GRANT”, “RECEIVED_POWER_WITH_GRANT”}		

6.5.4.2 Measurement Report Test Cases

Test cases in this section test the success path for each possible Measurement Report

6.5.4.2.1 [WINNF.FT.C.MES.2] Registration Response contains *measReportConfig*

This test case is mandatory for CBSD supporting RECEIVED_POWER_WITHOUT_GRANT.

The following steps describes the test execution steps:

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness	--	--
2	UUT sends a Registration Request message. Ensure Registration Request message is formatted correctly, including: • *userId* is present and correct • *fccId* is present and correct • *cbsdSerialNumber* is present and correct • *measCapability* = “RECEIVED_POWER_WITHOUT_GRANT”	PASS	FAIL
3	Test harness sends a Registration Response message, with the following parameters: • *cbsdId* = C = valid cbsdId for this UUT • *measReportConfig*= “RECEIVED_POWER_WITHOUT_GRANT” • *responseCode* = 0	--	--
4	UUT sends a message: • If message is type Spectrum Inquiry Request, go to step 5, or • If message is type Grant Request, go to step 7	--	--
5	UUT sends message type Spectrum Inquiry Request. Ensure message contains all required parameters properly formatted, and specifically: • *cbsdId* = C	PASS	FAIL

	• *measReport* is present, and is a properly formatted *rcvdPowerMeasReport.*		
6	Test harness sends a Spectrum Inquiry Response, with the following parameters: • *cbsdId* = C • *availableChannel* is an array of *availableChannel* objects • *responseCode* = 0		
7	UUT sends message type Grant Request message. Ensure message contains all required parameters properly formatted, and specifically: • *cbsdId* = C • *measReport* is present, and is a properly formatted *rcvdPowerMeasReport.*	PASS	FAIL

6.5.4.2.2 [WINNF.FT.D.MES.3] Domain Proxy: Registration Response contains *measReportConfig*

This test case is mandatory for Domain Proxy supervising CBSD which support RECEIVED_POWER_WITHOUT_GRANT.

The following steps describes the test execution steps:

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with test harness	--	--
2	DP sends a Registration Request message for each of two CBSD. This may occur in a separate Request message per CBSD, or together in a single Request message with array of 2. Ensure Registration Request message contains all required parameters properly formatted for CBSDi, i={1,2}, and specifically: • *userId* is present and correct • *fccId* is present and correct • *cbsdSerialNumber* is present and correct • *measCapability* = "RECEIVED_POWER_WITHOUT_GRANT"	PASS	FAIL
3	If a separate Registration Request message was sent for each CBSD by the DP, the test harness shall respond to each Registration Request message with a separate Registration Response message.	--	--

	If a single Registration Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Registration Response message containing a 2-object array. Parameters for each CBSD within the Registration Response message should be as follows, for CBSDi: • *cbsdId* = Ci • *measReportConfig*= “RECEIVED_POWER_WITHOUT_GRANT” • *responseCode* = 0		
4	UUT sends a message: • If message is type Spectrum Inquiry Request, go to step 5, or • If message is type Grant Request, go to step 7	--	--
5	UUT sends message type Spectrum Inquiry Request. This may occur in a separate message per CBSD, or together in a single message with array of 2. Ensure Spectrum Inquiry Request message contains all required parameters properly formatted for CBSDi, i= {1,2}, and specifically: • *cbsdId* = Ci • *measReport* is present, and is a properly formatted *rcvdPowerMeasReport.*	PASS	FAIL
6	If a separate Spectrum Inquiry Request message was sent for each CBSD by the DP, the test harness shall respond to each Spectrum Inquiry Request message with a separate Spectrum Inquiry Response message. If a single Spectrum Inquiry Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Spectrum Inquiry Response message containing a 2-object array. Parameters for each CBSD within the Spectrum Inquiry Response message should be as follows: • *cbsdId* = Ci • *availableChannel* is an array of *availableChannel* objects • *responseCode* = 0		
7	UUT sends message type Grant Request message. This may occur in a separate message per CBSD, or together in a single message with array of 2. Ensure the Grant Request message contains all required parameters properly formatted for CBSDi, i= {1,2}, and specifically:	PASS	FAIL

	• *cbsdId* = Ci • *measReport* is present, and is a properly formatted *rcvdPowerMeasReport.*		

6.5.4.2.3 [WINNF.FT.C.MES.4] Grant Response contains *measReportConfig*

This test case is mandatory for UUT supporting RECEIVED_POWER_WITH_GRANT measurement reports.

The following steps describes the test execution steps:

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT has successfully registered with test harness, with *cbsdId*=C and *measCapability* = “RECEIVED_POWER_WITH_GRANT”	--	--
2	UUT sends a Grant Request message. Ensure Grant Request message contains all required parameters properly formatted, and specifically: • *cbsdId* = C • *operationParam* is present and format is valid	PASS	FAIL
3	Test harness sends a Grant Response message, with the following parameters: • *cbsdId* = C • *grantId* = G = valid grant ID • *grantExpireTime* = UTC time in the future • *heartbeatInterval* = 30 seconds • *measReportConfig*= “RECEIVED_POWER_WITH_GRANT” • *operationParam* is set to valid operating parameters • *channelType* = “GAA” • *responseCode* = 0	--	--
4	UUT sends a Heartbeat Request message. Ensure message contains all required parameters properly formatted, and specifically: • *cbsdId* = C • *grantId* = G	PASS	FAIL
5	If Heartbeat Request message (step 4) contains *measReport* object, then:	PASS	FAIL

	• ensure *measReport* is properly formatted as object *rcvdPowerMeasReport* • end test, with PASS result else, if Heartbeat Request message (step 4) does not contain measReport object, then: If number of Heartbeat Requests sent by UUT after Step 3 is = 5, then stop test with result of FAIL		
6	Test harness sends a Heartbeat Response message, containing all required parameters properly formatted, and specifically: • *cbsdId* = C • *grantId = G* • *responseCode* = 0 Go to Step 4, above	--	--

6.5.4.2.4 [WINNF.FT.C.MES.5] Heartbeat Response contains *measReportConfig*

This test case is mandatory for UUT supporting RECEIVED_POWER_WITH_GRANT measurement reports.

The following steps describes the test execution steps:

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with test harness • UUT has successfully registered with test harness, with *cbsdId*=C and *measCapability* = “RECEIVED_POWER_WITH_GRANT” • UUT has received a valid grant with grantId = G • UUT is in Grant State AUTHORIZED and is actively transmitting within the bounds of its grant. • Grant has *heartbeatInterval* = 60 seconds	--	--
2	UUT sends a Heartbeat Request message. Ensure Heartbeat Request message contains all required parameters properly formatted, and specifically: • *cbsdId* = C • *grantId* = G • *operationState* = “AUTHORIZED”	PASS	FAIL
3	Test harness sends a Heartbeat Response message, containing all required parameters properly formatted, and specifically: • *cbsdId* = C	--	--

	• *grantId = G* • *measReportConfig*= "RECEIVED_POWER_WITH_GRANT" • *responseCode* = 0		
4	UUT sends a Heartbeat Request message. Ensure message contains all required parameters properly formatted, and specifically: • *cbsdId* = C • *grantId* = G • operationState = "AUTHORIZED"	PASS	FAIL
5	If Heartbeat Request message (step 4) contains *measReport* object, then: • ensure *measReport* is properly formatted as object *rcvdPowerMeasReport* • end test, with PASS result else, if Heartbeat Request message (step 4) does not contain measReport object, then: • If number of Heartbeat Requests sent by UUT after Step 3 is = 5, then stop test with result of FAIL	PASS	FAIL
6	Test harness sends a Heartbeat Response message, containing all required parameters properly formatted, and specifically: • *cbsdId* = C • *grantId = G* • *responseCode* = 0 Go to Step 4, above	--	--

6.5.4.2.5 [WINNF.FT.D.MES.6] Domain Proxy: Heartbeat Response contains *measReportConfig*

This test case is mandatory for Domain Proxy supervising CBSD which support RECEIVED_POWER_WITH_GRANT measurement reports.

The following steps describes the test execution steps:

#	Test Execution Steps	**Results**	
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with test harness	--	--

	• DP has successfully registered 2 CBSD with test harness, each with *cbsdId*=Ci, i={1,2} and *measCapability* = “RECEIVED_POWER_WITH_GRANT” • DP has received a valid grant with grantId = Gi, i={1,2} for each CBSD • Both CBSD are in Grant State AUTHORIZED and actively transmitting within the bounds of their grants. • Grants have *heartbeatInterval* =60 seconds		
2	Ensure DP sends a Heartbeat Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Ensure Heartbeat Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • *cbsdId* = Ci • *grantId* = Gi • *operationState* = “AUTHORIZED”	PASS	FAIL
3	If a separate Heartbeat Request message was sent for each CBSD by the DP, the test harness shall respond to each Heartbeat Request message with a separate Heartbeat Response message. If a single Heartbeat Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Heartbeat Response message containing a 2-object array. Parameters for each CBSD within the Heartbeat Response message containing all required parameters properly formatted, and specifically: • *cbsdId* = Ci • *grantId = Gi* • *measReportConfig*= “RECEIVED_POWER_WITH_GRANT” • *responseCode* = 0	--	--
4	Ensure DP sends a Heartbeat Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Ensure Heartbeat Request message contains all required parameters properly formatted for each CBSD, and specifically, for CBSDi, i = {1,2}: • *cbsdId* = Ci • *grantId* = Gi • operationState = “AUTHORIZED” • Check whether *measReport* is present, and if present, ensure it is a properly formatted *rcvdPowerMeasReport* object, and record its reception for each CBSDi, i = {1,2}.	PASS	FAIL

5	If Heartbeat Request message (step 4) contains *measReport* object, then: • ensure *measReport* is properly formatted as object *rcvdPowerMeasReport* • record which CBSD have successfully sent a *measReport* object If all CBSDi, i = {1,2} have successfully sent a measReport object, then • end test, with PASS result else, if the number of Heartbeat Requests sent per CBSD is 5 or more, then stop test with result of FAIL	PASS	FAIL
6	If a separate Heartbeat Request message was sent for each CBSD by the DP, the test harness shall respond to each Heartbeat Request message with a separate Heartbeat Response message. If a single Heartbeat Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Heartbeat Response message containing a 2-object array. Parameters for each CBSD within the Heartbeat Response message containing all required parameters properly formatted, and specifically: • *cbsdId* = Ci • *grantId = Gi* • *responseCode* = 0 Go to Step 4, above.	--	--

6.6 CBSD Relinquishment Process

6.6.1 Definition and applicability and Scope of Test Case

This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Relinquishment Procedure. It assumes as a precondition the CBSD has successfully discovered the SAS it wants to communicate with.

The main approach is for each test to generate a CBSD relinquishment request and to validate the CBSD takes the appropriate action following the SAS relinquishment response.

6.6.2 Test Characteristics

Table 6-6 CBSD Relinquish Process Test Characteristics

1	**Test ID**	WINNF.FT.C.RLQ
2	**Title**	CBSD-SAS Relinquishment Process

3	**Working Group / Entity**	WG3
4	**Test Type**	Functionality
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification**	[n.5], sections 8.7, 10.9 and 10.10

6.6.3 Method of Test

6.6.3.1 Initial Conditions / Test Pre-conditions

- The pre-conditions of the test case are:
  - CBSD has gone through SAS discovery process and can authenticate with the SAS. The exact condition of the CBSD after the discovery process are detailed in each test case.
  - The CBSD has already registered with SAS and has obtained a valid CBSD ID.

- The applicable structure of the relinquishmentRequest parameter and relinquishmentResponse objects are defined in [n.5].

In summary, the CBSD parameters used for the process are categorized into the following [n.5]:

- Required
- Conditional
- Optional

6.6.4 Test Procedure

6.6.4.1 Successful Relinquishment Request (responseCode 0)

6.6.4.1.1 [WINNF.FT.C.RLQ.1] Valid and correct cbsdId and grantId

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state Invoke trigger to relinquish UUT Grant from the SAS, UUT shall cease transmission associated with the associated Grant		
1	UUT sends Relinquishment Request to SAS Test Harness including its cbsdId and grantId	PASS	FAIL
2	SAS Test Harness approves the request by sending a Relinquishment Response as follows: – SAS response shall include same cbsdId as in the request. – SAS response shall include same grantId as in the request	-	-

	– The responseCode in Response Object shall be 0.		
3	UUT receives Relinquishment Response and obtains the responseCode = 0.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.6.4.1.2 [WINNF.FT.D.RLQ.1] Valid and correct cbsdId and grantId: Domain Proxy serving two CBSDs

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with test harness • Each UUT is in the authorized state Invoke trigger to relinquish each UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	DP with two CBSDs sends Relinquishment Request with two RelinquishmentRequest object to the SAS Test Harness. This may occur in a separate message per CBSD, or together in a single message with array of 2. The two objects should have {cbsdId C1, grantId G1} and {cbsdId C2 grantId G2} in that order.	PASS	FAIL
2	If a separate Relinquishment Request message was sent for each CBSD by the DP, the test harness shall respond to each request message with a separate response message. If a single Relinquishment Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Deregistration Response shall be as follows: • cbsdId in the first object = C1 • grantId in the first object = G1 • responseCode in the first object = 0 • cbsdId in the second object = C2 • grantId in the second object = G2 • responseCode in the second object = 0 ○	-	-
3	Each CBSD receives CBSD Relinquishment Response and obtains the responseCode = 0.	PASS	FAIL
4	Monitor the RF output of each UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.6.4.2 Missing Parameter (responseCode 102)

CBSD under test cannot be expected to generate a message with a missing or invalid parameter. To test for responseCode not equal to 0, the SAS test harness will respond to a message with a non-zero responseCode.

6.6.4.2.1 [WINNF.FT.C.RLQ.2] Missing parameter grantId: single object in the RelinquishmentRequest

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state Invoke trigger to Relinquish UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	UUT sends Relinquishment Request to SAS Test Harness	-	-
2	The SAS Test Harness sends the Relinquishment Response Message to UUT with: • cbsdId as in the request • No grantId • responseCode (Response) = 102	-	-
3	UUT receives Relinquishment Response and obtains the responseCode = 102.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.6.4.2.2 [WINNF.FT.D.RLQ.2] Missing parameter grantId: Domain Proxy serving two CBSDs

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with test harness • Each UUT is in the authorized state Invoke trigger to Relinquish each UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	DP with two CBSDs sends Relinquishment Request with two objects to the SAS Test Harness. This may occur in a separate message per CBSD, or together in a single message with array of 2.	-	-
2	If a separate Relinquishment Request message was sent for each CBSD by the DP, the test harness shall respond to each request message with a separate response message.	-	-

	If a single Relinquishment Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Relinquishment Response Message shall be as follows: • cbsdId as in the request for CBSD #1 • No grantId in the first object • responseCode in the first object = 102 • cbsdId as in the request for CBSD #2 • No grantId in the second object • responseCode in the second object = 102		
3	each UUT receives Relinquishment Response and obtains the responseCode = 102.	PASS	FAIL
4	Monitor the RF output of each UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.6.4.3 Invalid Parameter (responseCode 103)

CBSD under test cannot be expected to generate a message with a missing or invalid parameter. To test for responseCode not equal to 0, the SAS test harness will respond to a message with a non-zero responseCode.

6.6.4.3.1 [WINNF.FT.C.RLQ.3] grantId Does Not Exist in the SAS: single request object

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state Invoke trigger to Relinquish UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	UUT sends Relinquishment Request to SAS Test Harness including cbsdId (C).	-	-
2	The SAS Test Harness sends the Relinquishment Response Message to the UUT with: • Same cbsdID as in the request • No grantId • responseCode= 103 • responseData set as “grantId”	-	-
3	UUT receives Relinquishment Response and obtains the responseCode = 103.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify:	PASS	FAIL

	UUT shall stop RF transmission prior to step 1		

6.6.4.3.2 [WINNF.FT.C.RLQ.4] cbsdId Does Not Exist in the SAS: single request object

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state Invoke trigger to Relinquish UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	UUT sends Relinquishment Request to SAS Test Harness including cbsdId (C).	-	-
2	The SAS Test Harness sends the Relinquishment Response Message to the UUT with: • No cbsdID • No grantId • responseCode= 103 • responseData set as “cbsdId”	-	-
3	UUT receives Relinquishment Response and obtains the responseCode = 103.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.6.4.3.3 [WINNF.FT.D.RLQ.3] grantId Does Not Exist in SAS: Domain Proxy serving two CBSDs

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with test harness • Each UUT is in the authorized state Invoke trigger to Relinquish each UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	DP with two CBSDs sends Relinquishment Request with two objects to the SAS Test Harness. This may occur in a separate message per CBSD, or together in a single message with array of 2.	-	-
2	If a separate Relinquishment Request message was sent for each CBSD by the DP, the test harness shall respond to each request message with a separate response message.	-	-

	If a single Relinquishment Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Relinquishment Response Message shall be as follows: • cbsdId as in the request for CBSD #1 • No grantId in the first object • responseCode in the first object = 103 • responseData in the first object = grantId • cbsdId as in the request for CBSD #2 • No grantId in the second object • responseCode in the second object = 103 • responseData in the second object = grantId		
3	each UUT receives Relinquishment Response and obtains the responseCode = 103.	PASS	FAIL
4	Monitor the RF output of each UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.6.4.3.4 [WINNF.FT.D.RLQ.4] cbsdId Does Not Exist in SAS: Domain Proxy serving two CBSDs

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with test harness • Each UUT is in the authorized state Invoke trigger to Relinquish each UUT Grant from the SAS, UUT shall cease transmission associated with any Grants		
1	DP with two CBSDs sends Relinquishment Request with two objects to the SAS Test Harness. This may occur in a separate message per CBSD, or together in a single message with array of 2.	-	-
2	If a separate Relinquishment Request message was sent for each CBSD by the DP, the test harness shall respond to each request message with a separate response message. If a single Relinquishment Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Relinquishment Response Message shall be as follows: • No cbsdId in the first object	-	-

	• No grantId in the first object • responseCode in the first object = 103 • responseData in the first object = cbsdId • No cbsdId in the second object • No grantId in the second object • responseCode in the second object = 103 • responseData in the second object = cbsdId		
3	each UUT receives Relinquishment Response and obtains the responseCode = 103.	PASS	FAIL
4	Monitor the RF output of each UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.7 CBSD Deregistration Process

6.7.1 Definition and applicability and Scope of Test Case

This section explains test steps/condition/procedure for the CBSD Deregistration Request and its subsequent actions following the reception of the Deregistration Responses from the SAS.

A Deregistration request is issued by a CBSD to request a SAS to deregister the CBSD from the SAS. A Deregistration Request Message issued by a CBSD is provided in [n.5], Section 10.11.

In the Deregistration Response message, the SAS should echo back an array of DeregistrationResponse object. Each deregistrationResponse object consists of a cbsdId and a responseCode. If the deregistration request was successful, the responseCode should be set to 0, otherwise responseCode is set to appropriate error value. The deregistrationResponse Message and the deregistrationResponse object are provided in [n.5], Section 10.12.

The main approach is for each test to generate a CBSD deregistration request and take appropriate actions following the SAS deregistration response covering all the defined responseCodes available.
These deregistration test cases assume the CBSD is the source (operator initiated, for instance reset site). Deregistrations triggered by the SAS in a response message with a responseCode of 105 are covered in other test cases.

6.7.2 Test Characteristics

Table 6-7 CBSD Deregistration Process Test Characteristics

1	**Test ID**	WINNF.FT.C.DRG
2	**Title**	CBSD Deregistration Process
3	**Working Group / Entity**	WG3
4	**Test Type**	Functionality

5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification / Feature**	[n.5], sections 8.8, 10.11, and 10.12

6.7.3 Method of test

6.7.3.1 Initial Conditions / Test Pre-conditions

The typical pre-conditions of the test case are the following:

- CBSD has obtained its CBSD ID (C) after successfully registering with the SAS test harness.
- CBSD has requested and received a successful grant. The CBSD is in the authorized state and is transmitting.
- CBSD UUT must provide functionality to trigger a deregistration.
- From [n.5], the CBSD should send a RelinquishmentRequest object for each Grant prior to sending the DeregistrationRequest object. The CBSD will cease RF transmission with the relinquishment message.

6.7.4 Test Procedure

A Deregistration request is issued by a CBSD to request a SAS to deregister the CBSD from the SAS. A Deregistration Request Message issued by a CBSD is provided in [n.5], Section 10.11.

6.7.4.1 Successful Deregistration Request (responseCode 0)

6.7.4.1.1 [WINNF.FT.C.DRG.1] Valid and correct cbsdId

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state Invoke trigger to deregister UUT from the SAS, UUT shall cease transmission associated with any Grants		
1	UUT sends Deregistration Request to SAS Test Harness including its cbsdId.	PASS	FAIL
2	SAS Test Harness approves the request by sending a Deregistration Response as follows: – SAS response shall include same cbsdId as in the request. – The responseCode in Response Object shall be 0.	-	-
3	UUT receives Deregistration Response and obtains the responseCode = 0.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.7.4.1.2 [WINNF.FT.C.DRG.2] Valid and correct cbsdId: Domain Proxy serving two CBSDs

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with test harness • Each UUT is in the authorized state Invoke trigger to deregister each UUT from the SAS, UUT shall cease transmission associated with any Grants		
1	DP with two CBSDs sends Deregistration Request with two DeregistrationRequest object to the SAS Test Harness. This may occur in a separate message per CBSD, or together in a single message with array of 2. The two objects should have cbsdIds C1 and C2 in that order.	PASS	FAIL
2	If a separate Deregistration Request message was sent for each CBSD by the DP, the test harness shall respond to each request message with a separate response message. If a single Deregistration Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Deregistration Response shall be as follows: • cbsdId in the first object = C1 • responseCode in the first object = 0 • cbsdId in the second object = C2 • responseCode in the second object = 0 ○	-	-
3	Each CBSD receives CBSD Deregistration Response and obtains the responseCode = 0.	PASS	FAIL
4	Monitor the RF output of each UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.7.4.2 Missing Parameter (responseCode 102)

CBSD under test cannot be expected to generate a message with a missing or invalid parameter. To test for responseCode not equal to 0, the SAS test harness will respond to a message with a non-zero responseCode.

6.7.4.2.1 [WINNF.FT.C.DRG.3] Missing cbsdId: single object in the DeregistrationRequest

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state		

	Invoke trigger to deregister UUT from the SAS, UUT shall cease transmission associated with any Grants		
1	UUT sends Deregistration Request to SAS Test Harness	-	-
2	The SAS Test Harness sends the Deregistration Response Message to UUT with: • No cbsdId • responseCode (Response) = 102	-	-
3	UUT receives Deregistration Response and obtains the responseCode = 102.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 4 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.7.4.2.2 [WINNF.FT.C.DRG.4] Missing cbsdId: Domain Proxy serving two CBSDs

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with test harness • Each UUT is in the authorized state Invoke trigger to deregister each UUT from the SAS, UUT shall cease transmission associated with any Grants		
1	DP with two CBSDs sends Deregistration Request with two objects to the SAS Test Harness. This may occur in a separate message per CBSD, or together in a single message with array of 2.	-	-
2	If a separate Deregistration Request message was sent for each CBSD by the DP, the test harness shall respond to each request message with a separate response message. If a single Deregistration Request message was sent by the DP containing a 2-object array (one per CBSD), the test harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Deregistration Response Message shall be as follows: • No cbsdId in the first object • responseCode in the first object = 102 • No cbsdId in the second object • responseCode in the second object = 102	-	-
3	each UUT receives Deregistration Response and obtains the responseCode = 102.	PASS	FAIL
4	Monitor the RF output of each UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: • UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.7.4.3 Invalid Parameter (responseCode 103)

CBSD under test cannot be expected to generate a message with a missing or invalid parameter. To test for responseCode not equal to 0, the SAS test harness will respond to a message with a non-zero responseCode.

6.7.4.3.1 [WINNF.FT.C.DRG.5] cbsdId Does Not Exist in the SAS: single request object

The following are the test execution steps.

#	Test Execution Steps	Results	
	Ensure the following conditions are met for test entry: • UUT has successfully registered with test harness • UUT is in the authorized state Invoke trigger to deregister UUT from the SAS, UUT shall cease transmission associated with any Grants		
1	UUT sends Deregistration Request to SAS Test Harness including cbsdId (C).	-	-
2	The SAS Test Harness sends the Deregistration Response Message to the UUT with: • No cbsdId • responseCode= 103 • responseData set as “cbsdId”	-	-
3	UUT receives Deregistration Response and obtains the responseCode = 103.	PASS	FAIL
4	Monitor the RF output of the UUT from start of test until 60 seconds after Step 3 is complete. This is the end of the test. Verify: UUT shall stop RF transmission prior to step 1	PASS	FAIL

6.8 CBSD Security Validation

6.8.1 Definition and applicability and Scope of Test Case

This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the Security Establishment Procedure. It assumes as a precondition the CBSD has successfully discovered the SAS it wants to communicate with.

The main approach is for each test to initiate communication between CBSD and SAS and verify that the communication is started over a secured communication.

6.8.2 Test Characteristics

Table 6-8 CBSD Security Process Test Characteristics

1	**Test ID**	WINNF.FT.C.SCS
2	**Title**	CBSD Security Validation Process
3	**Working Group / Entity**	WG4

4	**Test Type**	Functionality
5	**Test Class**	Certification
6	**Component / Interface**	CBSD / CBSD ← → SAS
7	**Target Specification**	[n.4]

6.8.3 Method of test

6.8.3.1 Initial Conditions / Test Pre-conditions

- The pre-conditions of the test case are:
  - CBSD has gone through SAS discovery process. The exact condition of the CBSD after the discovery process are detailed in each test case.
  - The test certificates are loaded on CBSD as well as on Mock-SAS.

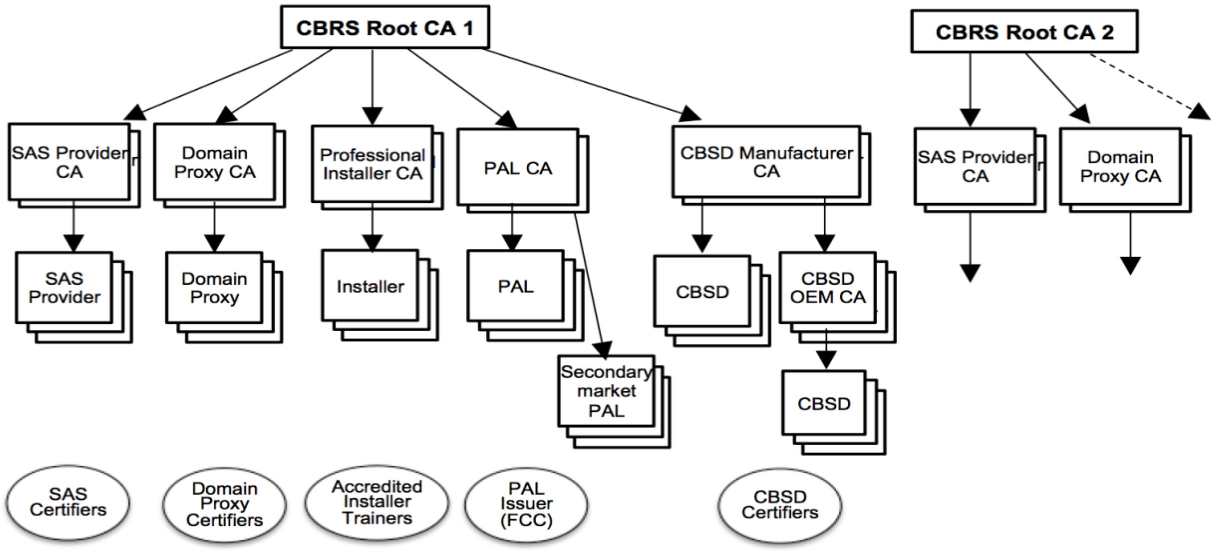
6.8.3.2 Test Certificate Generation & Use

This section describes the generation of "test certificates" as per [n.8] and their use for CBSD certification testing. Though the following sections describe and recommend the "test" certificates use for CBSD security procedure verification assuming certificates generated by [i.1] scripts, however, this does not prevent CBSD test harnesses from using other scripts / methods to generate "test" certificates for use. In case different scripts / methods are used for generating "test" certificates, the generated certificates need to follow the certificate profile described in [n.8], with the modifications as outlined in [6.8.3.2.1]. The method of use described in the section below remains same for "test" certificates generated using a different script / method.

6.8.3.2.1 Test Certificate Generation

Every execution of [i.1] script is designed to generate "test" certificate for each entity$^{\text{Note-1}}$ – Root CA, Sub-CA, End Entity – of the CBRS PKI Hierarchy in [n.8], shown below:

Note-1: Current WinnF GitHub cert repo scripts do not generate PAL CA and PAL end entity certificates.



Following are the key files in WinnF GitHub cert repo for generation of test certificates:

- Openssl.conf – this is a configuration file for the cert generation scripts. This file provides configuration for the Role OIDs as per [n.8] and other certificate fields like CN, policy id, basic constraints for different certificates etc.
- CACreationScript.sh – this is linux cert generation script. This script generates RSA "test" certificates for the following CBRS PKI hierarchy entities:
  - CBRS Root CA
  - Sub-CAs signed by Root CA – SAS Provider CA, Domain Proxy CA (with Operator Role OID), Professional Installer CA, CBSD Manufacturer CA
  - Sub-CA signed by higher order Sub-CA (CBSD Manufacturer CA): CBSD OEM CA
  - End Entity signed by respective Sub-CA – SAS Provider, Domain Proxy, Installer, CBSD OEM (signed by CBSD OEM CA), CBSD (signed by CBSD Manufacturer CA)
- PowerShell_CACreationScript.ps1 – this is Windows cert generation script. This script generates the same set of "test" certificates as CACreationScript.sh.

Note: Scripts in [i.1] generate only RSA certificates. If a CBSD vendor implementation uses ECC certificates as per profile defined in [n.8], then CBSD vendor can either enhance these scripts or develop new scripts/methods for the same.

Following options are provided for key-pair to be used for "test" certificate generation:

- CA, Sub-CA certificates: Auto-generated key-pair is used. The script first uses "openssl" command to auto-generate the key pair and then uses the same to generate the "test" certificate.
- Installer end-entity certificate: Auto-generated key-pair is used.
- CBSD OEM, CBSD, DP and SAS provider end-entity certificate: 2 options are provided:
  - Auto-generated key pair option is the default

  - CBSD vendor can provide the key-pair on the script command-line and the same would be used to generate the CBSD and CBSD OEM certificates
  - For DP entity certificate, CBSD vendor can provide the key-pair on the script command-line and the same would be used to generate the certificate
  - SAS vendor can provide the key-pair on the script command line and the same would be used to generate the SAS Provider certificate

In the "test" certificates, WInnForum Poison extension (TEST OID 1.3.6.1.4.1.46609.1.999) is present as a critical extension with ASN.1 encoded NULL data (0x05 0x00)

- Presence of TEST OID in the "test" cert allows a production / deployed CBRS peer entity to reject the TLS connection when presented with "test" certificate
- A peer entity, not conforming to WInnForum CBRS PKI specification, shall reject the TLS connection request since it would not be able to identify the TEST OID critical extension

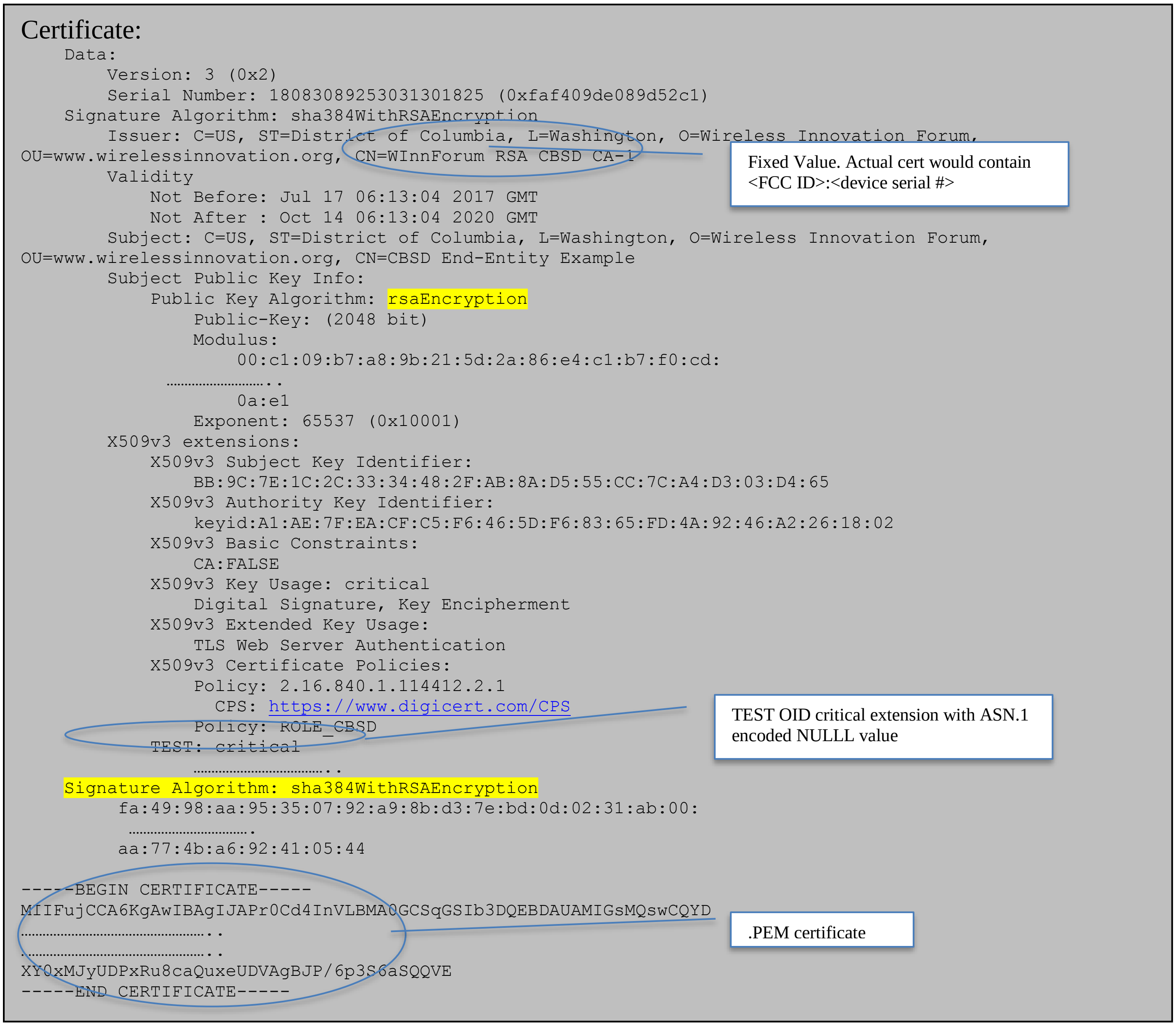
The "test" certificates for CBRS PKI entities are ".pem" format and follow the RSA certificate profile as per [n.8] with the following exceptions:

- X.509v3 Certificate policy ID value references "https://www.digicert.com/CPS" – in future it would reference WInnForum CBRS PKI CP. However, this does not impact any of the CBSD certification test case.
- Some of the fields in Subject DN are fixed (through configuration file) – example: State, OU, CN fields. These fixed values do not impact any of the CBSD certification test case.
- CBSD / CBSD OEM end entity certificate CN field does not contain the Device-id (of the form <FCC ID>:<device serial number>) specified in RSA subscriber certificate profile in [n.8]. Instead, CN has a fixed value string, in the absence of availability of FCC-id for unit under certification test.
- Domain Proxy end entity certificate CN field does not contain the Device-id (of the form <FRN>:<unique ID>) specified in RSA subscriber certificate profile in [n.8]. Instead, CN has a fixed value string, in the absence of availability of FRN for DP under certification test.
- SAS end entity certificate CN field does not contain the Device-id (of the form <FQDN of SAS Provider Server>) specified in RSA subscriber certificate profile in [n.8]. Instead, CN has a fixed value string.

Note: Generated certificates do not contain some optional and non-critical fields like Subject Alt Name, CRL Distribution Point (so no CRL or OSCP server links to check the certificate revocation status).

Each execution of the CACreationScript.sh generates certificates having a new certificate serial number, validity duration (starting with current system time), key-pair (auto-generated case only) and the digital signature though the rest of the fields including Subject DN and CN remain the same.

Following is a sample of CBSD End entity “test” certificate generated by the CACreationScript.sh script and openssl.conf file:



```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 18083089253031301825 (0xfaf409de089d52c1)
    Signature Algorithm: sha384WithRSAEncryption
        Issuer: C=US, ST=District of Columbia, L=Washington, O=Wireless Innovation Forum,
OU=www.wirelessinnovation.org, CN=WInnForum RSA CBSD CA-1
        Validity
            Not Before: Jul 17 06:13:04 2017 GMT
            Not After : Oct 14 06:13:04 2020 GMT
        Subject: C=US, ST=District of Columbia, L=Washington, O=Wireless Innovation Forum,
OU=www.wirelessinnovation.org, CN=CBSD End-Entity Example
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c1:09:b7:a8:9b:21:5d:2a:86:e4:c1:b7:f0:cd:
             ……………………..
                    0a:e1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                BB:9C:7E:1C:2C:33:34:48:2F:AB:8A:D5:55:CC:7C:A4:D3:03:D4:65
            X509v3 Authority Key Identifier:
                keyid:A1:AE:7F:EA:CF:C5:F6:46:5D:F6:83:65:FD:4A:92:46:A2:26:18:02
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 Certificate Policies:
                Policy: 2.16.840.1.114412.2.1
                  CPS: https://www.digicert.com/CPS
                Policy: ROLE_CBSD
            TEST: critical
                ……………………………..
    Signature Algorithm: sha384WithRSAEncryption
         fa:49:98:aa:95:35:07:92:a9:8b:d3:7e:bd:0d:02:31:ab:00:
         …………………………….
         aa:77:4b:a6:92:41:05:44

-----BEGIN CERTIFICATE-----
MIIFujCCA6KgAwIBAgIJAPr0Cd4InVLBMA0GCSqGSIb3DQEBDAUAMIGsMQswCQYD
……………………………………………..
……………………………………………..
XY0xMJyUDPxRu8caQuxeUDVAgBJP/6p3S6aSQQVE
-----END CERTIFICATE-----
```

Fixed Value. Actual cert would contain <FCC ID>:<device serial #>

TEST OID critical extension with ASN.1 encoded NULLL value

.PEM certificate

6.8.3.2.2 Test Certificate Use

UUT and the simulated entities in the Test Harness setup a secure connection using the "Test" certificates for mutual authentication.

"Test" certificates generated as per [i.1] or different method/scripts need to be installed in CBSD/DP and mockSAS before use. Following are some key points before the use of the "test" certificates:

- Complete certificate chain from end entity certificate to the Root CA certificate needs to be installed at CBSD, DP (if used) and at mockSAS.
- CBSD vendor needs to install only one of the CBSD EE or the CBSD OEM EE certificates. If CBSD vendor uses CBSD OEM EE certificate, then CBSD vendor also needs to install the CBSD OEM CA certificate.
- CBSD Vendor & Tester needs to ensure that the same Root CA certificate part of the trust chain of the CBSD/DP/SAS entity certificate is installed at CBSD, DP and mockSAS.
- If CBSD vendor implementation requires verification of SAS Root CA certificate hash, then CBSD vendor needs to calculate the hash of above generated Root CA certificate and store that as well at CBSD / DP.

Note: Installation of the CBSD, DP certificates is specific to Vendor implementation.

Although "test" certificates generated as per [i.1] are valid certificates, still the mockSAS should perform the verification of "test" certificate from CBSD/DP and reject the TLS connection, logging an error, if any of these checks fail. CBSD/DP shall always verify the mockSAS "test" certificate.

Verification of the "Test" certificates by entities involved in secure connection comprises following:

- Verification of the validity period of the certificate
- Verification of the Digital signature of the certificate – this is done through following steps:
  - Calculate the HMAC of the entity certificate using signature algorithm indicated in the certificate
  - Decrypt the signature (HMAC) using the public key contained in the certificate
  - Compare the HMAC calculated above with the decrypted HMAC to verify both the integrity of the certificate and that it belongs to the peer entity owning the private key of the key-pair.
- CBSD, DP and mockSAS need to verify the complete certificate chain from the peer entity. CBSD, DP and mockSAS also need to verify the Root CA certificate or its hash is same as stored / installed at its end.
- Verification of the presence of TEST OID critical extension. A test unit should check for the presence of the TEST OID in the peer certificate to ensure that peer entity is part of the test ecosystem. (**Note**: A production unit would treat TEST OID as unrecognized critical extension OID or as a tagging a "test" certificate. In either case,

the production unit would reject the peer certificate thus preventing interaction with a test unit).

- CBSD, DP may optionally verify the SAS Role present in the certificate.

Some certificate verification checks, that a production CBSD, DP or SAS performs cannot be done on the "test" certificates. UUT, working with "test" certificates, need to bypass these checks and not fail peer authentication. Following are the checks which need to be bypassed:

- Verification of the *device-id* value (eg: <FCC ID>:<CBSD SN>, <FRN>:<Unique ID> etc.) in Subject DN -> CN field.
- Verification of certificate revocation status

Test certificates for verification of CBSD behavior due to invalid mock-SAS certificate: CBSD is required to check that the mock-SAS certificate is signed by a WInnForum approved CBRS PKI Root CA. CBSD also needs to verify that the duration of the mock-SAS certificate is also valid as described above. To enable verification of such behavior at CBSD, corrupted, invalid and different Root CA signed certificates are placed in same directory in WInnForum GitHub repository as the test scripts for security validation. When executing these test cases (described in the following sections), tester needs to install these corrupt certificates at mock-SAS.

6.8.4 Test Procedure

6.8.4.1 Successful TLS connection

In all test cases under this category, the TLS connection is established successfully between mock-SAS and CBSD. The security procedure is irrespective of the procedures defined for mock-SAS-CBSD communication.
Certificates at CBSD are correct and valid.

6.8.4.1.1 [WINNF.FT.C.SCS.1] Successful TLS connection between CBSD and mock-SAS

The following are the test execution steps.

#	Test Execution Steps	Results	
1	• Configure CBSD to start CBSD-SAS communication to start the security procedure • The CBSD should establish a TLS handshake with the mock-SAS using configured certificate. • Configure mock-SAS to accept the security procedure and establish the connection	PASS	FAIL
2	• Make sure that Mutual authentication happens between CBSD and mock-SAS. • Make sure that CBSD used TLS v1.2 • Make sure that cipher suites from one of the following is selected, • TLS_RSA_WITH_AES_128_GCM_SHA256 • TLS_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	PASS	FAIL

	- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256		
3	- Make sure that application data is successfully sent and corresponding response is received from mock-SAS.	PASS	FAIL
4	- Verify using a spectrum analyzer that CBSD is not transmitting if no authorized grant is present with the CBSD at any instance	PASS	FAIL

6.8.4.1.2 [WINNF.FT.D.SCS.2] Successful TLS connection between Domain-proxy and mock-SAS – Domain Proxy

The following steps describe the test execution.

#	Test Execution Steps	**Results**	
1	- The Domain-proxy should establish a TLS handshake with the mock-SAS using configured certificate. - Configure Domain-proxy to start Domain-proxy -SAS communication to start the security procedure - Configure mock-SAS to accept the security procedure and establish the connection	PASS	FAIL
2	- Make sure that Mutual authentication happens between Domain-proxy and mock-SAS. - Make sure that CBSD used TLS v1.2 - Make sure that cipher suites from one of the following is selected, - TLS_RSA_WITH_AES_128_GCM_SHA256 - TLS_RSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	PASS	FAIL
3	- Make sure that application data is successfully sent and corresponding response is received from mock-SAS.	PASS	FAIL
4	- Verify using spectrum analyzers that none of the CBSD shall transmit until an authorized grant is available for the CBSD	PASS	FAIL

6.8.4.2 Unsuccessful TLS connection

In all test cases under this category, the TLS connection is not established successfully between mock-SAS and CBSD. The security procedure is irrespective of the procedures defined for mock-SAS-CBSD communication.

6.8.4.2.1 [WINNF.FT.C.SCS.3] TLS failure due to revoked certificate

Test case pre-requisite:

- The certificate at the mock-SAS shall be marked as revoked

The following are the test execution steps.

#	Test Execution Steps	Results	
1	• Configure CBSD to start CBSD-SAS communication to start the security procedures	PASS	FAIL
2	• Make sure that CBSD used TLS v1.2 for security establishment. • Make sure CBSD selects the correct cipher suite. • CBSD may use CRL or OCSP to verify the validity of the server certificate. • Make sure that Mutual authentication does not happen between CBSD and mock-SAS.	PASS	FAIL
3	CBSD may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	PASS	FAIL
4	• Verify using a spectrum analyzer that CBSD is not transmitting if no authorized grant is present with the CBSD at any instance	PASS	FAIL

6.8.4.2.2 [WINNF.FT.C.SCS.4] TLS failure due to expired server certificate

Test case pre-requisite:

- Configure mock-SAS such that server certificate is valid but expired.

The following are the test execution steps.

#	Test Execution Steps	Results	
1	• Configure CBSD to start CBSD-SAS communication to start the security procedures	PASS	FAIL
2	• Make sure that CBSD used TLS v1.2 for security establishment. • Make sure CBSD selects the correct cipher suite. • Make sure that Mutual authentication does not happen between CBSD and mock-SAS.	PASS	FAIL
3	CBSD may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	PASS	FAIL
4	• Verify using a spectrum analyzer that CBSD is not transmitting if no authorized grant is present with the CBSD at any instance	PASS	FAIL

6.8.4.2.3 [WINNF.FT.C.SCS.5] TLS failure when mock-SAS certificate is issue by an unknown CA

Test case pre-requisite:

- Equip the mock-SAS with certificate signed a valid CA but not signed with the same Root CA certificate as the one signing the CBSD certificate chain.

The following are the test execution steps.

#	**Test Execution Steps**	**Results**	
1	• Configure CBSD to start CBSD-SAS communication to start the security procedures	PASS	FAIL
2	• Make sure that CBSD used TLS v1.2 for security establishment. • Make sure CBSD selects the correct cipher suite. • Make sure that Mutual authentication does not happen between CBSD and mock-SAS.	PASS	FAIL
3	CBSD may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	PASS	FAIL
4	• Verify using a spectrum analyzer that CBSD is not transmitting if no authorized grant is present with the CBSD at any instance	PASS	FAIL

6.8.4.2.4 [WINNF.FT.C.SCS.6] TLS failure when certificate at mock-SAS is corrupted

Test case pre-requisite:

- The end-entity certificate at the mock-SAS shall be corrupted

The following steps describe the test execution.

#	Test Execution Steps	**Results**	
1	• Equip the mock-SAS with a corrupt certificate and CBSD with valid end-entity certificates signed with CBRS PKI (or test PKI). • Configure CBSD to start CBSD-SAS communication to start the security procedures	SUC CESS	FAIL
2	• Make sure that CBSD used TLS v1.2 for security establishment. • Make sure CBSD selects the correct cipher suite. • Make sure that Mutual authentication does not happen between CBSD and mock-SAS.	SUC CESS	FAIL
3	CBSD may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	SUC CESS	FAIL
4	• Verify using spectrum analyzers that the CBSD shall transmit until an authorized grant is available for the CBSD	SUC CESS	FAIL

6.8.4.2.5 [WINNF.FT.D.SCS.7] TLS failure due to revoked certificate – Domain Proxy

Test case pre-requisite:

- The certificate at the mock-SAS shall be marked as revoked

The following steps describe the test execution.

#	Test Execution Steps	**Results**

1	• Configure Domain-proxy to start Domain-proxy -SAS communication to start the security procedures	PASS	FAIL
2	• Make sure that Domain-proxy used TLS v1.2 for security establishment. • Make sure Domain-proxy selects the correct cipher suite. • CBSD may use CRL or OCSP to verify the validity of the server certificate. • Make sure that Mutual authentication does not happen between Domain-proxy and mock-SAS.	PASS	FAIL
3	Domain-proxy may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	PASS	FAIL
4	• Verify using spectrum analyzers that none of the CBSD shall transmit until an authorized grant is available for the CBSD	PASS	FAIL

6.8.4.2.6 [WINNF.FT.D.SCS.8] TLS failure due to expired server certificate – Domain Proxy

Test case pre-requisite:

- Configure mock-SAS such that server certificate is valid but expired.

The following steps describe the test execution.

#	Test Execution Steps	Results	
1	• Configure Domain-proxy to start Domain-proxy -SAS communication to start the security procedures	PASS	FAIL
2	• Make sure that Domain-proxy used TLS v1.2 for security establishment. • Make sure Domain-proxy selects the correct cipher suite. • Make sure that Mutual authentication does not happen between Domain-proxy and mock-SAS.	PASS	FAIL
3	Domain-proxy may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	PASS	FAIL
4	• Verify using spectrum analyzers that none of the CBSD shall transmit until an authorized grant is available for the CBSD	PASS	FAIL

6.8.4.2.7 [WINNF.FT.D.SCS.9] TLS failure when mock-SAS certificate is issue by unknown CA – Domain Proxy

Test case pre-requisite:

- Equip the mock-SAS with certificate signed a valid CA but not signed with the same Root CA certificate as the one signing the CBSD certificate chain.

The following steps describe the test execution.

#	Test Execution Steps	**Results**	
1	• Equip the mock-SAS with certificate signed a valid CA but not in the CBRS PKI. • Equip the Domain-proxy with valid end-entity certificates signed with CBRS PKI (or test PKI). • Configure Domain-proxy to start Domain-proxy -SAS communication to start the security procedures	PASS	FAIL
2	• Make sure that Domain-proxy used TLS v1.2 for security establishment. • Make sure Domain-proxy selects the correct cipher suite. • Make sure that Mutual authentication does not happen between Domain-proxy and mock-SAS.	PASS	FAIL
3	Domain-proxy may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	PASS	FAIL
4	• Verify using spectrum analyzers that none of the CBSD shall transmit until an authorized grant is available for the CBSD	PASS	FAIL

6.8.4.2.8 [WINNF.FT.D.SCS.10] TLS failure when certificate at mock-SAS is corrupted – Domain Proxy

Test case pre-requisite:

- The end-entity certificate at the mock-SAS shall be corrupted

The following steps describe the test execution.

#	Test Execution Steps	**Results**	
1	• Equip the mock-SAS with a corrupt certificate and Domain proxy with valid end-entity certificates signed with CBRS PKI (or test PKI). • Configure DP to start DP-SAS communication to start the security procedures	SUC CESS	FAIL
2	• Make sure that DP used TLS v1.2 for security establishment. • Make sure DP selects the correct cipher suite. • Make sure that Mutual authentication does not happen between DP and mock-SAS.	SUC CESS	FAIL
3	DP may retry for the security procedure which shall fail again till external measures are taken to ensure the certificates are valid at both end.	SUC CESS	FAIL
4	• Verify using spectrum analyzers that none of the CBSD shall transmit until an authorized grant is available for the CBSD	SUC CESS	FAIL

7 History

Document history		
V0.0.4	June 19, 2017	Skeleton of Technical Report. Merged test case contributions: Registration, Spectrum Inquiry, Spectrum Inquiry by DP, Spectrum Grant, Spectrum Grant by DP, Heartbeat, Grant Relinquishment, Grant Relinquishment by DP, Deregistration
V0.05	July 11, 2017	Merged Measurement Report and Test Equipment Requirement contributions
V0.0.5 (IR1)	July 19, 2017	Updated all references in document, fixed TC numbering for HBT and MES sections
V0.0.5 (IR2)	July 21, 2017	Fixed section numbers, TC numbering issues. Updated REL, RLQ sections
V0.0.5 (IR3)	July 26, 2017	Changes to Security Validation Test cases. Addition of section 6.8.3.2. Removal of optional security validation test cases
V0.0.5 (IR4)	July 27, 2017	Some editorial work. Minor updates to HBT and MES sections.
V0.0.5 (IR5)		Updates to RLQ sections
V0.0.5 (IR6)	Aug 17, 2017	Editorial work, minor updates
V0.0.5(IR7)	Aug 22,2017	Add Test Case ID Table for certification