ONE WORLD ○ OUR APPROVAL

Wireless test report – 370566-2TRFWL

Applicant:

JMA Wireless (Teko Telecom Srl)

Product name:

XRAN

Model:

N/A*

FCC ID:

N/A*

Specifications:

WINNF-TS-0122, Version V1.0.1 – Domain Proxy requirements

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

Note: * - Domain proxy is a software product

Date of issue: June 25, 2019

Test engineer(s):	Andrey Adelberg, Senior Wireless/EMC Specialist	Signature:	Andrey Adelberg
Reviewed by:	Tom Tidwell	Signature:	

Test location(s)

Company name	Nemko Canada Inc.
Address	303 River Road
City	Ottawa
Province	Ontario
Postal code	K1V 1H2
Country	Canada
Telephone	+1 613 737 9680
Facsimile	+1 613 737 9691
Toll free	+1 800 563 6336
Website	www.nemko.com
Site number	FCC: CA2040; IC: 2040A-4 (3 m SAC)

Limits of responsibility

Note that the results contained in this report relate only to the items tested and were obtained in the period between the date of initial receipt of samples and the date of issue of the report.
This test report has been completed in accordance with the requirements of ISO/IEC 17025. All results contain in this report are within Nemko Canada's ISO/IEC 17025 accreditation.

Copyright notification

Table of contents

Section 1. Report summary

1.1 Applicant and manufacturer

Company name	JMA Wireless (Teko Telecom Srl)
Address	Via Antonio Meucci, 24
City	Castel San Pietro Terme
Province/State	BO
Postal/Zip code	40024
Country	Italy

1.2 Test specifications

WINNF-TS-0122 Version V1.0.1 (28 September 2018)	Test and Certification for Citizens Broadband Radio Service (CBRS); Conformance and Performance Test Technical Specification; CBSD/DP as Unit Under Test (UUT)
FCC 47 CFR Part 96	Citizens Broadband Radio Service
WINNF-TS-0016 Version V1.2.3	Signaling Protocols and Procedures for Citizens Broadband Radio Service (CBRS): Spectrum Access System (SAS) - Citizens Broadband Radio Service Device (CBSD) Interface Technical Specification

1.3 Statement of compliance

In the configuration tested, the EUT was found compliant.

Testing was performed against all relevant requirements of the test standard except as noted in section 1.5 below. Results obtained indicate that the product under test complies in full with the requirements tested. The test results relate only to the items tested.

See "Summary of test results" for full details.

1.4 Exclusions

This test report covers only Domain Proxy requirements.

1.5 Test report revision history

Revision #	Date of issue	Details of changes made to test report
TRF	June 25, 2019	Original report issued

Section 2. Summary of test results

2.1 WINNF-TS-0122 requirements test results

Table 2.1-1: *Domain Proxy requirements results*

Section	Test description	Verdict
6.1.4.1.2	Domain Proxy Multi-Step registration	Pass
6.1.4.1.4	Domain Proxy Single-Step registration for Cat A CBSD	Pass
6.1.4.2.2	Domain Proxy Missing Required parameters (responseCode 102)	Pass
6.1.4.2.4	Domain Proxy Pending registration (responseCode 200)	Pass
6.1.4.2.6	Domain Proxy Invalid parameters (responseCode 103)	Pass
6.1.4.2.8	Domain Proxy Blacklisted CBSD (responseCode 101)	Pass
6.1.4.2.10	Domain Proxy Unsupported SAS protocol version responseCode 100)	Pass
6.1.4.2.12	Domain Proxy Group Error (responseCode 201)	Pass
6.4.4.1.2	Domain Proxy Heartbeat Success Case (first Heartbeat Response)	Pass
6.4.4.2.6	Domain Proxy Heartbeat responseCode=500 (TEMINATED_GRANT)	Pass
6.5.4.2.2	Domain Proxy Registration Response contains measReportConfig	Pass
6.5.4.2.5	Domain Proxy Heartbeat Response contains measReportConfig	Pass
6.6.4.1.2	Domain Proxy Successful Relinquishment	Pass
6.6.4.2.2	Domain Proxy Unsuccessful Relinquishment, responseCode=102	Pass
6.6.4.3.2	Domain Proxy Unsuccessful Relinquishment, responseCode=103	Pass
6.7.4.1.2	Domain Proxy Successful Deregistration	Pass
6.7.4.2.2	Domain Proxy Deregistration responseCode=102	Pass

Notes: none

Section 3. Equipment under test (EUT) details

3.1 Sample information

Receipt date	May 30, 2019
Nemko sample ID number	1, 2 and 3

3.2 EUT information

Product name	XRAN
CPE RF card model	XR19AX35WM2/48Y
Base Station model	THWPC-R-XT2AC
Serial numbers	1012482003, 1012482006
Revision number	1.0
Harness software version	1.0.0.3
Software version	V1

3.3 Technical information

Frequency band	CBRS band: 3550–3700 MHz
Type of modulation	QPSK½ to 64QAM
Power requirements	48 V_{DC} via PoE powered from 120 V_{AC} / 60 Hz

3.4 Product description and theory of operation

TEKO™ CellHub Distributed Radio System

- IT-centric offering for enterprise, in-building, venue and outdoor densification mobile connectivity
- Multiple operators/spectrum, including CBRS in single device
- Directly connects to a standard server locally or data center up to 12 miles away
- Domain proxy support to CBRS SAS

The TEKO CellHub is a JMA Wireless radio unit that supports high capacity, multi-channel CBRS (3550-3700 MHz, FCC Part 96) bands with, or without, simultaneous licensed carrier cellular bands. CellHubs support LTE for CBRS and licensed bands. As part of the JMA Wireless XRAN system it has the option of working in conjunction with new or existing distributed antenna systems.

Each CellHub acts as a CBSD under a Domain Proxy with compliance to the WInnForum SAS-CBSD interface, CBRS Alliance OnGo, and FCC Part 96 requirements. The Domain Proxy is provided as an independent software service module on the XRAN system. Multiple CellHubs can daisy chain miles apart to save on cabling and number of headend connections and homeruns.

3.5 EUT exercise details

SAS Installed and connected to Domain Proxy that acts on behalf of CBSDs. Spectrum analyzer connected to CBSDs' RF output.

3.6 EUT setup diagram

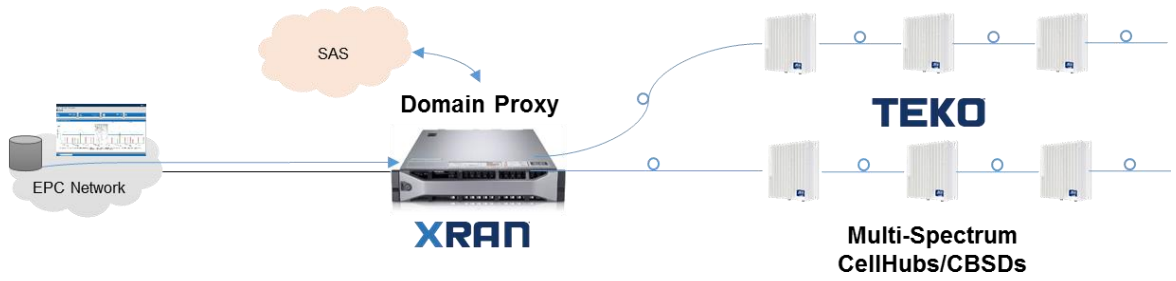

Figure 3.6-1: *Setup diagram*

3.7 EUT security per CBRS requirements

Requirement	Compliance
What communication protocol is used between the SAS and the CBSD?	The SAS-CBSD protocol is based on the HTTPS (HTTP over TLS version 1.2). The HTTPS protocol provides transport level assurance that a message has been received by the intended recipient. Communication includes mutual authentication using pki certificates.
How are communications initiated?	Per standard specification, SAS server discovery: SAS server URL is provided to CBSD's. CBSD via domain proxy communicate to server per URL provided and TLS mutual authentication will be performed. The CBSD/Domain Proxy initiating the TLS connection shall authenticate the SAS, and the SAS shall authenticate the CBSD/Domain Proxy.
How does the CBSD validate messages from the SAS?	Each massage session is encrypted and validated with TLSv1.2 and CA certificates verification. Messages also checked against protocol structure json.
How does the device handle failure to communicate or authenticate the SAS?	On communication failure/authentication, devices we re-try to communicate if fails, alarm will raise, and TX will stop.
How does the SAS validate messages from a CBSD?	Each massage session is encrypted and validated with TLSv1.2 and CA certificates verification. Messages also checked against protocol structure json.
What encryption method is used?	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
How does the SAS ensure secure registration of protected devices?	By using user name and ID, also CPI signature can be usded.

Note: Protocols in accordance with: Document WINNF-TS-0016 Version V1.2.3 from October 31st, 2018

Section 4. Engineering considerations

4.1 Modifications incorporated in the EUT

There were no modifications performed to the EUT during this assessment.

4.2 Technical judgment

None

4.3 Deviations from laboratory tests procedures

No deviations were made from laboratory procedures.

Section 5. Test conditions

5.1 Atmospheric conditions

Temperature	15–30 °C
Relative humidity	20–75 %
Air pressure	860–1060 mbar

When it is impracticable to carry out tests under these conditions, a note to this effect stating the ambient temperature and relative humidity during the tests shall be recorded and stated.

5.2 Power supply range

The normal test voltage for equipment to be connected to the mains shall be the nominal mains voltage. For the purpose of the present document, the nominal voltage shall be the declared voltage, or any of the declared voltages ±5 %, for which the equipment was designed.

Section 6. Measurement uncertainty

6.1 Uncertainty of measurement

UKAS Lab 34 and TIA-603-B have been used as guidance for measurement uncertainty reasonable estimations with regards to previous experience and validation of data. Nemko Canada, Inc. follows these test methods in order to satisfy ISO/IEC 17025 requirements for estimation of uncertainty of measurement for wireless products.

Measurement uncertainty budgets for the tests are detailed below. Measurement uncertainty calculations assume a coverage factor of K = 2 with 95% certainty.

Table 6.1-1: *Measurement uncertainty*

Test name	Measurement uncertainty, dB
All antenna port measurements	0.55

Section 7. Test equipment

7.1 Test equipment list

Table 7.1-1: *Equipment list*

Equipment	Manufacturer	Model no.	Asset no.	Cal cycle	Next cal.
Spectrum analyzer	Rohde & Schwarz	FSU	FA001877	1 year	October 26, 2019

Section 8. Testing data

8.1 6.1.4.1.2 [WINNF.FT.D.REG.2] Domain Proxy Multi-Step registration

8.1.1 Definitions and limits

6.1 CBSD Registration Process
This section provides test steps, conditions and procedures to test the conformance of the CBSD implementation for the CBSD Registration Procedure. A precondition is the CBSD has successfully discovered the SAS it wants to register with.
This test is mandatory for the Domain proxy that is controlling CBSDs which support multi-step registration. 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.

8.1.2 Test date

Start date	May 28, 2019

8.1.3 Observations, settings and special notes

None

8.1.4 Test data

Table 8.1-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	DP with two CBSD sends correct Registration request information, as specified in [n.5], in the form of one 2-element Array or as individual messages to the SAS Test Harness: • The required userId, fccId and cbsdSerialNumber registration parameters shall be sent for each CBSD and conform to proper format and acceptable ranges. • Any REG-conditional or optional registration parameters that may be included in the message shall be verified that they conform to proper format and are within acceptable ranges.	☒	☐
3	• SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or individual messages as follows: ○ cbsdId = Ci ○ measReportConfig shall not be included ○ responseCode = 0 for each CBSD	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.2 6.1.4.1.4 [WINNF.FT.D.REG.4] Domain Proxy Single-Step registration for Cat A CBSD

8.2.1 Definitions and limits

6.1 CBSD Registration Process
This section provides test steps, conditions and procedures to test the conformance of the CBSD implementation for the CBSD Registration Procedure. A precondition is the CBSD has successfully discovered the SAS it wants to register with.
This test is mandatory for DP connected to CBSDs which report all Required and REG-Conditional parameters in the Registration request to the SAS, without 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.
For a Category A CBSD which determine own location, the test lab and vendor must agree on the required evidence showing the UUT meets the location requirement. In lieu of location verification, the vendor shall supply their test approach/procedure along with compliance data.

8.2.2 Test date

Start date	May 28, 2019

8.2.3 Observations, settings and special notes

None

8.2.4 Test data

Table 8.2-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	The DP with two CBSDs sends Registration requests in the form of one 2-element Array or as individual messages to SAS Test Harness. • The required userId, fccId and cbsdSerialNumber and REG-Conditional cbsdCategory, airInterface, installationParam, and measCapability registration parameters shall be sent from the CBSD and conform to proper format and acceptable ranges. • Any optional registration parameters that may be included in the message shall be verified that they conform to proper format and are within acceptable ranges.	☒	☐
3	• SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or individual messages as follows: o cbsdId = Ci o measReportConfig shall not be included o responseCode = 0 for each CBSD	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.3 6.1.4.2.2 [WINNF.FT.D.REG.9] Domain Proxy Missing Required parameters (responseCode 102)

8.3.1 Definitions and limits

6.1 CBSD Registration Process
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 valid registrationRequest message with a registrationResponse with a non-zero responseCode.
The purpose of these tests is to ensure that the CBSD does not transmit when a responseCode other than 0 is received. The information sent in the registration request message is not important, only that it shall conform to the protocol.
Missing/Invalid response codes are tested by injecting those responseCodes into the SAS Test Harness generated response message, even though the UUT has sent a valid message
This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps where the Registration response contains responseCode (Ri) = 102 for each CBSD

8.3.2 Test date

Start date	May 28, 2019

8.3.3 Observations, settings and special notes

None

8.3.4 Test data

Table 8.3-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	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.	–	–
3	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: o SAS response does not include a cbsdId. o responseCode = Ri for CBSD1 and CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.4 6.1.4.2.4 [WINNF.FT.D.REG.11] Domain Proxy Pending registration (responseCode 200)

8.4.1 Definitions and limits

6.1 CBSD Registration Process
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 valid registrationRequest message with a registrationResponse with a non-zero responseCode.
The purpose of these tests is to ensure that the CBSD does not transmit when a responseCode other than 0 is received. The information sent in the registration request message is not important, only that it shall conform to the protocol.
Missing/Invalid response codes are tested by injecting those responseCodes into the SAS Test Harness generated response message, even though the UUT has sent a valid message
The same steps provided for WINNF.FT.D.REG.9 shall be executed for this test, with the exception that the Registration response contains responseCode (Ri) = 200 for each CBSD.

8.4.2 Test date

Start date	May 28, 2019

8.4.3 Observations, settings and special notes

None

8.4.4 Test data

Table 8.4-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	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.	–	–
3	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: o SAS response does not include a cbsdId. o responseCode (Ri) = 200 for CBSD1 and CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.5 6.1.4.2.6 [WINNF.FT.D.REG.13] Domain Proxy Invalid parameters (responseCode 103)

8.5.1 Definitions and limits

6.1 CBSD Registration Process
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 valid registrationRequest message with a registrationResponse with a non-zero responseCode.
The purpose of these tests is to ensure that the CBSD does not transmit when a responseCode other than 0 is received. The information sent in the registration request message is not important, only that it shall conform to the protocol.
Missing/Invalid response codes are tested by injecting those responseCodes into the SAS Test Harness generated response message, even though the UUT has sent a valid message
The same steps provided for WINNF.FT.D.REG.9 shall be executed for this test, with the exception that the Registration response contains responseCode R1 = 0 for CBSD1 and R2 = 103 for CBSD2.

8.5.2 Test date

Start date	May 28, 2019

8.5.3 Observations, settings and special notes

None

8.5.4 Test data

Table 8.5-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	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.	–	–
3	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: o SAS response does not include a cbsdId. o responseCode (R1) = 0 for CBSD1 o responseCode (R2) = 103 for CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.6 6.1.4.2.8 [WINNF.FT.D.REG.15] Domain Proxy Blacklisted CBSD (responseCode 101)

8.6.1 Definitions and limits

6.1 CBSD Registration Process
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 valid registrationRequest message with a registrationResponse with a non-zero responseCode.
The purpose of these tests is to ensure that the CBSD does not transmit when a responseCode other than 0 is received. The information sent in the registration request message is not important, only that it shall conform to the protocol.
Missing/Invalid response codes are tested by injecting those responseCodes into the SAS Test Harness generated response message, even though the UUT has sent a valid message
The same steps provided for WINNF.FT.D.REG.9 shall be executed for this test, with the exception that the Registration response contains responseCode R1 = 0 for CBSD1 and R2 = 101 for CBSD2.

8.6.2 Test date

Start date	May 28, 2019

8.6.3 Observations, settings and special notes

None

8.6.4 Test data

Table 8.6-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	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.	–	–
3	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: o SAS response does not include a cbsdId. o responseCode (R1) = 0 for CBSD1 o responseCode (R2) = 101 for CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.7 6.1.4.2.10 [WINNF.FT.D.REG.17] Domain Proxy Unsupported SAS protocol version (responseCode 100)

8.7.1 Definitions and limits

6.1 CBSD Registration Process
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 valid registrationRequest message with a registrationResponse with a non-zero responseCode.
The purpose of these tests is to ensure that the CBSD does not transmit when a responseCode other than 0 is received. The information sent in the registration request message is not important, only that it shall conform to the protocol.
Missing/Invalid response codes are tested by injecting those responseCodes into the SAS Test Harness generated response message, even though the UUT has sent a valid message
The same steps provided for WINNF.FT.D.REG.9 shall be executed for this test, with the exception that the Registration response contains responseCode (Ri) = 100 for each CBSD.

8.7.2 Test date

Start date	May 28, 2019

8.7.3 Observations, settings and special notes

None

8.7.4 Test data

Table 8.7-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	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.	–	–
3	SAS Test Harness sends a CBSD Registration Response in the form of one 2-element Array or as individual messages as follows: o SAS response does not include a cbsdId. o responseCode (Ri) = 100 for CBSD1 and CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.8 6.1.4.2.12 [WINNF.FT.D.REG.19] Domain Proxy Group Error (responseCode 201)

8.8.1 Definitions and limits

6.1 CBSD Registration Process
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 valid registrationRequest message with a registrationResponse with a non-zero responseCode.
The purpose of these tests is to ensure that the CBSD does not transmit when a responseCode other than 0 is received. The information sent in the registration request message is not important, only that it shall conform to the protocol.
Missing/Invalid response codes are tested by injecting those responseCodes into the SAS Test Harness generated response message, even though the UUT has sent a valid message
The registrationRequest groupingParam is an optional field and will be validated by the SAS Test Harness if provided in the Registration Request message.
This test will validate that the CBSD will remain Unregistered after receiving responseCode 201.
The same steps provided for WINNF.FT.D.REG.9 shall be executed for this test, with the exception that the Registration response contains responseCode R1 = 0 for CBSD1 and R2 = 201 for CBSD2.

8.8.2 Test date

Start date	May 28, 2019

8.8.3 Observations, settings and special notes

None

8.8.4 Test data

Table 8.8-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • UUT has successfully completed SAS Discovery and Authentication with SAS Test Harness • UUT is in the Unregistered state	–	–
2	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.	–	–
3	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. ○ responseCode (R1) = 0 for CBSD1 ○ responseCode (R2) = 201 for CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 not transmit RF	☒	☐

For the test log please refer to Section 9 of this test report.

8.9 6.4.4.1.2 [WINNF.FT.D.HBT.2] Domain Proxy Heartbeat Success Case (first Heartbeat Response)

8.9.1 Definitions and limits

6.4 CBSD Heart Beat Process
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 test cases in this section test the success path for the Heartbeat process. The SAS Test Harness shall use a heartBeatInterval of 60 seconds, unless specifically provided in the test case.
This test case incorporates validation of successful Spectrum Inquiry messaging (if present) and successful Grant messaging into the Heartbeat Success case.
This test case applies to Domain Proxy supervising two CBSDs.

8.9.2 Test date

Start date	May 28, 2019

8.9.3 Observations, settings and special notes

None

8.9.4 Test data

Table 8.9-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has two CBSD registered successfully with SAS Test Harness, with cbsdId = Ci, i={1,2}	–	–
2	DP sends a message: • If message is a Spectrum Inquiry Request, go to step 3 • If message is a Grant Request, go to step 5	–	–
3	DP sends a Spectrum Inquiry Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Spectrum Inquiry Request message is formatted correctly for each CBSD, including for CBSDi, i={1,2}: • cbsdId = Ci • List of frequencyRange objects sent by DP are within the CBRS frequency range	☒	☐
4	If a separate Spectrum Inquiry Request message was sent for each CBSD, the SAS 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 containing a 2-object array (one per CBSD), the SAS Test Harness shall respond with a single Spectrum Inquiry Response message containing a 2-object array. Verify parameters for each CBSD within the Spectrum Inquiry Response message are as follows, for CBSDi, i={1,2}: • cbsdId = Ci • availableChannel is an array of availableChannel objects • responseCode = 0	–	–
5	DP sends a Grant Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Grant Request message is formatted correctly for each CBSD, including for CBSDi, i={1,2}: • cbsdId = C • maxEIRP is at or below the limit appropriate for CBSD category as defined by Part 96 • operationFrequencyRange, Fi, sent by UUT is a valid range within the CBRS band	☒	☐

Step	Test Execution Steps	Pass	Fail
6	If a separate Grant Request message was sent for each CBSD, the SAS Test Harness shall respond to each Grant Request message with a separate Grant Response message. If a single Grant Request message was sent containing a 2-object array (one per CBSD), the SAS Test Harness shall respond with a single Grant Response message containing a 2-object array. Verify parameters for each CBSD within the Grant Response message are as follows, for CBSDi, i={1,2}: • cbsdId = Ci • grantId = Gi = a valid grant ID • grantExpireTime = UTC time greater than duration of the test • responseCode = 0	–	–
7	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. Verify 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"	☒	☐
8	If a separate Heartbeat Request message was sent for each CBSD by the DP, the SAS 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 SAS Test Harness shall respond with a single Heartbeat Response message containing a 2-object array. Verify parameters for each CBSD within the Heartbeat Response message are as follows, for CBSDi: • cbsdId = Ci • grantId = Gi • transmitExpireTime = current UTC time + 200 seconds • responseCode = 0	–	–
9	For further Heartbeat Request messages sent from DP after completion of step 8, validate message is sent within latest specified heartbeatInterval for CBSDi: • cbsdId = Ci • grantId = Gi • operationState = "AUTHORIZED" and SAS 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	☒	☐
10	Monitor the RF output of the UUT from start of test until UUT transmission commences. Monitor the RF output of the UUT from start of test until RF transmission commences. Verify: • UUT does not transmit at any time prior to completion of the first heartbeat response • UUT transmits after step 8 is complete, and its transmission is limited to within the bandwidth range Fi.	☒	☐

For the test log please refer to Section 9 of this test report.

8.10 6.4.4.2.6 [WINNF.FT.D.HBT.8] Domain Proxy Heartbeat responseCode=500 (TERMINATED_GRANT)

8.10.1 Definitions and limits

6.4 CBSD Heart Beat Process
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 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 SAS Test Harness shall not allow any new Grant Request from the UUT to succeed.
This test case applies to Domain Proxy supervising two CBSDs.

8.10.2 Test date

Start date	May 28, 2019

8.10.3 Observations, settings and special notes

None

8.10.4 Test data

Table 8.10-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has two CBSD registered successfully with SAS 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 o grantExpireTime = UTC time greater than duration of the test • 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. Verify Heartbeat Request message is sent within latest specified heartbeatInterval, and is formatted correctly for each CBSD, including, for CBSDi i={1,2}: • cbsdId = Ci, i = {1,2} • grantId = Gi, i = {1,2} • operationState = "AUTHORIZED"	☒	☐

Step	Test Execution Steps	Pass	Fail
3	If separate Heartbeat Request message was sent for each CBSD by the DP, the SAS 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 SAS 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, SAS Test Harness shall not allow any further grants to the UUT. If CBSD sends further Heartbeat Request messages for CBSD1, SAS Test Harness shall respond with a Heartbeat Response message with parameters: • cbsdId = C1 • grantId = G1 • transmitExpireTime = current UTC time + 200 seconds • responseCode = 0 • Heartbeat Request message is within heartbeatInterval of previous Heartbeat Request message	–	–
5	Monitor the RF output of CBSD2. Verify: • CBSD2 shall stop transmission within bandwidth F2 within (T + 60 seconds) of completion of step 3	☒	☐

For the test log please refer to Section 9 of this test report.

8.11 6.5.4.2.2 [WINNF.FT.D.MES.2] Domain Proxy Registration Response contains measReportConfig

8.11.1 Definitions and limits

6.5 CBSD Measurement Report
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 subsequent Request message from the UUT).
Devices which support one measurement capability must satisfy the test cases mandatory for that measurement capability. Devices which support multiple measurement capabilities must satisfy the test cases mandatory for each type of supported measurement capability.
This test case is mandatory for Domain Proxy supervising CBSD which support RECEIVED_POWER_WITHOUT_GRANT.

8.11.2 Test date

Start date	May 28, 2019

8.11.3 Observations, settings and special notes

None

8.11.4 Test data

Table 8.11-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with SAS 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. Verify 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"	☒	☐
3	If a separate Registration Request message was sent for each CBSD by the DP, the SAS 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 SAS 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. Verify 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.	☒	☐

Step	Test Execution Steps	Pass	Fail
6	If a separate Spectrum Inquiry Request message was sent for each CBSD by the DP, the SAS 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 SAS 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. Verify the Grant 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*.	☒	☐

For the test log please refer to Section 9 of this test report.

8.12 6.5.4.2.5 [WINNF.FT.D.MES.5] Domain Proxy Heartbeat Response contains measReportConfig

8.12.1 Definitions and limits

6.5 CBSD Measurement Report
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 subsequent Request message from the UUT).
Devices which support one measurement capability must satisfy the test cases mandatory for that measurement capability. Devices which support multiple measurement capabilities must satisfy the test cases mandatory for each type of supported measurement capability.
This test case is mandatory for Domain Proxy supervising CBSD which support RECEIVED_POWER_WITH_GRANT measurement reports.

8.12.2 Test date

Start date	May 28, 2019

8.12.3 Observations, settings and special notes

None

8.12.4 Test data

Table 8.12-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with SAS Test Harness • DP has successfully registered 2 CBSD with SAS 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	Verify 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. Verify Heartbeat Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • cbsdId = Ci • grantId = Gi • operationState = "AUTHORIZED"	☒	☐
3	If a separate Heartbeat Request message was sent for each CBSD by the DP, the SAS 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 SAS 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	–	–

Step	Test Execution Steps	Pass	Fail
4	Verify 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. Verify 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 o	☒	☐
5	If Heartbeat Request message (step 4) contains measReport object, then: • Verify 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	☒	☐
6	If a separate Heartbeat Request message was sent for each CBSD by the DP, the SAS 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 SAS 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.	–	–

For the test log please refer to Section 9 of this test report.

8.13 6.6.4.1.2 [WINNF.FT.D.RLQ.2] Domain Proxy Successful Relinquishment

8.13.1 Definitions and limits

6.6 CBSD Relinquishment Process
This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Relinquishment Procedure. A precondition is the CBSD has successfully discovered the SAS it wants to communicate with.
Each test generates a CBSD relinquishment request and validates the CBSD takes the appropriate action following the SAS relinquishment response. The CBSD shall send the Relinquishment request message after stopping the RF transmission.
Successful Relinquishment Request (responseCode 0)

8.13.2 Test date

Start date	May 28, 2019

8.13.3 Observations, settings and special notes

None

8.13.4 Test data

Table 8.13-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with SAS Test Harness • DP has successfully registered 2 CBSD with SAS Test Harness, each with cbsdId=Ci, i={1,2} • 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. Invoke trigger to relinquish each UUT Grant from the SAS Test Harness	–	–
2	Verify DP sends a Relinquishment Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Relinquishment Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • cbsdId = Ci • grantId = Gi	☒	☐
3	If a separate Relinquishment Request message was sent for each CBSD by the DP, the SAS 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 SAS Test Harness shall respond with a single Response message containing a 2-object array. Parameters for each CBSD within the Relinquishment Response shall be as follows: • cbsdId = Ci • grantId = Gi • responseCode = 0	–	–
4	After completion of step 3, SAS Test Harness will not provide any additional positive response (responseCode=0) to further request messages from the UUT.	–	–
5	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 at any time between triggering the relinquishments and UUT sending the relinquishment requests for each CBSD.	☒	☐

For the test log please refer to Section 9 of this test report.

8.14 6.6.4.1.4 [WINNF.FT.D.RLQ.4] Domain Proxy Unsuccessful Relinquishment, responseCode=102

8.14.1 Definitions and limits

6.6 CBSD Relinquishment Process
This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Relinquishment Procedure. A precondition is the CBSD has successfully discovered the SAS it wants to communicate with.
Each test generates a CBSD relinquishment request and validates the CBSD takes the appropriate action following the SAS relinquishment response. The CBSD shall send the Relinquishment request message after stopping the RF transmission.
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.
This test case applies to Domain Proxy supervising two CBSDs. The following are the test execution steps where the Relinquishment response contains responseCode (Ri) = 102 for each CBSD.

8.14.2 Test date

Start date	May 28, 2019

8.14.3 Observations, settings and special notes

None

8.14.4 Test data

Table 8.14-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with SAS Test Harness • DP has successfully registered 2 CBSD with SAS Test Harness, each with cbsdId=Ci, i={1,2} • 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. Invoke trigger on UUT to Relinquish Grant from the SAS Test Harness	–	–
2	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. Verify DP sends a Relinquishment Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Relinquishment Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • cbsdId = Ci • grantId = Gi	☒	☐
3	If a separate Relinquishment Request message was sent for each CBSD by the DP, the SAS 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 SAS 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 = Ci • No grantId • responseCode = Ri	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–

Step	Test Execution Steps	Pass	Fail
5	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 stopped RF transmission at any time between triggering the relinquishment and UUT sending the relinquishment request	☒	☐

For the test log please refer to Section 9 of this test report.

8.15 6.6.4.3.2 [WINNF.FT.D.RLQ.6] Domain Proxy Unsuccessful Relinquishment, responseCode=103

8.15.1 Definitions and limits

6.6 CBSD Relinquishment Process
This section provides test steps, condition and procedures to test the conformance of the CBSD implementation for the CBSD Relinquishment Procedure. A precondition is the CBSD has successfully discovered the SAS it wants to communicate with.
Each test generates a CBSD relinquishment request and validates the CBSD takes the appropriate action following the SAS relinquishment response. The CBSD shall send the Relinquishment request message after stopping the RF transmission.
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.
The same steps provided for WINNF.FT.D.RLQ.4 shall be executed for this test, with the exception that the Relinquishment response contains responseCode (Ri) = 103 and responseData = “grantId” for each CBSD.

8.15.2 Test date

Start date	May 28, 2019

8.15.3 Observations, settings and special notes

None

8.15.4 Test data

Table 8.15-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with SAS Test Harness • DP has successfully registered 2 CBSD with SAS Test Harness, each with cbsdId=Ci, i={1,2} • 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. Invoke trigger on UUT to Relinquish Grant from the SAS Test Harness	–	–
2	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. Verify DP sends a Relinquishment Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Relinquishment Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • cbsdId = Ci • grantId = Gi	☒	☐
3	If a separate Relinquishment Request message was sent for each CBSD by the DP, the SAS 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 SAS 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 = Ci • responseCode (Ri) = 103 for CBSD1 and CBSD2 • responseData = “grantId” for CBSD1 and CBSD2	–	–
4	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–

Step	Test Execution Steps	Pass	Fail
5	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 stopped RF transmission at any time between triggering the relinquishment and UUT sending the relinquishment request	☒	☐

For the test log please refer to Section 9 of this test report.

8.16 6.7.4.1.2 [WINNF.FT.D.DRG.2] Domain Proxy Successful Deregistration

8.16.1 Definitions and limits

6.7 CBSD Deregistration Process
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.
Each test generates a CBSD deregistration request and validates the CBSD takes the appropriate actions following the SAS deregistration response.
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.
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.
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.
Each test generates a CBSD deregistration request and validates the CBSD takes the appropriate actions following the SAS deregistration response.
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.
Successful Deregistration Request (responseCode 0)

8.16.2 Test date

Start date	May 28, 2019

8.16.3 Observations, settings and special notes

None

8.16.4 Test data

Table 8.16-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • Each UUT has successfully registered with SAS Test Harness • Each UUT is in the authorized state • DP has successfully completed SAS Discovery and Authentication with SAS Test Harness • DP has successfully registered 2 CBSD with SAS Test Harness, each with cbsdId=Ci, i={1,2} • 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. Invoke trigger to deregister each UUT from the SAS Test Harness	–	–
2	UUT sends a Relinquishment request and receives Relinquishment response with responseCode=0	–	–
3	Verify DP sends a Deregistration Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Deregistration Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • cbsdId = Ci	☒	☐

Step	Test Execution Steps	Pass	Fail
4	If a separate Deregistration Request message was sent for each CBSD by the DP, the SAS 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 SAS 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 = Ci • responseCode = 0	–	–
5	After completion of step 4, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
6	Monitor the RF output of each UUT from start of test until 60 seconds after Step 4 is complete. This is the end of the test. Verify: • UUT stopped RF transmission at any time between triggering the deregistration and either A OR B occurs: A. UUT sending a Registration Request message, as this is not mandatory B. UUT sending a Deregistration Request message	☒	☐

For the test log please refer to Section 9 of this test report.

8.17 6.7.4.2.2 [WINNF.FT.D.DRG.4] Domain Proxy Deregistration responseCode=102

8.17.1 Definitions and limits

6.7 CBSD Deregistration Process
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.
Each test generates a CBSD deregistration request and validates the CBSD takes the appropriate actions following the SAS deregistration response.
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.
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.
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.
Each test generates a CBSD deregistration request and validates the CBSD takes the appropriate actions following the SAS deregistration response.
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.
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 The following are the test execution steps where the Deregistration response contains responseCode (Ri) = 102 for each CBSD.

8.17.2 Test date

Start date	May 28, 2019

8.17.3 Observations, settings and special notes

None

8.17.4 Test data

Table 8.17-1: *Test results*

Step	Test Execution Steps	Pass	Fail
1	Ensure the following conditions are met for test entry: • DP has successfully completed SAS Discovery and Authentication with SAS Test Harness • DP has successfully registered 2 CBSD with SAS Test Harness, each with cbsdId=Ci, i={1,2} • 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. Invoke trigger to deregister each UUT from the SAS Test Harness	–	–
2	UUT sends a Relinquishment request and receives Relinquishment response with responseCode=0 for each CBSD	–	–
3	Verify DP sends a Deregistration Request message for each CBSD. This may occur in a separate message per CBSD, or together in a single message with array of 2. Verify Deregistration Request message contains all required parameters properly formatted for each CBSD, specifically, for CBSDi: • cbsdId = Ci	–	–

Step	Test Execution Steps	Pass	Fail
4	If a separate Deregistration Request message was sent for each CBSD by the DP, the SAS 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 SAS 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 either response • responseCode (Ri) = 102	–	–
5	After completion of step 3, SAS Test Harness will not provide any positive response (responseCode=0) to further request messages from the UUT.	–	–
6	Monitor the RF output of each UUT from start of test until 60 seconds after Step 4 is complete. This is the end of the test. Verify: • UUT stopped RF transmission at any time between triggering the deregistration and either A OR B occurs: A. UUT sending a Registration Request message, as this is not mandatory B. UUT sending a Deregistration Request message	☒	☐

For the test log please refer to Section 9 of this test report.

Section 9. Log files library

9.1 Log file for test case ID: WINNF.FT.D.REG.2


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>20:50:43.805 Sas.cpp          post          17030 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>20:50:43.805 Sas.cpp          post          17030 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [],
   "userId": "abc"
  }
 ]
}
<7>20:50:43.863 Sas.cpp          post          17030 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
```

9.2 Log file for test case ID: WINNF.FT.D.REG.4


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:14:22.550 Sas.cpp post 45267 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:14:22.551 Sas.cpp post 45267 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:14:22.601 Sas.cpp post 45267 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
```

9.3 Log file for test case ID: WINNF.FT.C.REG.9


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:21:26.319 Sas.cpp post 45678 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 102
   }
  }
 ]
}
<7>15:21:26.319 Cbsd.cpp cbsd_main_ 45678 [36;1mDBG[0m ERROR state reset to
UNREGISTERED
<7>15:21:26.319 Sas.cpp post 45678 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:21:26.377 Sas.cpp post 45678 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 102
   }
  }
 ]
}
```

9.4 Log file for test case ID: WINNF.FT.D.REG.11


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:25:36.678 Sas.cpp post 46036 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 200
   }
  }
 ]
}
<7>15:25:36.678 Sas.cpp post 46036 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
"longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:25:36.736 Sas.cpp post 46036 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 200
   }
  }
 ]
}
<6>15:25:36.737 CbrsDaemon.cpp onLoop 46036 [34;1mINF[0m Listening for 59 seconds
<7>15:25:36.737 SpvLaunchdProxy.cpp create 46036 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>15:25:36.737 SpvLaunchdProxy.cpp create 46036 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>15:25:36.737 SpvLaunchdProxy.cpp initSpvLaunchdProxy 46036 [36;1mDBG[0m SpvLaunchd is
running.
<7>15:25:36.738 SpvLaunchdProxy.cpp logDBusMessage 46036 [36;1mDBG[0m handleRequest:
signal sender=org.freedesktop.DBus -> dest=:1.141 serial=2 path=/org/freedesktop/DBus;
interface=org.freedesktop.DBus; member=NameAcquired; signature=s
<7>15:25:36.738 SpvLaunchdProxy.cpp dbusHandler 46036 [36;1mDBG[0m NameAcquired:
:1.141
<7>15:25:36.738 SpvLaunchdProxy.cpp dbusHandler 46036 [36;1mDBG[0m Connection name:
:1.141
<6>15:25:37.780 CbrsDaemon.cpp parseTree 46036 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>15:25:37.782 CbrsDaemon.cpp parseTree 46036 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<7>15:25:37.786 Sas.cpp post 46036 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
 "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
],
   "userId": "abc"
  }
```

```
]
}
<7>15:25:37.787 Sas.cpp          post          46036 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 200
   }
  }
 ]
}
<7>15:25:37.787 Sas.cpp          post          46036 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:25:37.827 Sas.cpp          post          46036 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 200
   }
  }
 ]
}
```

9.5 Log file for test case ID: WINNF.FT.D.REG.13


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:27:36.451 Sas.cpp          post            46202 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:27:36.451 Sas.cpp          post            46202 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
"latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:27:36.510 Sas.cpp          post            46202 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 103
   }
  }
 ]
}
```

9.6 Log file for test case ID: WINNF.FT.D.REG.15


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:30:06.660 Sas.cpp          post            46381 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:30:06.660 Sas.cpp          post            46381 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:30:06.718 Sas.cpp          post            46381 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 101
   }
  }
 ]
}
```

9.7 Log file for test case ID: WINNF.FT.D.REG.17


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:31:51.027 Sas.cpp          post            46540 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 100
   }
  }
 ]
}
<7>15:31:51.027 Cbsd.cpp         cbsd_main_      46540 [36;1mDBG[0m ERROR state reset to
UNREGISTERED
<7>15:31:51.027 Sas.cpp          post            46540 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
"latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:31:51.085 Sas.cpp          post            46540 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 100
   }
  }
 ]
}
```

9.8 Log file for test case ID: WINNF.FT.D.REG.19


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:33:40.848 Sas.cpp          post            46703 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:33:40.848 Sas.cpp          post            46703 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
"latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:33:40.907 Sas.cpp          post            46703 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "response": {
    "responseCode": 201
   }
  }
 ]
}
```

9.9 Log file for test case ID: WINNF.FT.D.HBT.2


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:42:08.887 Sas.cpp post 47131
[36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:42:08.888 Sas.cpp post 47131
[36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:42:08.952 Sas.cpp post 47131 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>15:42:08.953 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<7>15:42:08.953 SpvLaunchdProxy.cpp create 47131 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>15:42:08.954 SpvLaunchdProxy.cpp create 47131 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>15:42:08.954 SpvLaunchdProxy.cpp initSpvLaunchdProxy 47131 [36;1mDBG[0m SpvLaunchd is running.
<7>15:42:08.954 SpvLaunchdProxy.cpp logDBusMessage 47131 [36;1mDBG[0m handleRequest: signal sender=org.freedesktop.DBus ->
dest=:1.152 serial=2 path=/org/freedesktop/DBus; interface=org.freedesktop.DBus; member=NameAcquired; signature=s
<7>15:42:08.954 SpvLaunchdProxy.cpp dbusHandler 47131 [36;1mDBG[0m NameAcquired: :1.152
<7>15:42:08.954 SpvLaunchdProxy.cpp dbusHandler 47131 [36;1mDBG[0m Connection name: :1.152
<6>15:42:09.996 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:09.998 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>15:42:10.002 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<7>15:42:14.466 SpvLaunchdProxy.cpp logDBusMessage 47131 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null)
serial=258 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess; signature=s
<7>15:42:14.466 SpvLaunchdProxy.cpp logActiveEnbs 47131 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONNECTED"}]}
<6>15:42:15.508 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:15.511 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:15.516 CbrsDaemon.cpp persistEntities 47131 [36;1mDBG[0m Grant for cell 0, belonging to eNB 1 created.
<6>15:42:15.517 ManagerCbsd.cpp command 47131 [34;1mINF[0m Send command to CBSD on fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559050935,"user":"user"}, with timeout of 25
<6>15:42:15.517 ManagerCbsd.cpp getResponseFromReque 47131 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 : 5556] Send (timeout 25
seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559050935,"user":"user"}
<6>15:42:15.550 ManagerCbsd.cpp getResponseFromReque 47131 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 : 5556] Socket response
received (199152 bytes)
<7>15:42:15.553 Sas.cpp post 47131 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
```

```
},
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -95
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -99
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -98
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -94
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -97
      }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3635000000,
     "lowFrequency": 3625000000
    }
   }
  }
 ]
}
<7>15:42:15.561 Sas.cpp          post            47131 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T13:42:15Z",
   "grantId": "860630773",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:42:15.561 Sas.cpp          post            47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "operationState": "GRANTED"
  }
 ]
}
<7>15:42:15.603 Sas.cpp          post            47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:35Z"
  }
 ]
}
<6>15:42:15.604 ManagerEnb.cpp     command          47131 [34;1mINF[0m Sending tx_expire to eNB(1), with
expiration: 60000
<6>15:42:15.604 CbrsDaemon.cpp     onLoop           47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:15.951 Enb.cpp            onData           47163 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<7>15:42:18.096 SpvLaunchdProxy.cpp  logDBusMessage     47131 [36;1mDBG[0m handleRequest: signal
sender=:1.0 -> dest=(null) serial=260 path=/com/jmawireless/jsoft/SpvLaunchd;
interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess; signature=s
<7>15:42:18.096 SpvLaunchdProxy.cpp  logActiveEnbs      47131 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"",
"state":"CONNECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":
"CONNECTED"}]}
<6>15:42:19.144 CbrsDaemon.cpp     parseTree        47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>15:42:19.149 CbrsDaemon.cpp     parseTree        47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<7>15:42:19.155 CbrsDaemon.cpp     persistEntities  47131 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2
created.
<7>15:42:19.156 Sas.cpp          post            47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:19.160 Sas.cpp          post            47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:39Z"
  }
 ]
}
<6>15:42:19.160 ManagerCbsd.cpp    command          47131 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559050939,"user":"user"}, with
timeout of 25
<6>15:42:19.160 ManagerCbsd.cpp    getResponseFromReque 47131 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef :
5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559050939,"user":"user"}
<6>15:42:19.188 ManagerCbsd.cpp    getResponseFromReque 47131 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef :
5556] Socket response received (162122 bytes)
<7>15:42:19.191 Sas.cpp          post            47131 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -100
     },
```

```
      {
        "measBandwidth": 10000000,
        "measFrequency": 3560000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3570000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3580000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -100
      }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3645000000,
     "lowFrequency": 3635000000
    }
   }
  }
 ]
}
```

```
<7>15:42:19.237 Sas.cpp post 47131 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T13:42:19Z",
   "grantId": "352986227",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:42:19.237 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "operationState": "GRANTED"
  }
 ]
}
<7>15:42:19.280 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:39Z"
  }
 ]
}
<6>15:42:19.280 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:19.280 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:19.281 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:19.381 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:19.420 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:42:25.326 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:25.330 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:25.334 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:25.338 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:45Z"
  }
 ]
}
<7>15:42:25.338 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:25.381 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:45Z"
```

```
  }
 ]
}
<6>15:42:25.381 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:25.381 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:25.382 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:25.482 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:25.482 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:42:31.427 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:31.433 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:31.439 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:31.443 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:51Z"
  }
 ]
}
<7>15:42:31.443 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:31.486 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:51Z"
  }
 ]
}
<6>15:42:31.486 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:31.486 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:31.487 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:31.587 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:31.587 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:42:37.534 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:37.538 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:37.542 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:37.546 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:57Z"
  }
 ]
}
<7>15:42:37.546 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:37.589 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:45:57Z"
  }
 ]
}
<6>15:42:37.589 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:37.589 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:37.590 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:37.690 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:37.690 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:42:43.636 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:43.641 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:43.648 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:42:43.652 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "860630773",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:46:03Z"
  }
 ]
}
<7>15:42:43.652 Sas.cpp post 47131 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "352986227",
   "operationState": "AUTHORIZED"
  }
 ]
}
```

<7>15:42:43.695 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:03Z"
}
]
}
<6>15:42:43.695 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:43.695 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:43.696 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:43.796 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:43.796 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:42:49.742 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:49.747 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:49.752 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:42:49.754 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:09Z"
}
]
}
<7>15:42:49.755 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:42:49.797 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:09Z"
}
]
}
<6>15:42:49.797 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:49.797 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:49.798 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:49.898 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:49.898 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:42:55.843 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:42:55.849 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:42:55.855 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:42:55.856 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:15Z"
}
]
}
<7>15:42:55.857 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:42:55.897 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:15Z"
}
]
}
<6>15:42:55.897 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:42:55.897 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:42:55.898 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:42:55.998 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:42:55.998 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:43:01.949 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:43:01.953 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:43:01.957 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:01.959 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:21Z"
}
]
}

<7>15:43:01.959 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:01.999 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:21Z"
}
]
}
<6>15:43:01.999 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:43:01.999 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:43:02.000 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:43:02.100 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:43:02.100 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:43:08.046 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:43:08.051 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:43:08.058 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:08.059 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:28Z"
}
]
}
<7>15:43:08.059 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:08.099 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:28Z"
}
]
}
<6>15:43:08.099 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:43:08.099 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:43:08.100 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:43:08.200 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:43:08.200 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:43:14.153 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:43:14.157 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:43:14.161 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:14.162 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:34Z"
}
]
}
<7>15:43:14.163 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:14.203 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:34Z"
}
]
}
<6>15:43:14.203 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:43:14.203 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:43:14.204 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:43:14.304 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:43:14.304 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:43:20.255 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:43:20.259 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:43:20.265 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}

<7>15:43:20.266 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:40Z"
}
]
}
<7>15:43:20.266 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:20.307 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:40Z"
}
]
}
<6>15:43:20.307 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:43:20.307 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:43:20.308 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:43:20.408 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:43:20.408 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:43:26.354 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:43:26.359 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:43:26.364 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:26.365 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:46Z"
}
]
}
<7>15:43:26.366 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:26.406 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:46Z"
}
]
}
<6>15:43:26.406 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:43:26.406 ManagerEnb.cpp command 47131 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:43:26.407 CbrsDaemon.cpp onLoop 47131 [34;1mINF[0m Listening for 5 seconds
<6>15:43:26.507 Enb.cpp onData 47163 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:43:26.507 Enb.cpp onData 47193 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:43:32.458 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:43:32.463 CbrsDaemon.cpp parseTree 47131 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:43:32.468 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:32.469 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "860630773",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:52Z"
}
]
}
<7>15:43:32.469 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"operationState": "AUTHORIZED"
}
]
}
<7>15:43:32.510 Sas.cpp post 47131 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "352986227",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:46:52Z"
}
]
}

9.10 Log file for test case ID: WINNF.FT.D.HBT.8


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:47:33.815 Sas.cpp post 47461 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:47:33.815 Sas.cpp post 47461 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>15:47:33.879 Sas.cpp post 47461 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>15:47:33.880 CbrsDaemon.cpp onLoop 47461 [34;1mINF[0m Listening for 5 seconds
<7>15:47:33.880 SpvLaunchdProxy.cpp create 47461 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>15:47:33.880 SpvLaunchdProxy.cpp create 47461 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>15:47:33.881 SpvLaunchdProxy.cpp initSpvLaunchdProxy 47461 [36;1mDBG[0m SpvLaunchd is running.
<7>15:47:33.881 SpvLaunchdProxy.cpp logDBusMessage 47461 [36;1mDBG[0m handleRequest: signal
sender=org.freedesktop.DBus -> dest=:1.154 serial=2 path=/org/freedesktop/DBus; interface=org.freedesktop.DBus;
member=NameAcquired; signature=s
<7>15:47:33.881 SpvLaunchdProxy.cpp dbusHandler 47461 [36;1mDBG[0m NameAcquired: :1.154
<7>15:47:33.881 SpvLaunchdProxy.cpp dbusHandler 47461 [36;1mDBG[0m Connection name: :1.154
<6>15:47:34.922 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>15:47:34.925 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<6>15:47:34.929 CbrsDaemon.cpp onLoop 47461 [34;1mINF[0m Listening for 5 seconds
<7>15:47:37.633 SpvLaunchdProxy.cpp logDBusMessage 47461 [36;1mDBG[0m handleRequest: signal sender=:1.0
-> dest=(null) serial=268 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>15:47:37.641 SpvLaunchdProxy.cpp logActiveEnbs 47461 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","sta
te":"CONNECTED"}]}
<6>15:47:38.683 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>15:47:38.686 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<7>15:47:38.691 CbrsDaemon.cpp persistEntities 47461 [36;1mDBG[0m Grant for cell 0, belonging to eNB 1
created.
<6>15:47:38.692 ManagerCbsd.cpp command 47461 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559051258,"user":"user"}, with
timeout of 25
<6>15:47:38.692 ManagerCbsd.cpp getResponseFromReque 47461 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559051258,"user":"user"}
<6>15:47:38.724 ManagerCbsd.cpp getResponseFromReque 47461 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Socket response received (198670 bytes)
<7>15:47:38.728 Sas.cpp post 47461 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -97
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
```

```
      {
        "measBandwidth": 10000000,
        "measFrequency": 3570000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3580000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -92
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -99
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -89
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -94
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -97
      }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3635000000,
     "lowFrequency": 3625000000
    }
   }
  }
 ]
}
```

```
<7>15:47:38.735 Sas.cpp post 47461 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T13:47:38Z",
   "grantId": "768439759",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>15:47:38.735 Sas.cpp post 47461 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "768439759",
   "operationState": "GRANTED"
  }
 ]
}
<7>15:47:38.777 Sas.cpp post 47461 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "768439759",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:50:58Z"
  }
 ]
}
<6>15:47:38.777 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration:
60000
<6>15:47:38.778 CbrsDaemon.cpp onLoop 47461 [34;1mINF[0m Listening for 5 seconds
<6>15:47:38.924 Enb.cpp onData 47493 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<7>15:47:40.166 SpvLaunchdProxy.cpp logDBusMessage 47461 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=270 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>15:47:40.166 SpvLaunchdProxy.cpp logActiveEnbs 47461 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":
"CONNECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"
}]}
<6>15:47:41.257 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:47:41.261 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:47:41.266 CbrsDaemon.cpp persistEntities 47461 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>15:47:41.267 Sas.cpp post 47461 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "768439759",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>15:47:41.271 Sas.cpp post 47461 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "768439759",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:51:01Z"
  }
 ]
}
```

<6>15:47:41.271 ManagerCbsd.cpp command 47461 [34;1mINF[0m Send command to CBSD on fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559051261,"user":"user"}, with timeout of 25
<6>15:47:41.271 ManagerCbsd.cpp getResponseFromReque 47461 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559051261,"user":"user"}
<6>15:47:41.304 ManagerCbsd.cpp getResponseFromReque 47461 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Socket response received (198421 bytes)
<7>15:47:41.307 Sas.cpp post 47461 [36;1mDBG[0m {

```
 "grantRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -97
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -93
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -100
     }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3645000000,
     "lowFrequency": 3635000000
    }
   }
  }
 ]
}
```

<7>15:47:41.352 Sas.cpp post 47461 [36;1mDBG[0m {

```
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T13:47:41Z",
   "grantId": "656363028",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
```

<7>15:47:41.352 Sas.cpp post 47461 [36;1mDBG[0m {

```
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "656363028",
   "operationState": "GRANTED"
  }
 ]
}
```

<7>15:47:41.395 Sas.cpp post 47461 [36;1mDBG[0m {

```
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "656363028",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T13:51:01Z"
  }
 ]
}
```

<6>15:47:41.395 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:47:41.395 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:47:41.396 CbrsDaemon.cpp onLoop 47461 [34;1mINF[0m Listening for 5 seconds
<6>15:47:41.520 Enb.cpp onData 47493 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:47:41.535 Enb.cpp onData 47525 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:47:47.447 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:47:47.451 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:47:47.456 Sas.cpp post 47461 [36;1mDBG[0m {

```
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "768439759",
   "operationState": "AUTHORIZED"
  }
 ]
}
```

<7>15:47:47.460 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "768439759",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:51:07Z"
}
]
}
<7>15:47:47.461 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "656363028",
"operationState": "AUTHORIZED"
}
]
}
<7>15:47:47.504 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "656363028",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:51:07Z"
}
]
}
<6>15:47:47.504 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:47:47.504 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:47:47.505 CbrsDaemon.cpp onLoop 47461 [34;1mINF[0m Listening for 5 seconds
<6>15:47:47.605 Enb.cpp onData 47493 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:47:47.605 Enb.cpp onData 47525 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:47:53.557 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:47:53.562 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:47:53.567 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "768439759",
"operationState": "AUTHORIZED"
}
]
}
<7>15:47:53.572 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "768439759",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:51:13Z"
}
]
}
<7>15:47:53.572 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "656363028",
"operationState": "AUTHORIZED"
}
]
}
<7>15:47:53.616 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "656363028",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:51:13Z"
}
]
}
<6>15:47:53.616 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>15:47:53.616 ManagerEnb.cpp command 47461 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>15:47:53.617 CbrsDaemon.cpp onLoop 47461 [34;1mINF[0m Listening for 5 seconds
<6>15:47:53.717 Enb.cpp onData 47493 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>15:47:53.717 Enb.cpp onData 47525 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>15:47:59.669 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>15:47:59.673 CbrsDaemon.cpp parseTree 47461 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>15:47:59.678 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "768439759",
"operationState": "AUTHORIZED"
}
]
}
<7>15:47:59.680 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "768439759",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T13:51:19Z"
}
]
}
<7>15:47:59.680 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "656363028",
"operationState": "AUTHORIZED"
}
]
}
<7>15:47:59.724 Sas.cpp post 47461 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "656363028",
"response": {
"responseCode": 500
},
"transmitExpireTime": "2019-05-28T13:47:59Z"
}
]
}

9.11 Log file for test case ID: WINNF.FT.D.MES.2


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:02:52.365 Sas.cpp          post            48321 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReportConfig": [
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:02:52.366 Sas.cpp          post            48321 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:02:52.430 Sas.cpp          post            48321 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "measReportConfig": [
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:02:52.431 CbrsDaemon.cpp   onLoop          48321 [34;1mINF[0m Listening for 5 seconds
<7>16:02:52.431 SpvLaunchdProxy.cpp  create          48321 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:02:52.431 SpvLaunchdProxy.cpp  create          48321 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:02:52.431 SpvLaunchdProxy.cpp  initSpvLaunchdProxy  48321 [36;1mDBG[0m SpvLaunchd is running.
<7>16:02:52.432 SpvLaunchdProxy.cpp  logDBusMessage    48321 [36;1mDBG[0m handleRequest: signal
sender=org.freedesktop.DBus -> dest=:1.159 serial=2 path=/org/freedesktop/DBus; interface=org.freedesktop.DBus;
member=NameAcquired; signature=s
<7>16:02:52.432 SpvLaunchdProxy.cpp  dbusHandler       48321 [36;1mDBG[0m NameAcquired: :1.159
<7>16:02:52.432 SpvLaunchdProxy.cpp  dbusHandler       48321 [36;1mDBG[0m Connection name: :1.159
<6>16:02:53.475 CbrsDaemon.cpp   parseTree       48321 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:02:53.478 CbrsDaemon.cpp   parseTree       48321 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<6>16:02:53.481 CbrsDaemon.cpp   onLoop          48321 [34;1mINF[0m Listening for 5 seconds
<7>16:02:56.906 SpvLaunchdProxy.cpp  logDBusMessage    48321 [36;1mDBG[0m handleRequest: signal
sender=:1.0 -> dest=(null) serial=290 path=/com/jmawireless/jsoft/SpvLaunchd;
interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess; signature=s
<7>16:02:56.906 SpvLaunchdProxy.cpp  logActiveEnbs     48321 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","s
tate":"CONNECTED"}]}
<6>16:02:57.947 CbrsDaemon.cpp   parseTree       48321 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:02:57.950 CbrsDaemon.cpp   parseTree       48321 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<7>16:02:57.954 CbrsDaemon.cpp   persistEntities    48321 [36;1mDBG[0m Grant for cell 0, belonging to eNB 1
created.
<6>16:02:57.955 ManagerCbsd.cpp  command         48321 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052177,"user":"user"}, with
timeout of 25
<6>16:02:57.955 ManagerCbsd.cpp  getResponseFromReque 48321 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052177,"user":"user"}
<6>16:02:57.988 ManagerCbsd.cpp  getResponseFromReque 48321 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Socket response received (198734 bytes)
<7>16:02:57.991 Sas.cpp          post            48321 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
```

```
      {
        "measBandwidth": 10000000,
        "measFrequency": 3560000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3570000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3580000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -81
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -98
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -87
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -95
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -97
      }
     ]
    },
    "operationParam": {
     "maxEirp": 0,
     "operationFrequencyRange": {
      "highFrequency": 3635000000,
      "lowFrequency": 3625000000
     }
    }
   }
  ]
}
```

```
<7>16:02:57.998 Sas.cpp          post             48321 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:02:57Z",
   "grantId": "22853067",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:02:57.998 Sas.cpp          post             48321 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "22853067",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:02:58.039 Sas.cpp          post             48321 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "22853067",
   "response": {
    "responseCode": 501
   },
   "transmitExpireTime": "2019-05-28T14:02:57Z"
  }
 ]
}
<3>16:02:58.039 Grant.cpp        grant_main_      48321 [31;1mERR[0m Heartbeat procedure failed for CBSD XM2-
X19AX35M2Mock-SAS1012482003, grantId 22853067
<7>16:02:58.039 Grant.cpp        grant_main_      48321 [36;1mDBG[0m ERROR state reset to IDLE
<6>16:02:58.040 CbrsDaemon.cpp   onLoop           48321 [34;1mINF[0m Listening for 5 seconds
<7>16:03:00.028 SpvLaunchdProxy.cpp logDBusMessage    48321 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=292 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:03:00.028 SpvLaunchdProxy.cpp logActiveEnbs     48321 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONN
ECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:03:01.069 CbrsDaemon.cpp   parseTree        48321 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:03:01.075 CbrsDaemon.cpp   parseTree        48321 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:03:01.082 CbrsDaemon.cpp   persistEntities  48321 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:03:01.083 Sas.cpp          post             48321 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -81
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
```

```
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -98
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -87
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -95
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -97
      }
     ]
    },
    "operationParam": {
     "maxEirp": 0,
     "operationFrequencyRange": {
      "highFrequency": 3635000000,
      "lowFrequency": 3625000000
     }
    }
   }
  ]
}<7>16:03:01.086 Sas.cpp          post              48321 [36;1mDBG[0m {
  "grantResponse": [
   {
    "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
    "response": {
     "responseCode": 400
    }
   }
  ]
}
<3>16:03:01.086 Grant.cpp         grant_main_       48321 [31;1mERR[0m Grant procedure failed for
CBSD XM2-X19AX35M2Mock-SAS1012482003, cell 0 eNB 1
<7>16:03:01.086 Grant.cpp         grant_main_       48321 [36;1mDBG[0m ERROR state reset to IDLE
<6>16:03:01.086 ManagerCbsd.cpp   command           48321 [34;1mINF[0m Send command to CBSD
on fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052181,"user":
"user"}, with timeout of 25
<6>16:03:01.086 ManagerCbsd.cpp   getResponseFromReque 48321 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2ef : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052181,"user":
"user"}
<6>16:03:01.119 ManagerCbsd.cpp   getResponseFromReque 48321 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2ef : 5556] Socket response received (198590 bytes)
<7>16:03:01.122 Sas.cpp           post              48321 [36;1mDBG[0m {
"grantRequest": [
 {
  "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
  "measReport": {
   "rcvdPowerMeasReports": [
    {
     "measBandwidth": 10000000,
     "measFrequency": 3550000000,
     "measRcvdPower": -95
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3560000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3570000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3580000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3590000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3600000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3610000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3620000000,
     "measRcvdPower": -98
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3630000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3640000000,
     "measRcvdPower": -95
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3650000000,
     "measRcvdPower": -96
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3660000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3670000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3680000000,
     "measRcvdPower": -94
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3690000000,
     "measRcvdPower": -98
    }
   ]
  },}
```

```
    "operationParam": {
     "maxEirp": 0,
     "operationFrequencyRange": {
      "highFrequency": 3645000000,
      "lowFrequency": 3635000000
     }
    }
   }
 ]
}
<7>16:03:01.167 Sas.cpp          post            48321 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:03:01Z",
   "grantId": "897583289",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:03:01.167 Sas.cpp          post            48321 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "897583289",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:03:01.207 Sas.cpp          post            48321 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "897583289",
   "response": {
    "responseCode": 501
   },
   "transmitExpireTime": "2019-05-28T14:03:01Z"
  }
 ]
}
<3>16:03:01.207 Grant.cpp        grant_main_      48321 [31;1mERR[0m Heartbeat procedure failed
for CBSD XM2-XAF2335M2Mock-SAS1012482006, grantId 897583289
<7>16:03:01.207 Grant.cpp        grant_main_      48321 [36;1mDBG[0m ERROR state reset to IDLE
<6>16:03:01.208 CbrsDaemon.cpp   onLoop           48321 [34;1mINF[0m Listening for 5 seconds
<6>16:03:07.250 CbrsDaemon.cpp   parseTree        48321 [34;1mINF[0m Found CBRS Cell: cell_id
0, earfcn_dl 56040
<6>16:03:07.254 CbrsDaemon.cpp   parseTree        48321 [34;1mINF[0m Found CBRS Cell: cell_id
1, earfcn_dl 56140
<7>16:03:07.259 Sas.cpp          post            48321 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -81
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -87
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -97
     }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3635000000,
     "lowFrequency": 3625000000
    }
   }
  }
 ]
}
<7>16:03:07.261 Sas.cpp          post            48321 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 400
   }
  }
 ]
}
```

<3>16:03:07.261 Grant.cpp grant_main_ 48321 [31;1mERR[0m Grant procedure failed for CBSD XM2-X19AX35M2Mock-SAS1012482003, cell 0 eNB 1
<7>16:03:07.261 Grant.cpp grant_main_ 48321 [36;1mDBG[0m ERROR state reset to IDLE
<7>16:03:07.261 Sas.cpp post 48321 [36;1mDBG[0m {

```
 "grantRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -94
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -98
     }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3645000000,
     "lowFrequency": 3635000000
    }
   }
  }
 ]
}
```

<7>16:03:07.301 Sas.cpp post 48321 [36;1mDBG[0m {

```
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 400
   }
  }
 ]
}
```

<3>16:03:07.301 Grant.cpp grant_main_ 48321 [31;1mERR[0m Grant procedure failed for CBSD XM2-XAF2335M2Mock-SAS1012482006, cell 1 eNB 2
<7>16:03:07.301 Grant.cpp grant_main_ 48321 [36;1mDBG[0m ERROR state reset to IDLE
<6>16:03:07.302 CbrsDaemon.cpp onLoop 48321 [34;1mINF[0m Listening for 5 seconds
<6>16:03:13.348 CbrsDaemon.cpp parseTree 48321 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:03:13.354 CbrsDaemon.cpp parseTree 48321 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:03:13.360 Sas.cpp post 48321 [36;1mDBG[0m {

```
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -81
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -100
```

```
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -98
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -87
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -95
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -97
      }
     ]
    },
    "operationParam": {
     "maxEirp": 0,
     "operationFrequencyRange": {
      "highFrequency": 3635000000,
      "lowFrequency": 3625000000
     }
    }
   }
  ]
}
<7>16:03:13.361 Sas.cpp          post             48321 [36;1mDBG[0m {
  "grantResponse": [
   {
    "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
    "response": {
     "responseCode": 400
    }
   }
  ]
}
<3>16:03:13.361 Grant.cpp        grant_main_      48321 [31;1mERR[0m Grant procedure failed for
CBSD XM2-X19AX35M2Mock-SAS1012482003, cell 0 eNB 1
<7>16:03:13.361 Grant.cpp        grant_main_      48321 [36;1mDBG[0m ERROR state reset to IDLE
<7>16:03:13.362 Sas.cpp          post             48321 [36;1mDBG[0m {
  "grantRequest": [
   {
    "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
    "measReport": {
     "rcvdPowerMeasReports": [
      {
       "measBandwidth": 10000000,
       "measFrequency": 3550000000,
       "measRcvdPower": -95
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3560000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3570000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3580000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3590000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3600000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3610000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3620000000,
       "measRcvdPower": -98
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3630000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3640000000,
       "measRcvdPower": -95
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3650000000,
       "measRcvdPower": -96
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3660000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3670000000,
       "measRcvdPower": -100
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3680000000,
       "measRcvdPower": -94
      },
      {
       "measBandwidth": 10000000,
       "measFrequency": 3690000000,
       "measRcvdPower": -98
      }
     ]
    },
    "operationParam": {
     "maxEirp": 0,
     "operationFrequencyRange": {
      "highFrequency": 3645000000,
      "lowFrequency": 3635000000
     }
    }
   }
  ]
}
```

<7>16:03:13.402 Sas.cpp post 48321 [36;1mDBG[0m {
"grantResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"response": {
"responseCode": 400
}
}
]
}

9.12 Log file for test case ID: WINNF.FT.D.MES.5

{
"registrationRequest": [
{
"airInterface": {
"radioTechnology": "E_UTRA"
},
"callSign": "?",
"cbsdCategory": "A",
"cbsdInfo": {
"firmwareVersion": "v2.0.5",
"hardwareVersion": "v1.0.45",
"model": "CPRI_DEVICE-XXX",
"softwareVersion": "v1.2.1",
"vendor": "JMA Wireless"
},
"cbsdSerialNumber": "1012482003",
"fccId": "XM2-X19AX35M2",
"installationParam": {
"antennaAzimuth": 70,
"antennaBeamwidth": 45,
"antennaDowntilt": 36,
"antennaGain": 0,
"antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
"eirpCapability": 15,
"height": 15.0,
"heightType": "AMSL",
"horizontalAccuracy": 49,
"indoorDeployment": true,
"latitude": 43.09,
"longitude": -76.15,
"verticalAccuracy": 2
},
"measCapability": [
"RECEIVED_POWER_WITH_GRANT",
"RECEIVED_POWER_WITHOUT_GRANT"
],
"userId": "abc"
}
]
}
<7>16:07:43.895 Sas.cpp post 48631 [36;1mDBG[0m {
"registrationResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"response": {
"responseCode": 0
}
}
]
}
<7>16:07:43.895 Sas.cpp post 48631 [36;1mDBG[0m {
"registrationRequest": [
{
"airInterface": {
"radioTechnology": "E_UTRA"
},
"callSign": "?",
"cbsdCategory": "A",
"cbsdInfo": {
"firmwareVersion": "v2.0.5",
"hardwareVersion": "v1.0.45",
"model": "CPRI_DEVICE-XXX",
"softwareVersion": "v1.2.1",
"vendor": "JMA Wireless"
},
"cbsdSerialNumber": "1012482006",
"fccId": "XM2-XAF2335M2",
"installationParam": {
"antennaAzimuth": 70,
"antennaBeamwidth": 45,
"antennaDowntilt": 36,
"antennaGain": 0,
"antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
"eirpCapability": 15,
"height": 15.0,
"heightType": "AMSL",
"horizontalAccuracy": 49,
"indoorDeployment": true,
"latitude": 43.09,
"longitude": -76.15,
"verticalAccuracy": 2
},
"measCapability": [
"RECEIVED_POWER_WITH_GRANT",
"RECEIVED_POWER_WITHOUT_GRANT"
],
"userId": "abc"
}
]
}
<7>16:07:43.959 Sas.cpp post 48631 [36;1mDBG[0m {
"registrationResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"response": {
"responseCode": 0
}
}
]
}
<6>16:07:43.960 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<7>16:07:43.960 SpvLaunchdProxy.cpp create 48631 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:07:43.960 SpvLaunchdProxy.cpp create 48631 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:07:43.960 SpvLaunchdProxy.cpp initSpvLaunchdProxy 48631 [36;1mDBG[0m SpvLaunchd is running.
<7>16:07:43.961 SpvLaunchdProxy.cpp logDBusMessage 48631 [36;1mDBG[0m handleRequest: signal
sender=org.freedesktop.DBus -> dest=:1.162 serial=2 path=/org/freedesktop/DBus;
interface=org.freedesktop.DBus; member=NameAcquired; signature=s
<7>16:07:43.961 SpvLaunchdProxy.cpp dbusHandler 48631 [36;1mDBG[0m NameAcquired: :1.162
<7>16:07:43.961 SpvLaunchdProxy.cpp dbusHandler 48631 [36;1mDBG[0m Connection name: :1.162
<6>16:07:45.003 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:07:45.006 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<6>16:07:45.010 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<7>16:07:48.696 SpvLaunchdProxy.cpp logDBusMessage 48631 [36;1mDBG[0m handleRequest: signal
sender=:1.0 -> dest=(null) serial=301 path=/com/jmawireless/jsoft/SpvLaunchd;
interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess; signature=s
<7>16:07:48.697 SpvLaunchdProxy.cpp logActiveEnbs 48631 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"",
"state":"CONNECTED"}]}
<6>16:07:49.744 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:07:49.748 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<7>16:07:49.753 CbrsDaemon.cpp persistEntities 48631 [36;1mDBG[0m Grant for cell 0, belonging to eNB 1
created.

<6>16:07:49.754 ManagerCbsd.cpp command 48631 [34;1mINF[0m Send command to CBSD on fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052469,"user":"user"}, with timeout of 25
<6>16:07:49.754 ManagerCbsd.cpp getResponseFromReque 48631 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052469,"user":"user"}
<6>16:07:49.786 ManagerCbsd.cpp getResponseFromReque 48631 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 : 5556] Socket response received (198741 bytes)
<7>16:07:49.789 Sas.cpp post 48631 [36;1mDBG[0m {
"grantRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"measReport": {
"rcvdPowerMeasReports": [
{
"measBandwidth": 10000000,
"measFrequency": 3550000000,
"measRcvdPower": -96
},
{
"measBandwidth": 10000000,
"measFrequency": 3560000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3570000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3580000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3590000000,
"measRcvdPower": -95
},
{
"measBandwidth": 10000000,
"measFrequency": 3600000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3610000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3620000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3630000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3640000000,
"measRcvdPower": -99
},
{
"measBandwidth": 10000000,
"measFrequency": 3650000000,
"measRcvdPower": -98
},
{
"measBandwidth": 10000000,
"measFrequency": 3660000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3670000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3680000000,
"measRcvdPower": -95
},
{
"measBandwidth": 10000000,
"measFrequency": 3690000000,
"measRcvdPower": -97
}
]
},
"operationParam": {
"maxEirp": 0,
"operationFrequencyRange": {
"highFrequency": 3635000000,
"lowFrequency": 3625000000
}
}
}
]
}
<7>16:07:49.796 Sas.cpp post 48631 [36;1mDBG[0m {
"grantResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"channelType": "GAA",
"grantExpireTime": "2019-06-04T14:07:49Z",
"grantId": "992128042",
"heartbeatInterval": 60,
"response": {
"responseCode": 0
}
}
]
}
<7>16:07:49.796 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"operationState": "GRANTED"
}
]
}
<7>16:07:49.838 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:09Z"
}
]
}
<6>16:07:49.838 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:07:49.839 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:07:49.985 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<7>16:07:50.686 SpvLaunchdProxy.cpp logDBusMessage 48631 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=303 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess; signature=s
<7>16:07:50.686 SpvLaunchdProxy.cpp logActiveEnbs 48631 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONNECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:07:51.733 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:07:51.738 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:07:51.744 CbrsDaemon.cpp persistEntities 48631 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:07:51.745 Sas.cpp post 48631 [36;1mDBG[0m {

```
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:07:51.749 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:11Z"
  }
 ]
}
<6>16:07:51.749 ManagerCbsd.cpp    command         48631 [34;1mINF[0m Send command to CBSD
on fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052471,"user":
"user"}, with timeout of 25
<6>16:07:51.749 ManagerCbsd.cpp    getResponseFromReque 48631 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2ef : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052471,"user":
"user"}
<6>16:07:51.781 ManagerCbsd.cpp    getResponseFromReque 48631 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2ef : 5556] Socket response received (198725 bytes)
<7>16:07:51.784 Sas.cpp          post            48631 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -97
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -97
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -97
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -94
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -98
     }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3645000000,
     "lowFrequency": 3635000000
    }
   }
  }
 ]
}
<7>16:07:51.830 Sas.cpp          post            48631 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:07:51Z",
   "grantId": "681638789",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:07:51.830 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:07:51.873 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:11Z"
  }
 ]
}
```

```
<6>16:07:51.873 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:07:51.873 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:07:51.874 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:07:51.974 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:07:52.013 Enb.cpp onData 48693 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:07:57.919 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:07:57.924 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:07:57.930 Sas.cpp post 48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:07:57.934 Sas.cpp post 48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "measReportConfig": [
    "RECEIVED_POWER_WITH_GRANT"
   ],
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:17Z"
  }
 ]
}
<7>16:07:57.934 Sas.cpp post 48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:07:57.977 Sas.cpp post 48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:17Z"
  }
 ]
}
<6>16:07:57.977 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:07:57.977 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:07:57.978 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:07:58.078 Enb.cpp onData 48693 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:07:58.080 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:08:04.024 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:08:04.029 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:08:04.035 Sas.cpp post 48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -99
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -97
     }
    ]
   },
   "operationState": "AUTHORIZED"
  }
 ]
}
```

<7>16:08:04.040 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:24Z"
}
]
}
<7>16:08:04.040 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:04.083 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"measReportConfig": [
"RECEIVED_POWER_WITH_GRANT"
],
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:24Z"
}
]
}
<6>16:08:04.083 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:08:04.083 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:08:04.084 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:08:04.184 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:08:04.184 Enb.cpp onData 48693 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:08:10.130 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:08:10.136 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:08:10.142 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:10.146 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:30Z"
}
]
}
<7>16:08:10.146 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"measReport": {
"rcvdPowerMeasReports": [
{
"measBandwidth": 10000000,
"measFrequency": 3550000000,
"measRcvdPower": -95
},
{
"measBandwidth": 10000000,
"measFrequency": 3560000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3570000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3580000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3590000000,
"measRcvdPower": -97
},
{
"measBandwidth": 10000000,
"measFrequency": 3600000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3610000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3620000000,
"measRcvdPower": -97
},
{
"measBandwidth": 10000000,
"measFrequency": 3630000000,
"measRcvdPower": -98
},
{
"measBandwidth": 10000000,
"measFrequency": 3640000000,
"measRcvdPower": -95
},
{
"measBandwidth": 10000000,
"measFrequency": 3650000000,
"measRcvdPower": -97
},
{
"measBandwidth": 10000000,
"measFrequency": 3660000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3670000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3680000000,
"measRcvdPower": -94
},
{
"measBandwidth": 10000000,
"measFrequency": 3690000000,
"measRcvdPower": -98
}
]
},
"operationState": "AUTHORIZED"
}
]
}

<7>16:08:10.190 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:30Z"
}
]
}
<6>16:08:10.190 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:08:10.190 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:08:10.191 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:08:10.291 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:08:10.291 Enb.cpp onData 48693 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:08:16.237 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:08:16.242 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:08:16.248 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:16.252 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:36Z"
}
]
}
<7>16:08:16.252 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:16.295 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:36Z"
}
]
}
<6>16:08:16.295 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:08:16.295 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:08:16.296 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:08:16.396 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:08:16.396 Enb.cpp onData 48693 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:08:22.347 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:08:22.352 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:08:22.357 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:22.361 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:42Z"
}
]
}
<7>16:08:22.362 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:22.405 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "681638789",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:42Z"
}
]
}
<6>16:08:22.405 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:08:22.405 ManagerEnb.cpp command 48631 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:08:22.406 CbrsDaemon.cpp onLoop 48631 [34;1mINF[0m Listening for 5 seconds
<6>16:08:22.506 Enb.cpp onData 48663 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:08:22.506 Enb.cpp onData 48693 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:08:28.450 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:08:28.455 CbrsDaemon.cpp parseTree 48631 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:08:28.459 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"operationState": "AUTHORIZED"
}
]
}
<7>16:08:28.464 Sas.cpp post 48631 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "992128042",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:11:48Z"
}

```
 ]
}
<7>16:08:28.464 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:08:28.507 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:48Z"
  }
 ]
}
<6>16:08:28.507 ManagerEnb.cpp     command          48631 [34;1mINF[0m Sending tx_expire to
eNB(1), with expiration: 60000
<6>16:08:28.507 ManagerEnb.cpp     command          48631 [34;1mINF[0m Sending tx_expire to
eNB(2), with expiration: 60000
<6>16:08:28.508 CbrsDaemon.cpp     onLoop           48631 [34;1mINF[0m Listening for 5 seconds
<6>16:08:28.608 Enb.cpp            onData           48663 [34;1mINF[0m Answer received from eNB (1):
flags(129), {"message":"tx_expire"}
<6>16:08:28.608 Enb.cpp            onData           48693 [34;1mINF[0m Answer received from eNB (2):
flags(129), {"message":"tx_expire"}
<6>16:08:34.554 CbrsDaemon.cpp     parseTree        48631 [34;1mINF[0m Found CBRS Cell: cell_id
0, earfcn_dl 56040
<6>16:08:34.558 CbrsDaemon.cpp     parseTree        48631 [34;1mINF[0m Found CBRS Cell: cell_id
1, earfcn_dl 56140
<7>16:08:34.563 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:08:34.565 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "992128042",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:54Z"
  }
 ]
}
<7>16:08:34.565 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:08:34.608 Sas.cpp          post            48631 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "681638789",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:11:54Z"
  }
 ]
}
```

9.13 Log file for test case ID: WINNF.FT.D.RLQ.2


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:14:18.892 Sas.cpp          post            48982 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:14:18.892 Sas.cpp          post            48982 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:14:18.951 Sas.cpp          post            48982 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:14:18.952 CbrsDaemon.cpp      onLoop          48982 [34;1mINF[0m Listening for 5 seconds
<7>16:14:18.952 SpvLaunchdProxy.cpp  create          48982 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:14:18.952 SpvLaunchdProxy.cpp  create          48982 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:14:18.952 SpvLaunchdProxy.cpp  initSpvLaunchdProxy  48982 [36;1mDBG[0m SpvLaunchd
is running.
<7>16:14:18.953 SpvLaunchdProxy.cpp  logDBusMessage      48982 [36;1mDBG[0m
handleRequest: signal sender=org.freedesktop.DBus -> dest=:1.164 serial=2
path=/org/freedesktop/DBus; interface=org.freedesktop.DBus; member=NameAcquired;
signature=s
<7>16:14:18.953 SpvLaunchdProxy.cpp  dbusHandler       48982 [36;1mDBG[0m NameAcquired:
:1.164
<7>16:14:18.953 SpvLaunchdProxy.cpp  dbusHandler       48982 [36;1mDBG[0m Connection
name: :1.164
<6>16:14:19.994 CbrsDaemon.cpp      parseTree        48982 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:14:19.997 CbrsDaemon.cpp      parseTree        48982 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<6>16:14:20.001 CbrsDaemon.cpp      onLoop          48982 [34;1mINF[0m Listening for 5 seconds
<6>16:14:26.046 CbrsDaemon.cpp      parseTree        48982 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:14:26.049 CbrsDaemon.cpp      parseTree        48982 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<6>16:14:26.054 CbrsDaemon.cpp      onLoop          48982 [34;1mINF[0m Listening for 5 seconds
<7>16:14:27.277 SpvLaunchdProxy.cpp  logDBusMessage      48982 [36;1mDBG[0m
handleRequest: signal sender=:1.0 -> dest=(null) serial=311
path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:14:27.277 SpvLaunchdProxy.cpp  logActiveEnbs      48982 [36;1mDBG[0m Dump
activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,
"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:14:28.319 CbrsDaemon.cpp      parseTree        48982 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:14:28.322 CbrsDaemon.cpp      parseTree        48982 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<7>16:14:28.327 CbrsDaemon.cpp      persistEntities    48982 [36;1mDBG[0m Grant for cell 0,
belonging to eNB 1 created.
<6>16:14:28.328 ManagerCbsd.cpp     command          48982 [34;1mINF[0m Send command to
CBSD on fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052868,"us
er":"user"}, with timeout of 25
<6>16:14:28.328 ManagerCbsd.cpp     getResponseFromReque 48982 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2f1 : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052868,"us
er":"user"}
<6>16:14:28.361 ManagerCbsd.cpp     getResponseFromReque 48982 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2f1 : 5556] Socket response received (198320 bytes)
<7>16:14:28.364 Sas.cpp          post            48982 [36;1mDBG[0m {
```

```
"grantRequest": [
 {
  "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
  "measReport": {
   "rcvdPowerMeasReports": [
    {
     "measBandwidth": 10000000,
     "measFrequency": 3550000000,
     "measRcvdPower": -97
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3560000000,
     "measRcvdPower": -99
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3570000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3580000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3590000000,
     "measRcvdPower": -84
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3600000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3610000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3620000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3630000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3640000000,
     "measRcvdPower": -99
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3650000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3660000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3670000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3680000000,
     "measRcvdPower": -95
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3690000000,
     "measRcvdPower": -97
    }
   ]
  },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3635000000,
     "lowFrequency": 3625000000
    }
   }
  }
 ]
}
<7>16:14:28.371 Sas.cpp post 48982 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:14:28Z",
   "grantId": "862833809",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:14:28.371 Sas.cpp post 48982 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "862833809",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:14:28.414 Sas.cpp post 48982 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "862833809",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:17:48Z"
  }
 ]
}
<6>16:14:28.414 ManagerEnb.cpp command 48982 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:14:28.415 CbrsDaemon.cpp onLoop 48982 [34;1mINF[0m Listening for 5 seconds
<6>16:14:28.963 Enb.cpp onData 49014 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<7>16:14:29.664 SpvLaunchdProxy.cpp logDBusMessage 48982 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=313 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:14:29.665 SpvLaunchdProxy.cpp logActiveEnbs 48982 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONNEC
TED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:14:30.705 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:14:30.709 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:14:30.715 CbrsDaemon.cpp persistEntities 48982 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:14:30.715 Sas.cpp post 48982 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "862833809",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:14:30.719 Sas.cpp post 48982 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "862833809",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:17:50Z"
  }
 ]
}
```

<6>16:14:30.719 ManagerCbsd.cpp command 48982 [34;1mINF[0m Send command to CBSD on fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052870,"user":"user"}, with timeout of 25
<6>16:14:30.719 ManagerCbsd.cpp getResponseFromReque 48982 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559052870,"user":"user"}
<6>16:14:30.857 ManagerCbsd.cpp getResponseFromReque 48982 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Socket response received (198334 bytes)
<7>16:14:30.860 Sas.cpp post 48982 [36;1mDBG[0m {


```
 "grantRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -99
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -95
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -98
     }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3645000000,
     "lowFrequency": 3635000000
    }
   }
  }
 ]
}
```


<7>16:14:30.867 Sas.cpp post 48982 [36;1mDBG[0m {


```
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:14:30Z",
   "grantId": "33977948",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
```


<7>16:14:30.867 Sas.cpp post 48982 [36;1mDBG[0m {


```
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "33977948",
   "operationState": "GRANTED"
  }
 ]
}
```


<7>16:14:30.910 Sas.cpp post 48982 [36;1mDBG[0m {


```
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "33977948",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:17:50Z"
  }
 ]
}
```


<6>16:14:30.910 ManagerEnb.cpp command 48982 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:14:30.910 ManagerEnb.cpp command 48982 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:14:30.911 CbrsDaemon.cpp onLoop 48982 [34;1mINF[0m Listening for 5 seconds
<6>16:14:31.011 Enb.cpp onData 49014 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:14:31.050 Enb.cpp onData 49046 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:14:36.962 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:14:36.967 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:14:36.972 Sas.cpp post 48982 [36;1mDBG[0m {


```
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "862833809",
   "operationState": "AUTHORIZED"
  }
 ]
}
```

<7>16:14:36.976 Sas.cpp post 48982 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "862833809",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:17:56Z"
}
]
}
<7>16:14:36.976 Sas.cpp post 48982 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "33977948",
"operationState": "AUTHORIZED"
}
]
}
<7>16:14:37.019 Sas.cpp post 48982 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "33977948",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:17:56Z"
}
]
}
<6>16:14:37.019 ManagerEnb.cpp command 48982 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:14:37.019 ManagerEnb.cpp command 48982 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:14:37.020 CbrsDaemon.cpp onLoop 48982 [34;1mINF[0m Listening for 5 seconds
<6>16:14:37.120 Enb.cpp onData 49014 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:14:37.120 Enb.cpp onData 49046 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:14:41.041 SpvLaunchdProxy.cpp logDBusMessage 48982 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=315 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:14:41.041 SpvLaunchdProxy.cpp logActiveEnbs 48982 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:14:42.089 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:14:42.093 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:14:42.098 CbrsDaemon.cpp cleanupEntities 48982 [36;1mDBG[0m All grants belonging to CBRS cell 0, eNB 1 deleted (not enabled).
<6>16:14:42.099 Grant.cpp erase 48982 [34;1mINF[0m Grant relinquishment procedure for Grant 862833809
<7>16:14:42.099 Sas.cpp post 48982 [36;1mDBG[0m {
"relinquishmentRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "862833809"
}
]
}
<7>16:14:42.103 Sas.cpp post 48982 [36;1mDBG[0m {
"relinquishmentResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "862833809",
"response": {
"responseCode": 0
}
}
]
}
<7>16:14:42.104 Sas.cpp post 48982 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "33977948",
"operationState": "AUTHORIZED"
}
]
}
<7>16:14:42.147 Sas.cpp post 48982 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "33977948",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:18:02Z"
}
]
}
<6>16:14:42.147 ManagerEnb.cpp command 48982 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:14:42.148 CbrsDaemon.cpp onLoop 48982 [34;1mINF[0m Listening for 5 seconds
<6>16:14:42.248 Enb.cpp onData 49046 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:14:43.660 SpvLaunchdProxy.cpp logDBusMessage 48982 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=317 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:14:43.660 SpvLaunchdProxy.cpp logActiveEnbs 48982 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[]}
<6>16:14:44.704 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:14:44.708 CbrsDaemon.cpp parseTree 48982 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:14:44.713 CbrsDaemon.cpp cleanupEntities 48982 [36;1mDBG[0m All grants belonging to CBRS cell 1, eNB 2 deleted (not enabled).
<6>16:14:44.713 Grant.cpp erase 48982 [34;1mINF[0m Grant relinquishment procedure for Grant 33977948
<7>16:14:44.713 Sas.cpp post 48982 [36;1mDBG[0m {
"relinquishmentRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "33977948"
}
]
}
<7>16:14:44.717 Sas.cpp post 48982 [36;1mDBG[0m {
"relinquishmentResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "33977948",
"response": {
"responseCode": 0
}
}
]
}

9.14 Log file for test case ID: WINNF.FT.D.RLQ.4


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:18:30.425 Sas.cpp          post            49400 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:18:30.425 Sas.cpp          post            49400 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:18:30.484 Sas.cpp          post            49400 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:18:30.485 CbrsDaemon.cpp     onLoop          49400 [34;1mINF[0m Listening for 5 seconds
<7>16:18:30.485 SpvLaunchdProxy.cpp  create          49400 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:18:30.485 SpvLaunchdProxy.cpp  create          49400 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:18:30.485 SpvLaunchdProxy.cpp  initSpvLaunchdProxy  49400 [36;1mDBG[0m SpvLaunchd
is running.
<7>16:18:30.486 SpvLaunchdProxy.cpp  logDBusMessage     49400 [36;1mDBG[0m
handleRequest: signal sender=org.freedesktop.DBus -> dest=:1.167 serial=2
path=/org/freedesktop/DBus; interface=org.freedesktop.DBus; member=NameAcquired;
signature=s
<7>16:18:30.486 SpvLaunchdProxy.cpp  dbusHandler      49400 [36;1mDBG[0m NameAcquired:
:1.167
<7>16:18:30.486 SpvLaunchdProxy.cpp  dbusHandler      49400 [36;1mDBG[0m Connection
name: :1.167
<6>16:18:31.528 CbrsDaemon.cpp     parseTree       49400 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:18:31.531 CbrsDaemon.cpp     parseTree       49400 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<6>16:18:31.534 CbrsDaemon.cpp     onLoop          49400 [34;1mINF[0m Listening for 5 seconds
<7>16:18:35.255 SpvLaunchdProxy.cpp  logDBusMessage     49400 [36;1mDBG[0m
handleRequest: signal sender=:1.0 -> dest=(null) serial=322
path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:18:35.255 SpvLaunchdProxy.cpp  logActiveEnbs     49400 [36;1mDBG[0m Dump
activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,
"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:18:36.296 CbrsDaemon.cpp     parseTree       49400 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:18:36.299 CbrsDaemon.cpp     parseTree       49400 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<7>16:18:36.303 CbrsDaemon.cpp     persistEntities   49400 [36;1mDBG[0m Grant for cell 0,
belonging to eNB 1 created.
<6>16:18:36.304 ManagerCbsd.cpp    command         49400 [34;1mINF[0m Send command to
CBSD on fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053116,"us
er":"user"}, with timeout of 25
<6>16:18:36.304 ManagerCbsd.cpp    getResponseFromReque 49400 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2f1 : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053116,"us
er":"user"}
<6>16:18:36.336 ManagerCbsd.cpp    getResponseFromReque 49400 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2f1 : 5556] Socket response received (196414 bytes)
<7>16:18:36.338 Sas.cpp          post            49400 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
```

```
    {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -97
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3570000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3580000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3590000000,
      "measRcvdPower": -95
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3600000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3610000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3620000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3630000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3640000000,
      "measRcvdPower": -99
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3650000000,
      "measRcvdPower": -98
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3660000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3670000000,
      "measRcvdPower": -100
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3680000000,
      "measRcvdPower": -95
    },
    {
      "measBandwidth": 10000000,
      "measFrequency": 3690000000,
      "measRcvdPower": -97
    }
  ]
 },
 "operationParam": {
  "maxEirp": 0,
  "operationFrequencyRange": {
   "highFrequency": 3635000000,
   "lowFrequency": 3625000000
    }
   }
  }
 ]
}
<7>16:18:36.345 Sas.cpp          post            49400 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:18:36Z",
   "grantId": "369262936",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:18:36.345 Sas.cpp          post            49400 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:18:36.388 Sas.cpp          post            49400 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:21:56Z"
  }
 ]
}
<6>16:18:36.388 ManagerEnb.cpp     command          49400 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:18:36.389 CbrsDaemon.cpp     onLoop           49400 [34;1mINF[0m Listening for 5 seconds
<6>16:18:37.006 Enb.cpp            onData           49432 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<7>16:18:37.174 SpvLaunchdProxy.cpp  logDBusMessage     49400 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null)
serial=324 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess;
signature=s
<7>16:18:37.174 SpvLaunchdProxy.cpp  logActiveEnbs      49400 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONNECT
ED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:18:38.214 CbrsDaemon.cpp     parseTree        49400 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:18:38.218 CbrsDaemon.cpp     parseTree        49400 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:18:38.224 CbrsDaemon.cpp     persistEntities     49400 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:18:38.225 Sas.cpp          post            49400 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:18:38.229 Sas.cpp          post            49400 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:21:58Z"
  }
 ]
}
```

<6>16:18:38.229 ManagerCbsd.cpp command 49400 [34;1mINF[0m Send command to CBSD on fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053118,"user":"user"}, with timeout of 25
<6>16:18:38.229 ManagerCbsd.cpp getResponseFromReque 49400 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053118,"user":"user"}
<6>16:18:38.261 ManagerCbsd.cpp getResponseFromReque 49400 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Socket response received (195462 bytes)
<7>16:18:38.264 Sas.cpp post 49400 [36;1mDBG[0m {
"grantRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"measReport": {
"rcvdPowerMeasReports": [
{
"measBandwidth": 10000000,
"measFrequency": 3550000000,
"measRcvdPower": -96
},
{
"measBandwidth": 10000000,
"measFrequency": 3560000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3570000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3580000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3590000000,
"measRcvdPower": -97
},
{
"measBandwidth": 10000000,
"measFrequency": 3600000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3610000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3620000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3630000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3640000000,
"measRcvdPower": -95
},
{
"measBandwidth": 10000000,
"measFrequency": 3650000000,
"measRcvdPower": -98
},
{
"measBandwidth": 10000000,
"measFrequency": 3660000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3670000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3680000000,
"measRcvdPower": -94
},
{
"measBandwidth": 10000000,
"measFrequency": 3690000000,
"measRcvdPower": -99
}
]
},
"operationParam": {
"maxEirp": 0,
"operationFrequencyRange": {
"highFrequency": 3645000000,
"lowFrequency": 3635000000
}
}
}
]
}
<7>16:18:38.310 Sas.cpp post 49400 [36;1mDBG[0m {
"grantResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"channelType": "GAA",
"grantExpireTime": "2019-06-04T14:18:38Z",
"grantId": "291347170",
"heartbeatInterval": 60,
"response": {
"responseCode": 0
}
}
]
}
<7>16:18:38.310 Sas.cpp post 49400 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "291347170",
"operationState": "GRANTED"
}
]
}
<7>16:18:38.353 Sas.cpp post 49400 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "291347170",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:21:58Z"
}
]
}
<6>16:18:38.353 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:18:38.353 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:18:38.354 CbrsDaemon.cpp onLoop 49400 [34;1mINF[0m Listening for 5 seconds
<6>16:18:38.454 Enb.cpp onData 49432 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:18:38.493 Enb.cpp onData 49462 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:18:44.399 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:18:44.403 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:18:44.408 Sas.cpp post 49400 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "369262936",
"operationState": "AUTHORIZED"
}

```
 ]
}
<7>16:18:44.412 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:22:04Z"
  }
 ]
}
<7>16:18:44.412 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "291347170",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:18:44.456 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "291347170",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:22:04Z"
  }
 ]
}
<6>16:18:44.456 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to
eNB(1), with expiration: 60000
<6>16:18:44.456 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to
eNB(2), with expiration: 60000
<6>16:18:44.457 CbrsDaemon.cpp onLoop 49400 [34;1mINF[0m Listening for 5 seconds
<6>16:18:44.557 Enb.cpp onData 49432 [34;1mINF[0m Answer received from eNB (1):
flags(129), {"message":"tx_expire"}
<6>16:18:44.557 Enb.cpp onData 49462 [34;1mINF[0m Answer received from eNB (2):
flags(129), {"message":"tx_expire"}
<6>16:18:50.509 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell: cell_id
0, earfcn_dl 56040
<6>16:18:50.513 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell: cell_id
1, earfcn_dl 56140
<7>16:18:50.519 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:18:50.523 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:22:10Z"
  }
 ]
}
<7>16:18:50.523 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "291347170",
   "operationState": "AUTHORIZED"
  }
```

```
 ]
}
<7>16:18:50.567 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "291347170",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:22:10Z"
  }
 ]
}
<6>16:18:50.567 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to
eNB(1), with expiration: 60000
<6>16:18:50.567 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to
eNB(2), with expiration: 60000
<6>16:18:50.568 CbrsDaemon.cpp onLoop 49400 [34;1mINF[0m Listening for 5 seconds
<6>16:18:50.668 Enb.cpp onData 49432 [34;1mINF[0m Answer received from eNB
(1): flags(129), {"message":"tx_expire"}
<6>16:18:50.668 Enb.cpp onData 49462 [34;1mINF[0m Answer received from eNB
(2): flags(129), {"message":"tx_expire"}
<7>16:18:53.987 SpvLaunchdProxy.cpp logDBusMessage 49400 [36;1mDBG[0m
handleRequest: signal sender=:1.0 -> dest=(null) serial=326
path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StopProcess; signature=s
<7>16:18:53.987 SpvLaunchdProxy.cpp logActiveEnbs 49400 [36;1mDBG[0m Dump
activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,
"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:18:55.028 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:18:55.032 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<7>16:18:55.037 CbrsDaemon.cpp cleanupEntities 49400 [36;1mDBG[0m All grants
belonging to CBRS cell 0, eNB 1 deleted (not enabled).
<6>16:18:55.037 Grant.cpp erase 49400 [34;1mINF[0m Grant relinquishment
procedure for Grant 369262936
<7>16:18:55.037 Sas.cpp post 49400 [36;1mDBG[0m {
 "relinquishmentRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "369262936"
  }
 ]
}
<7>16:18:55.042 Sas.cpp post 49400 [36;1mDBG[0m {
 "relinquishmentResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 102,
    "responseData": [
     "grantId"
    ]
   }
  }
 ]
}
<3>16:18:55.042 Grant.cpp erase 49400 [31;1mERR[0m Relinquishment procedure
failed for Grant 369262936
<7>16:18:55.042 Sas.cpp post 49400 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "291347170",
   "operationState": "AUTHORIZED"
  }
 ]
}
```

<7>16:18:55.085 Sas.cpp post 49400 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "291347170",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:22:15Z"
}
]
}
<6>16:18:55.085 ManagerEnb.cpp command 49400 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:18:55.086 CbrsDaemon.cpp onLoop 49400 [34;1mINF[0m Listening for 5 seconds
<6>16:18:55.186 Enb.cpp onData 49462 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:18:55.996 SpvLaunchdProxy.cpp logDBusMessage 49400 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=328 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:18:55.996 SpvLaunchdProxy.cpp logActiveEnbs 49400 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[]}
<6>16:18:57.042 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:18:57.047 CbrsDaemon.cpp parseTree 49400 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:18:57.052 CbrsDaemon.cpp cleanupEntities 49400 [36;1mDBG[0m All grants belonging to CBRS cell 1, eNB 2 deleted (not enabled).
<6>16:18:57.053 Grant.cpp erase 49400 [34;1mINF[0m Grant relinquishment procedure for Grant 291347170
<7>16:18:57.053 Sas.cpp post 49400 [36;1mDBG[0m {
"relinquishmentRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "291347170"
}
]
}
<7>16:18:57.056 Sas.cpp post 49400 [36;1mDBG[0m {
"relinquishmentResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"response": {
"responseCode": 102,
"responseData": [
"grantId"
]
}
}
]
}

9.15 Log file for test case ID: WINNF.FT.D.RLQ.6

```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:23:56.557 Sas.cpp          post            49736 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:23:56.557 Sas.cpp          post            49736 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:23:56.616 Sas.cpp          post            49736 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:23:56.617 CbrsDaemon.cpp      onLoop          49736 [34;1mINF[0m Listening for 5 seconds
<7>16:23:56.617 SpvLaunchdProxy.cpp  create          49736 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:23:56.617 SpvLaunchdProxy.cpp  create          49736 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:23:56.617 SpvLaunchdProxy.cpp  initSpvLaunchdProxy  49736 [36;1mDBG[0m SpvLaunchd
is running.
<7>16:23:56.618 SpvLaunchdProxy.cpp  logDBusMessage     49736 [36;1mDBG[0m
handleRequest: signal sender=org.freedesktop.DBus -> dest=:1.169 serial=2
path=/org/freedesktop/DBus; interface=org.freedesktop.DBus; member=NameAcquired;
signature=s
<7>16:23:56.618 SpvLaunchdProxy.cpp  dbusHandler       49736 [36;1mDBG[0m NameAcquired:
:1.169
<7>16:23:56.618 SpvLaunchdProxy.cpp  dbusHandler       49736 [36;1mDBG[0m Connection
name: :1.169
<6>16:23:57.660 CbrsDaemon.cpp      parseTree       49736 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:23:57.663 CbrsDaemon.cpp      parseTree       49736 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<6>16:23:57.667 CbrsDaemon.cpp      onLoop          49736 [34;1mINF[0m Listening for 5 seconds
<6>16:24:03.713 CbrsDaemon.cpp      parseTree       49736 [34;1mINF[0m Found CBRS Cell:
cell_id 0, earfcn_dl 56040
<6>16:24:03.716 CbrsDaemon.cpp      parseTree       49736 [34;1mINF[0m Found CBRS Cell:
cell_id 1, earfcn_dl 56140
<7>16:24:03.721 CbrsDaemon.cpp      persistEntities    49736 [36;1mDBG[0m Grant for cell 0,
belonging to eNB 1 created.
<6>16:24:03.722 ManagerCbsd.cpp     command         49736 [34;1mINF[0m Send command to
CBSD on fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053443,"us
er":"user"}, with timeout of 25
<6>16:24:03.722 ManagerCbsd.cpp     getResponseFromReque 49736 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2f1 : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053443,"us
er":"user"}
<6>16:24:03.754 ManagerCbsd.cpp     getResponseFromReque 49736 [34;1mINF[0m
[fe80::72b3:d5ff:fe29:c2f1 : 5556] Socket response received (196751 bytes)
<7>16:24:03.758 Sas.cpp          post            49736 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
```

```
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3570000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3580000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -96
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -96
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -98
      }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3635000000,
     "lowFrequency": 3625000000
    }
   }
  }
 ]
}
```

```
<7>16:24:03.764 Sas.cpp post 49736 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:24:03Z",
   "grantId": "135551579",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:24:03.764 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:24:03.807 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:27:23Z"
  }
 ]
}
<6>16:24:03.808 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<7>16:24:03.808 SpvLaunchdProxy.cpp logDBusMessage 49736 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=332 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:24:03.808 SpvLaunchdProxy.cpp logActiveEnbs 49736 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONNEC
TED"}]}
<6>16:24:04.900 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:04.904 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:04.910 CbrsDaemon.cpp persistEntities 49736 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:24:04.910 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:24:04.914 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:27:24Z"
  }
 ]
}
<6>16:24:04.915 ManagerCbsd.cpp command 49736 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053444,"user":"user"}, with timeout of 25
<6>16:24:04.915 ManagerCbsd.cpp getResponseFromReque 49736 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Send
(timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053444,"user":"user"}
<6>16:24:04.948 ManagerCbsd.cpp getResponseFromReque 49736 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Socket
response received (196397 bytes)
<7>16:24:04.952 Sas.cpp post 49736 [36;1mDBG[0m {
```

```
"grantRequest": [
 {
  "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
  "measReport": {
   "rcvdPowerMeasReports": [
    {
     "measBandwidth": 10000000,
     "measFrequency": 3550000000,
     "measRcvdPower": -96
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3560000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3570000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3580000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3590000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3600000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3610000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3620000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3630000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3640000000,
     "measRcvdPower": -96
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3650000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3660000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3670000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3680000000,
     "measRcvdPower": -95
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3690000000,
     "measRcvdPower": -99
    }
   ]
  },
  "operationParam": {
   "maxEirp": 0,
   "operationFrequencyRange": {
    "highFrequency": 3645000000,
    "lowFrequency": 3635000000
   }
  }
 }
]
}
<7>16:24:04.997 Sas.cpp post 49736 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:24:04Z",
   "grantId": "772605572",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:24:04.997 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:24:05.040 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:27:25Z"
  }
 ]
}
<6>16:24:05.040 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration:
60000
<6>16:24:05.041 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<7>16:24:05.041 SpvLaunchdProxy.cpp logDBusMessage 49736 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=334 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:24:05.041 SpvLaunchdProxy.cpp logActiveEnbs 49736 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"C
ONNECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:24:05.180 Enb.cpp onData 49768 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<6>16:24:06.080 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:06.085 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:06.091 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:24:06.094 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:27:26Z"
  }
 ]
}
```

<7>16:24:06.095 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "772605572",
"operationState": "AUTHORIZED"
}
]
}
<7>16:24:06.137 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "772605572",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:27:26Z"
}
]
}
<6>16:24:06.137 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:24:06.137 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:24:06.138 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<6>16:24:06.238 Enb.cpp onData 49768 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:24:06.277 Enb.cpp onData 49797 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:24:12.183 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:12.187 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:12.192 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "135551579",
"operationState": "AUTHORIZED"
}
]
}
<7>16:24:12.196 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "135551579",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:27:32Z"
}
]
}
<7>16:24:12.196 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "772605572",
"operationState": "AUTHORIZED"
}
]
}
<7>16:24:12.239 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "772605572",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:27:32Z"
}
]
}
<6>16:24:12.239 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:24:12.239 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:24:12.240 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<6>16:24:12.340 Enb.cpp onData 49768 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:24:12.340 Enb.cpp onData 49797 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:24:18.286 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:18.290 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:18.295 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "135551579",
"operationState": "AUTHORIZED"
}
]
}
<7>16:24:18.299 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "135551579",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:27:38Z"
}
]
}
<7>16:24:18.299 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "772605572",
"operationState": "AUTHORIZED"
}
]
}
<7>16:24:18.341 Sas.cpp post 49736 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "772605572",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:27:38Z"
}
]
}
<6>16:24:18.341 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:24:18.341 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:24:18.342 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<6>16:24:18.442 Enb.cpp onData 49768 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:24:18.442 Enb.cpp onData 49797 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:24:24.446 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:24.450 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:24.455 CbrsDaemon.cpp cleanupEntities 49736 [36;1mDBG[0m All grants belonging to CBRS cell 0, eNB 1 deleted (not enabled).
<6>16:24:24.456 Grant.cpp erase 49736 [34;1mINF[0m Grant relinquishment procedure for Grant 135551579
<7>16:24:24.456 Sas.cpp post 49736 [36;1mDBG[0m {
"relinquishmentRequest": [

```
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "135551579"
  }
 ]
}
<7>16:24:24.460 Sas.cpp post 49736 [36;1mDBG[0m {
 "relinquishmentResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 103,
    "responseData": [
     "grantId"
    ]
   }
  }
 ]
}
<3>16:24:24.460 Grant.cpp erase 49736 [31;1mERR[0m Relinquishment procedure failed for Grant 135551579
<7>16:24:24.461 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:24:24.504 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:27:44Z"
  }
 ]
}
<6>16:24:24.504 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:24:24.505 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<7>16:24:24.505 SpvLaunchdProxy.cpp logDBusMessage 49736 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=336 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:24:24.505 SpvLaunchdProxy.cpp logActiveEnbs 49736 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:24:24.605 Enb.cpp onData 49797 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:24:25.546 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:25.550 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:25.555 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:24:25.559 Sas.cpp post 49736 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:27:45Z"
  }
 ]
}
<6>16:24:25.559 ManagerEnb.cpp command 49736 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:24:25.559 CbrsDaemon.cpp onLoop 49736 [34;1mINF[0m Listening for 5 seconds
<6>16:24:25.659 Enb.cpp onData 49797 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:24:26.570 SpvLaunchdProxy.cpp logDBusMessage 49736 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=338 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:24:26.570 SpvLaunchdProxy.cpp logActiveEnbs 49736 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[]}
<6>16:24:27.615 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:24:27.619 CbrsDaemon.cpp parseTree 49736 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:24:27.625 CbrsDaemon.cpp cleanupEntities 49736 [36;1mDBG[0m All grants belonging to CBRS cell 1, eNB 2 deleted (not enabled).
<6>16:24:27.625 Grant.cpp erase 49736 [34;1mINF[0m Grant relinquishment procedure for Grant 772605572
<7>16:24:27.625 Sas.cpp post 49736 [36;1mDBG[0m {
 "relinquishmentRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "772605572"
  }
 ]
}
<7>16:24:27.629 Sas.cpp post 49736 [36;1mDBG[0m {
 "relinquishmentResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 103,
    "responseData": [
     "grantId"
    ]
   }
  }
 ]
}
```

9.16 Log file for test case ID: WINNF.FT.D.DRG.2


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:26:19.502 Sas.cpp          post            49995 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:26:19.502 Sas.cpp          post            49995 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:26:19.576 Sas.cpp          post            49995 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:26:19.577 CbrsDaemon.cpp      onLoop          49995 [34;1mINF[0m Listening for 5 seconds
<7>16:26:19.577 SpvLaunchdProxy.cpp  create          49995 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:26:19.577 SpvLaunchdProxy.cpp  create          49995 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:26:19.578 SpvLaunchdProxy.cpp  initSpvLaunchdProxy  49995 [36;1mDBG[0m SpvLaunchd is running.
<7>16:26:19.578 SpvLaunchdProxy.cpp  logDBusMessage     49995 [36;1mDBG[0m handleRequest: signal
sender=org.freedesktop.DBus -> dest=:1.171 serial=2 path=/org/freedesktop/DBus; interface=org.freedesktop.DBus;
member=NameAcquired; signature=s
<7>16:26:19.578 SpvLaunchdProxy.cpp  dbusHandler       49995 [36;1mDBG[0m NameAcquired: :1.171
<7>16:26:19.578 SpvLaunchdProxy.cpp  dbusHandler       49995 [36;1mDBG[0m Connection name: :1.171
<6>16:26:20.620 CbrsDaemon.cpp      parseTree       49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:26:20.623 CbrsDaemon.cpp      parseTree       49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<6>16:26:20.627 CbrsDaemon.cpp      onLoop          49995 [34;1mINF[0m Listening for 5 seconds
<7>16:26:23.381 SpvLaunchdProxy.cpp  logDBusMessage     49995 [36;1mDBG[0m handleRequest: signal sender=:1.0
-> dest=(null) serial=342 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:26:23.381 SpvLaunchdProxy.cpp  logActiveEnbs      49995 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","sta
te":"CONNECTED"}]}
<6>16:26:24.429 CbrsDaemon.cpp      parseTree       49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:26:24.431 CbrsDaemon.cpp      parseTree       49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<7>16:26:24.436 CbrsDaemon.cpp      persistEntities     49995 [36;1mDBG[0m Grant for cell 0, belonging to eNB 1
created.
<6>16:26:24.436 ManagerCbsd.cpp     command         49995 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053584,"user":"user"}, with
timeout of 25
<6>16:26:24.436 ManagerCbsd.cpp     getResponseFromReque 49995 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053584,"user":"user"}
<6>16:26:24.468 ManagerCbsd.cpp     getResponseFromReque 49995 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Socket response received (198331 bytes)
<7>16:26:24.472 Sas.cpp          post            49995 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -98
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
```

```
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3570000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3580000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -84
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -99
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -83
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -95
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -97
      }
     ]
    },
    "operationParam": {
     "maxEirp": 0,
     "operationFrequencyRange": {
      "highFrequency": 3635000000,
      "lowFrequency": 3625000000
     }
    }
   }
  ]
}
```

```
<7>16:26:24.478 Sas.cpp          post            49995 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:26:24Z",
   "grantId": "156644594",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:26:24.479 Sas.cpp          post            49995 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:26:24.521 Sas.cpp          post            49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:29:44Z"
  }
 ]
}
<6>16:26:24.521 ManagerEnb.cpp      command          49995 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration:
60000
<6>16:26:24.522 CbrsDaemon.cpp      onLoop           49995 [34;1mINF[0m Listening for 5 seconds
<6>16:26:24.969 Enb.cpp             onData           50027 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<7>16:26:27.785 SpvLaunchdProxy.cpp  logDBusMessage     49995 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=344 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:26:27.785 SpvLaunchdProxy.cpp  logActiveEnbs       49995 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"C
ONNECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:26:28.832 CbrsDaemon.cpp      parseTree        49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:26:28.836 CbrsDaemon.cpp      parseTree        49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:26:28.843 CbrsDaemon.cpp      persistEntities  49995 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:26:28.843 Sas.cpp          post            49995 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:26:28.847 Sas.cpp          post            49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:29:48Z"
  }
 ]
}
```

<6>16:26:28.847 ManagerCbsd.cpp command 49995 [34;1mINF[0m Send command to CBSD on fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053588,"user":"user"}, with timeout of 25
<6>16:26:28.847 ManagerCbsd.cpp getResponseFromReque 49995 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053588,"user":"user"}
<6>16:26:28.879 ManagerCbsd.cpp getResponseFromReque 49995 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Socket response received (197326 bytes)
<7>16:26:28.882 Sas.cpp post 49995 [36;1mDBG[0m {
"grantRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"measReport": {
"rcvdPowerMeasReports": [
{
"measBandwidth": 10000000,
"measFrequency": 3550000000,
"measRcvdPower": -96
},
{
"measBandwidth": 10000000,
"measFrequency": 3560000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3570000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3580000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3590000000,
"measRcvdPower": -98
},
{
"measBandwidth": 10000000,
"measFrequency": 3600000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3610000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3620000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3630000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3640000000,
"measRcvdPower": -96
},
{
"measBandwidth": 10000000,
"measFrequency": 3650000000,
"measRcvdPower": -99
},
{
"measBandwidth": 10000000,
"measFrequency": 3660000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3670000000,
"measRcvdPower": -100
},
{
"measBandwidth": 10000000,
"measFrequency": 3680000000,
"measRcvdPower": -95
},
{
"measBandwidth": 10000000,
"measFrequency": 3690000000,
"measRcvdPower": -99
}
]
},
"operationParam": {
"maxEirp": 0,
"operationFrequencyRange": {
"highFrequency": 3645000000,
"lowFrequency": 3635000000
}
}
}
]
}
<7>16:26:28.928 Sas.cpp post 49995 [36;1mDBG[0m {
"grantResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"channelType": "GAA",
"grantExpireTime": "2019-06-04T14:26:28Z",
"grantId": "210735650",
"heartbeatInterval": 60,
"response": {
"responseCode": 0
}
}
]
}
<7>16:26:28.928 Sas.cpp post 49995 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "210735650",
"operationState": "GRANTED"
}
]
}
<7>16:26:28.970 Sas.cpp post 49995 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "210735650",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:29:48Z"
}
]
}
<6>16:26:28.970 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:26:28.970 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:26:28.971 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:26:29.070 Enb.cpp onData 50027 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:26:29.110 Enb.cpp onData 50059 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:26:35.016 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:26:35.021 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:26:35.027 Sas.cpp post 49995 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "156644594",
"operationState": "AUTHORIZED"

```
  }
 ]
}
<7>16:26:35.030 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:29:55Z"
  }
 ]
}
<7>16:26:35.031 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:26:35.073 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:29:55Z"
  }
 ]
}
<6>16:26:35.073 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:26:35.073 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:26:35.074 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:26:35.174 Enb.cpp onData 50027 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:26:35.174 Enb.cpp onData 50059 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:26:41.120 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:26:41.125 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:26:41.132 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:26:41.135 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:30:01Z"
  }
 ]
<7>16:26:41.136 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "operationState": "AUTHORIZED"
  }
}
 ]
}
<7>16:26:41.178 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:30:01Z"
  }
 ]
}
<6>16:26:41.178 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:26:41.178 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:26:41.179 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:26:41.279 Enb.cpp onData 50027 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:26:41.279 Enb.cpp onData 50059 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:26:44.899 SpvLaunchdProxy.cpp logDBusMessage 49995 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=346 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:26:44.899 SpvLaunchdProxy.cpp logActiveEnbs 49995 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:26:45.943 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:26:45.947 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:26:45.952 CbrsDaemon.cpp cleanupEntities 49995 [36;1mDBG[0m All grants belonging to CBRS cell 0, eNB 1 deleted (not enabled).
<6>16:26:45.953 Grant.cpp erase 49995 [34;1mINF[0m Grant relinquishment procedure for Grant 156644594
<7>16:26:45.953 Sas.cpp post 49995 [36;1mDBG[0m {
 "relinquishmentRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594"
  }
 ]
}
<7>16:26:45.956 Sas.cpp post 49995 [36;1mDBG[0m {
 "relinquishmentResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "156644594",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:26:45.957 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:26:45.999 Sas.cpp post 49995 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:30:05Z"
  }
 ]
}
```

```
<6>16:26:45.999 ManagerEnb.cpp command 49995 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:26:46.000 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:26:46.100 Enb.cpp onData 50059 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:26:47.211 SpvLaunchdProxy.cpp logDBusMessage 49995 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=348 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:26:47.211 SpvLaunchdProxy.cpp logActiveEnbs 49995 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[]}
<6>16:26:48.255 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:26:48.260 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:26:48.265 CbrsDaemon.cpp cleanupEntities 49995 [36;1mDBG[0m All grants belonging to CBRS cell 1, eNB 2 deleted (not enabled).
<6>16:26:48.266 Grant.cpp erase 49995 [34;1mINF[0m Grant relinquishment procedure for Grant 210735650
<7>16:26:48.266 Sas.cpp post 49995 [36;1mDBG[0m {
 "relinquishmentRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650"
  }
 ]
}
<7>16:26:48.269 Sas.cpp post 49995 [36;1mDBG[0m {
 "relinquishmentResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "210735650",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:26:48.270 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:26:54.317 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:26:54.321 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:26:54.327 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:00.374 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:00.378 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:00.384 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:06.430 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:06.434 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:06.439 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:12.488 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:12.492 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:12.498 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:18.548 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:18.552 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:18.557 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:24.602 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:24.606 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:24.612 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:30.660 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:30.665 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:30.670 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:38.217 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:38.222 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:38.228 CbrsDaemon.cpp onLoop 49995 [34;1mINF[0m Listening for 5 seconds
<6>16:27:44.251 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:27:44.251 CbrsDaemon.cpp parseTree 49995 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:27:44.251 Cbsd.cpp erase 49995 [34;1mINF[0m Deregistration procedure for CBSD XM2-X19AX35M2Mock-SAS1012482003
<7>16:27:44.251 Sas.cpp post 49995 [36;1mDBG[0m {
 "deregistrationRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003"
  }
 ]
}
<7>16:27:44.254 Sas.cpp post 49995 [36;1mDBG[0m {
 "deregistrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:27:44.254 CbrsDaemon.cpp cleanupEntities 49995 [36;1mDBG[0m CBSD 1012482003 deleted.
<6>16:27:44.255 Cbsd.cpp erase 49995 [34;1mINF[0m Deregistration procedure for CBSD XM2-XAF2335M2Mock-SAS1012482006
<7>16:27:44.255 Sas.cpp post 49995 [36;1mDBG[0m {
 "deregistrationRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006"
  }
 ]
}
<7>16:27:44.297 Sas.cpp post 49995 [36;1mDBG[0m {
 "deregistrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
```

9.17 Log file for test case ID: WINNF.FT.D.DRG.4


```
{
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482003",
   "fccId": "XM2-X19AX35M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:30:35.329 Sas.cpp post 50255 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:30:35.329 Sas.cpp post 50255 [36;1mDBG[0m {
 "registrationRequest": [
  {
   "airInterface": {
    "radioTechnology": "E_UTRA"
   },
   "callSign": "?",
   "cbsdCategory": "A",
   "cbsdInfo": {
    "firmwareVersion": "v2.0.5",
    "hardwareVersion": "v1.0.45",
    "model": "CPRI_DEVICE-XXX",
    "softwareVersion": "v1.2.1",
    "vendor": "JMA Wireless"
   },
   "cbsdSerialNumber": "1012482006",
   "fccId": "XM2-XAF2335M2",
   "installationParam": {
    "antennaAzimuth": 70,
    "antennaBeamwidth": 45,
    "antennaDowntilt": 36,
    "antennaGain": 0,
    "antennaModel": "CPRI_DEVICE-XXX-ext-antenna",
    "eirpCapability": 15,
    "height": 15.0,
    "heightType": "AMSL",
    "horizontalAccuracy": 49,
    "indoorDeployment": true,
    "latitude": 43.09,
    "longitude": -76.15,
    "verticalAccuracy": 2
   },
   "measCapability": [
    "RECEIVED_POWER_WITH_GRANT",
    "RECEIVED_POWER_WITHOUT_GRANT"
   ],
   "userId": "abc"
  }
 ]
}
<7>16:30:35.388 Sas.cpp post 50255 [36;1mDBG[0m {
 "registrationResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:30:35.389 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<7>16:30:35.389 SpvLaunchdProxy.cpp create 50255 [36;1mDBG[0m Added match-rule:
"sender='com.jmawireless.jsoft.SpvLaunchd',interface='com.jmawireless.jsoft.SpvLaunchd'"
<7>16:30:35.389 SpvLaunchdProxy.cpp create 50255 [36;1mDBG[0m Added match-rule:
"sender='org.freedesktop.DBus',interface='org.freedesktop.DBus'"
<7>16:30:35.389 SpvLaunchdProxy.cpp initSpvLaunchdProxy 50255 [36;1mDBG[0m SpvLaunchd is running.
<7>16:30:35.390 SpvLaunchdProxy.cpp logDBusMessage 50255 [36;1mDBG[0m handleRequest: signal
sender=org.freedesktop.DBus -> dest=:1.173 serial=2 path=/org/freedesktop/DBus;
interface=org.freedesktop.DBus; member=NameAcquired; signature=s
<7>16:30:35.390 SpvLaunchdProxy.cpp dbusHandler 50255 [36;1mDBG[0m NameAcquired: :1.173
<7>16:30:35.390 SpvLaunchdProxy.cpp dbusHandler 50255 [36;1mDBG[0m Connection name: :1.173
<6>16:30:36.432 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:30:36.435 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<6>16:30:36.439 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<7>16:30:39.543 SpvLaunchdProxy.cpp logDBusMessage 50255 [36;1mDBG[0m handleRequest: signal
sender=:1.0 -> dest=(null) serial=352 path=/com/jmawireless/jsoft/SpvLaunchd;
interface=com.jmawireless.jsoft.SpvLaunchd; member=StartProcess; signature=s
<7>16:30:39.543 SpvLaunchdProxy.cpp logActiveEnbs 50255 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"",
"state":"CONNECTED"}]}
<6>16:30:40.584 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl
56040
<6>16:30:40.587 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl
56140
<7>16:30:40.592 CbrsDaemon.cpp persistEntities 50255 [36;1mDBG[0m Grant for cell 0, belonging to eNB 1
created.
<6>16:30:40.593 ManagerCbsd.cpp command 50255 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2f1:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053840,"user":"user"}, with
timeout of 25
<6>16:30:40.593 ManagerCbsd.cpp getResponseFromReque 50255 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Send (timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053840,"user":"user"}
<6>16:30:40.625 ManagerCbsd.cpp getResponseFromReque 50255 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2f1 :
5556] Socket response received (198098 bytes)
<7>16:30:40.628 Sas.cpp post 50255 [36;1mDBG[0m {
 "grantRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "measReport": {
    "rcvdPowerMeasReports": [
     {
      "measBandwidth": 10000000,
      "measFrequency": 3550000000,
      "measRcvdPower": -96
     },
     {
      "measBandwidth": 10000000,
      "measFrequency": 3560000000,
      "measRcvdPower": -100
```

```
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3570000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3580000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3590000000,
        "measRcvdPower": -96
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3600000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3610000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3620000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3630000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3640000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3650000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3660000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3670000000,
        "measRcvdPower": -100
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3680000000,
        "measRcvdPower": -95
      },
      {
        "measBandwidth": 10000000,
        "measFrequency": 3690000000,
        "measRcvdPower": -98
      }
    ]
   },
   "operationParam": {
    "maxEirp": 0,
    "operationFrequencyRange": {
     "highFrequency": 3635000000,
     "lowFrequency": 3625000000
    }
   }
  }
 ]
}
```

```
<7>16:30:40.635 Sas.cpp          post            50255 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:30:40Z",
   "grantId": "387385356",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:30:40.635 Sas.cpp          post            50255 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "387385356",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:30:40.677 Sas.cpp          post            50255 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "387385356",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:34:00Z"
  }
 ]
}
<6>16:30:40.677 ManagerEnb.cpp     command          50255 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:30:40.678 CbrsDaemon.cpp     onLoop           50255 [34;1mINF[0m Listening for 5 seconds
<6>16:30:40.924 Enb.cpp            onData           50287 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<7>16:30:41.463 SpvLaunchdProxy.cpp  logDBusMessage     50255 [36;1mDBG[0m handleRequest: signal sender=:1.0 ->
dest=(null) serial=354 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd;
member=StartProcess; signature=s
<7>16:30:41.463 SpvLaunchdProxy.cpp  logActiveEnbs      50255 [36;1mDBG[0m Dump activeEnbs_ map:
{"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":0,"cell_key":1,"locked":false}],"enb_key":1,"invalid_cfg":"","state":"CONN
ECTED"},{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:30:42.504 CbrsDaemon.cpp     parseTree        50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:30:42.509 CbrsDaemon.cpp     parseTree        50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:30:42.516 CbrsDaemon.cpp     persistEntities  50255 [36;1mDBG[0m Grant for cell 1, belonging to eNB 2 created.
<7>16:30:42.517 Sas.cpp          post            50255 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "387385356",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:30:42.521 Sas.cpp          post            50255 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "387385356",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:34:02Z"
  }
 ]
}
<6>16:30:42.521 ManagerCbsd.cpp    command          50255 [34;1mINF[0m Send command to CBSD on
fe80::72b3:d5ff:fe29:c2ef:
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053842,"user":"user"}, with timeout of 25
<6>16:30:42.521 ManagerCbsd.cpp    getResponseFromReque 50255 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Send
(timeout 25 seconds):
{"attributes":{},"operation":"get","path":"/power_vectors","type":"request","uid":1559053842,"user":"user"}
<6>16:30:42.553 ManagerCbsd.cpp    getResponseFromReque 50255 [34;1mINF[0m [fe80::72b3:d5ff:fe29:c2ef : 5556] Socket
response received (197191 bytes)
<7>16:30:42.556 Sas.cpp          post            50255 [36;1mDBG[0m {
```

```
"grantRequest": [
 {
  "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
  "measReport": {
   "rcvdPowerMeasReports": [
    {
     "measBandwidth": 10000000,
     "measFrequency": 3550000000,
     "measRcvdPower": -95
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3560000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3570000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3580000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3590000000,
     "measRcvdPower": -99
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3600000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3610000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3620000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3630000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3640000000,
     "measRcvdPower": -95
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3650000000,
     "measRcvdPower": -98
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3660000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3670000000,
     "measRcvdPower": -100
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3680000000,
     "measRcvdPower": -94
    },
    {
     "measBandwidth": 10000000,
     "measFrequency": 3690000000,
     "measRcvdPower": -100
    }
   ]
  },
  "operationParam": {
   "maxEirp": 0,
   "operationFrequencyRange": {
    "highFrequency": 3645000000,
    "lowFrequency": 3635000000
   }
  }
 }
]
}
<7>16:30:42.602 Sas.cpp post 50255 [36;1mDBG[0m {
 "grantResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "channelType": "GAA",
   "grantExpireTime": "2019-06-04T14:30:42Z",
   "grantId": "121515067",
   "heartbeatInterval": 60,
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<7>16:30:42.602 Sas.cpp post 50255 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "121515067",
   "operationState": "GRANTED"
  }
 ]
}
<7>16:30:42.644 Sas.cpp post 50255 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "121515067",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:34:02Z"
  }
 ]
}
<6>16:30:42.644 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration:
60000
<6>16:30:42.644 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration:
60000
<6>16:30:42.645 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:30:42.744 Enb.cpp onData 50287 [34;1mINF[0m Answer received from eNB (1): flags(129),
{"message":"tx_expire"}
<6>16:30:42.784 Enb.cpp onData 50317 [34;1mINF[0m Answer received from eNB (2): flags(129),
{"message":"tx_expire"}
<6>16:30:48.696 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:30:48.700 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:30:48.705 Sas.cpp post 50255 [36;1mDBG[0m {
 "heartbeatRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "387385356",
   "operationState": "AUTHORIZED"
  }
 ]
}
<7>16:30:48.709 Sas.cpp post 50255 [36;1mDBG[0m {
 "heartbeatResponse": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
   "grantId": "387385356",
   "response": {
    "responseCode": 0
   },
   "transmitExpireTime": "2019-05-28T14:34:08Z"
  }
 ]
}
```

<7>16:30:48.709 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "121515067",
"operationState": "AUTHORIZED"
}
]
}
<7>16:30:48.752 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "121515067",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:34:08Z"
}
]
}
<6>16:30:48.752 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:30:48.752 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:30:48.753 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:30:48.853 Enb.cpp onData 50287 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:30:48.853 Enb.cpp onData 50317 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<6>16:30:54.798 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:30:54.804 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:30:54.810 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "387385356",
"operationState": "AUTHORIZED"
}
]
<7>16:30:54.814 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "387385356",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:34:14Z"
}
]
}
<7>16:30:54.814 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "121515067",
"operationState": "AUTHORIZED"
}
]
}
}
<7>16:30:54.856 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "121515067",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:34:14Z"
}
]
}
<6>16:30:54.856 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(1), with expiration: 60000
<6>16:30:54.856 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:30:54.857 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:30:54.957 Enb.cpp onData 50287 [34;1mINF[0m Answer received from eNB (1): flags(129), {"message":"tx_expire"}
<6>16:30:54.957 Enb.cpp onData 50317 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}
<7>16:30:58.376 SpvLaunchdProxy.cpp logDBusMessage 50255 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=356 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:30:58.376 SpvLaunchdProxy.cpp logActiveEnbs 50255 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[{"cell_status":[{"cell_id":1,"cell_key":2,"locked":false}],"enb_key":2,"invalid_cfg":"","state":"CONNECTED"}]}
<6>16:30:59.417 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:30:59.422 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:30:59.426 CbrsDaemon.cpp cleanupEntities 50255 [36;1mDBG[0m All grants belonging to CBRS cell 0, eNB 1 deleted (not enabled).
<6>16:30:59.427 Grant.cpp erase 50255 [34;1mINF[0m Grant relinquishment procedure for Grant 387385356
<7>16:30:59.427 Sas.cpp post 50255 [36;1mDBG[0m {
"relinquishmentRequest": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "387385356"
}
]
}
<7>16:30:59.430 Sas.cpp post 50255 [36;1mDBG[0m {
"relinquishmentResponse": [
{
"cbsdId": "XM2-X19AX35M2Mock-SAS1012482003",
"grantId": "387385356",
"response": {
"responseCode": 0
}
}
]
}
<7>16:30:59.431 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatRequest": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "121515067",
"operationState": "AUTHORIZED"
}
]
}
<7>16:30:59.473 Sas.cpp post 50255 [36;1mDBG[0m {
"heartbeatResponse": [
{
"cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
"grantId": "121515067",
"response": {
"responseCode": 0
},
"transmitExpireTime": "2019-05-28T14:34:19Z"
}
]
}
<6>16:30:59.473 ManagerEnb.cpp command 50255 [34;1mINF[0m Sending tx_expire to eNB(2), with expiration: 60000
<6>16:30:59.474 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:30:59.574 Enb.cpp onData 50317 [34;1mINF[0m Answer received from eNB (2): flags(129), {"message":"tx_expire"}

<7>16:31:00.384 SpvLaunchdProxy.cpp logDBusMessage 50255 [36;1mDBG[0m handleRequest: signal sender=:1.0 -> dest=(null) serial=358 path=/com/jmawireless/jsoft/SpvLaunchd; interface=com.jmawireless.jsoft.SpvLaunchd; member=StopProcess; signature=s
<7>16:31:00.384 SpvLaunchdProxy.cpp logActiveEnbs 50255 [36;1mDBG[0m Dump activeEnbs_ map: {"admin_status":"UP","enbs":[]}
<6>16:31:01.428 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:01.433 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<7>16:31:01.438 CbrsDaemon.cpp cleanupEntities 50255 [36;1mDBG[0m All grants belonging to CBRS cell 1, eNB 2 deleted (not enabled).
<6>16:31:01.439 Grant.cpp erase 50255 [34;1mINF[0m Grant relinquishment procedure for Grant 121515067
<7>16:31:01.439 Sas.cpp post 50255 [36;1mDBG[0m {
 "relinquishmentRequest": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "121515067"
  }
 ]
}
<7>16:31:01.442 Sas.cpp post 50255 [36;1mDBG[0m {
 "relinquishmentResponse": [
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006",
   "grantId": "121515067",
   "response": {
    "responseCode": 0
   }
  }
 ]
}
<6>16:31:01.443 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:07.492 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:07.496 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:07.501 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:13.550 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:13.554 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:13.560 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:19.607 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:19.611 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:19.617 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:25.659 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:25.663 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:25.669 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:31.715 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:31.719 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:31.724 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:37.772 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:37.777 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:37.783 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:43.827 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:43.831 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:43.837 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:49.884 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:49.888 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:49.894 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:31:55.942 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:31:55.946 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:31:55.951 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:32:02.002 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:32:02.007 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:32:02.013 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:32:08.060 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:32:08.064 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:32:08.070 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:32:14.119 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:32:14.123 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:32:14.129 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:32:22.354 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:32:22.359 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:32:22.365 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:32:28.413 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:32:28.417 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:32:28.423 CbrsDaemon.cpp onLoop 50255 [34;1mINF[0m Listening for 5 seconds
<6>16:32:34.448 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 0, earfcn_dl 56040
<6>16:32:34.448 CbrsDaemon.cpp parseTree 50255 [34;1mINF[0m Found CBRS Cell: cell_id 1, earfcn_dl 56140
<6>16:32:34.448 Cbsd.cpp erase 50255 [34;1mINF[0m Deregistration procedure for CBSD XM2-X19AX35M2Mock-SAS1012482003
<7>16:32:34.448 Sas.cpp post 50255 [36;1mDBG[0m {
 "deregistrationRequest": [
  {
   "cbsdId": "XM2-X19AX35M2Mock-SAS1012482003"
  }
 ]
}
<7>16:32:34.451 Sas.cpp post 50255 [36;1mDBG[0m {
 "deregistrationResponse": [
  {
   "response": {
    "responseCode": 102
   }
  }
 ]
}
<3>16:32:34.451 Cbsd.cpp erase 50255 [31;1mERR[0m Deregistration procedure failed for CSBD XM2-X19AX35M2Mock-SAS1012482003
<7>16:32:34.451 CbrsDaemon.cpp cleanupEntities 50255 [36;1mDBG[0m CBSD 1012482003 deleted.
<6>16:32:34.452 Cbsd.cpp erase 50255 [34;1mINF[0m Deregistration procedure for CBSD XM2-XAF2335M2Mock-SAS1012482006
<7>16:32:34.452 Sas.cpp post 50255 [36;1mDBG[0m {
 "deregistrationRequest"
  {
   "cbsdId": "XM2-XAF2335M2Mock-SAS1012482006"
  }
 ]
}
<7>16:32:34.494 Sas.cpp post 50255 [36;1mDBG[0m {
 "deregistrationResponse": [
  {
   "response": {
    "responseCode": 102
   }
  }
 ]
}: [

END OF REPORT