

OpenCable Test Plans

Tuning Adapter Interface UDCP Acceptance Test Plan

TP-ATP-TRIF-UDCP-D05-080509

Draft

Notice

This Test Plan document is furnished by Cable Television Laboratories, Inc., as part of the OpenCable process on an “AS-IS” basis and CableLabs or other participating entity does not provide any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose. This document is copyrighted by CableLabs. Anyone designing, manufacturing, distributing, or servicing products, or providing services, based upon this document, is responsible for obtaining all intellectual property licenses for technology referenced in this document. **CableLabs shall have no liability for any party’s implementation of this document without any such required licenses.** Distribution of this document is restricted pursuant to the terms of separate access agreements negotiated with each of the participating entities.

© Copyright 2008 Cable Television Laboratories, Inc.

All rights reserved.

Document Status Sheet

Document Control Number:	TP-ATP-TRIF-UDCP-D05-080507			
Reference:	Acceptance Test Plan			
Date:	May 9, 2008			
Status Code:	Work in Process	Draft	Issued	Closed
Distribution Restrictions:	CableLabs® & Members Only	CableLabs®, Members, and Vendors Only	Public	

Key to Document Status Codes

Work in Process	An incomplete document designed to guide discussion and generate feedback, which may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs

Trademarks

DOCSIS®, eDOCSIS™, PacketCable™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, M-CMTS™, and CableLabs® are trademarks of Cable Television Laboratories, Inc.

Revision History:	TP-ATP-TRIF-UDCP-D05-080509	Added USB Data Rate Test
	TP-ATP-TRIF-UDCP-D04-080507	
	TP-ATP-TRIF-UDCP-D03-080403	
	TP-ATP-TRIF-UDCP-D02-080307	
	TP-ATP-TRIF-UDCP-D01-080229	
	TP-ATP-OC-TRIF-D01-071228	

Contents

1	TUNING RESOLVER INTERFACE ACCEPTANCE TEST PROCEDURES	6
1.1	<i>PURPOSE AND SCOPE</i>	6
1.2	<i>TEST DEFINITIONS</i>	6
1.3	<i>EQUIPMENT LIST</i>	6
1.4	<i>VENDOR DOCUMENTATION PACKAGE</i>	7
2	TEST PROCEDURE SETUP	8
3	TEST: TRIF INITIALIZATION	9
3.1	<i>POSITIVE TESTS</i>	9
3.1.1	<i>Testcase: TRIF init</i>	9
3.1.2	<i>Testcase: SCTE65 SI usage</i>	10
3.2	<i>NEGATIVE TESTS</i>	11
3.2.1	<i>Testcase: No tr_init_rsp() sent</i>	11
3.2.2	<i>Testcase: TR firmware upgrade reported in tr_init_rsp()</i>	11
3.2.3	<i>Testcase: TR not ready reported in tr_init_rsp()</i>	12
3.2.4	<i>Testcase: Authentication Failure Test</i>	12
4	TEST: CHANNEL TABLE DELIVERY	13
4.1.1	<i>Testcase: Channel table delivery and update</i>	13
4.1.2	<i>Testcase: Channel Table Update</i>	14
4.1.3	<i>Testcase: Large Channel table delivery</i>	15
5	TEST: TUNING TESTS	16
5.1.1	<i>Testcase: Tuner use status test</i>	16
5.1.2	<i>Testcase: Invalid HMAC Key Test</i>	17
5.1.3	<i>Testcase: Resolve Tuning Update Test</i>	19
6	TEST: USER ACTIVITY TESTS	22
6.1.1	<i>Testcase: Inactivity dialog test</i>	22
6.1.2	<i>Testcase: Unspecified Message Test</i>	22
6.1.3	<i>Testcase: User activity test</i>	24
7	TEST: EMERGENCY ALERT SYSTEM	25
8	TEST: USB INTERFACE TESTS	27
8.1.1	<i>Testcase: Device Descriptor Test</i>	27
8.1.2	<i>Testcase: Data Transfer Rate</i>	28
9	TEST: MECHANICAL	31
9.1.1	<i>Testcase: Visual Inspection of Keyed Connector</i>	31
10	TEST: DIAGNOSTIC MESSAGES	32

10.1	TESTCASE: POSITIVE TEST – DIAGNOSTIC REQUEST	32
11	TEST: DEVICE INTEROPERABILITY.....	33
11.1	INTEROPERABILITY TESTS EQUIPMENT SETUP	33
11.2	INTEROPERABILITY TESTS	33
11.2.1	<i>Testcase: Power-up scenarios and channel map acquisition</i>	<i>34</i>
11.2.2	<i>Testcase: SDV manager interaction</i>	<i>34</i>
11.2.3	<i>Testcase: TR diagnostics</i>	<i>35</i>
11.2.4	<i>Testcase: EAS messaging.....</i>	<i>35</i>
11.2.5	<i>Testcase: CableCard Firmware Upgrade.....</i>	<i>36</i>
11.2.6	<i>Testcase: Head end Interoperability.....</i>	<i>36</i>
11.2.7	<i>Testcase: Tuning Adapter Upgrade.....</i>	<i>38</i>
12	TEST: VENDOR DOCUMENTATION.....	39
12.1.1	<i>Testcase: Required documentation</i>	<i>39</i>
12.1.2	<i>Testcase: Optional Requirements</i>	<i>40</i>
ANNEX A	TRIS UDCP ATP SCRIPTS	41
ANNEX B	MOTOROLA SDV HEAD-END SPECIFICS	42
ANNEX C	SA/CISCO HEAD-END SDV SPECIFICS.....	43

This page intentionally left blank.

1 TUNING RESOLVER INTERFACE ACCEPTANCE TEST PROCEDURES

1.1 Purpose and Scope

The Acceptance Test Plan (ATP) is a compendium of test procedures that may be used to demonstrate that the “Device Under Test” complies with the Tuning Resolver Interface specification. CableLabs and OpenCable Vendors have developed these procedures to facilitate the product development and certification efforts of OpenCable equipment suppliers.

The ATP techniques are not necessarily the only methods for demonstrating compliance. It may be possible to demonstrate compliance using other procedures. CableLabs will tend to use ATP procedures when auditing applications for certification on behalf of the OpenCable certification authority.

The ATP procedures are “black box” tests; that is, they do not require opening the equipment under test to access special test points or to invoke test modes of operation. There are requirements that cannot be verified by black box techniques and supplier-proprietary procedures are required to test such requirements. These supplier-proprietary test procedures are beyond the scope of the ATP.

The ATP procedures are not complete. Subsequent revisions of this document may contain additional test procedures. However, for various reasons, additional procedures will always be required to show OpenCable compliance to some requirements. Furthermore, some procedures in some revisions of the ATP are incompletely specified outlines of tests that are under development.

1.2 Test Definitions

APDU	Application Protocol Data Unit
DUT	Device under Test
EAS	Emergency Alert System
RCU	Remote Control Unit
SDV	Switched Digital Video
SPDU	Session Protocol Data Unit
TR	Tuning Resolver
TRIF	Tuning Resolver Interface
TRIS	Tuning Resolver Interface Simulator
UDCP	Uni-directional Cable Product
USB	Universal Serial Bus

1.3 Equipment List

Some tests may have more than one brand of equipment that may be used to complete the test and may be substituted when applicable.

Model Number	QTY	Description
TRIS v1.1.1 Build 11177	1	Tuning Resolver Interface Simulator (TRIS)
	1	HPNX PRO hardware/software/laptop (see below for laptop requirements)
Ver. 1.7.2007.12191	1	HPNX PRO Software

Minimum system requirements for HPNX PRO PC (or current)

OS – Windows 2000, XP, or Windows Server 2003 with latest service pack updates
CPU – Pentium 4, 1.7Ghz or faster
Hard Disk - 1 Gig free disk space for software and stream libraries
Video 1024x768, 24 bit color, video controller capable of real time MPEG2 decoding (Newer NVidia or ATI cards will do).
RAM 256MB
CD - ROM for installation
Network (10/100 base T) Ethernet card or built in
MPEG@ Software Decoder For local rendering of selected channel program. The decoder needs to be installed for the HPNX PRO to work. Software MPEG2 decoder from the following companies. Nvidia, InterVideo, CyberLink (NOTE: The HPNX PRO will not install if there is not a valid MPEG2 Software Decoder installed on the computer).
Sound - a sound card or built in installed for local audio rendering of selected program.
Appropriate cabling.

Catalyst SBAE-30	1	USB Protocol Analyzer
Cable Headend	1	Cable headend with CableCARD and SDV support

Note1: Ability to provide cables, adapters, splitters; custom wiring to extender cards is required.

Note2: This Equipment List is subject to changes, additions, or alternatives. CableLabs does not endorse such Equipment in any manner.

1.4 Vendor Documentation Package

A Vendor's complete product verification submission SHALL include both (1) vendor submitted documentation for each REQ item that is not identified in this ATP and (2) Pass/Fail test results for all REQs in the ATP.

2 TEST PROCEDURE SETUP

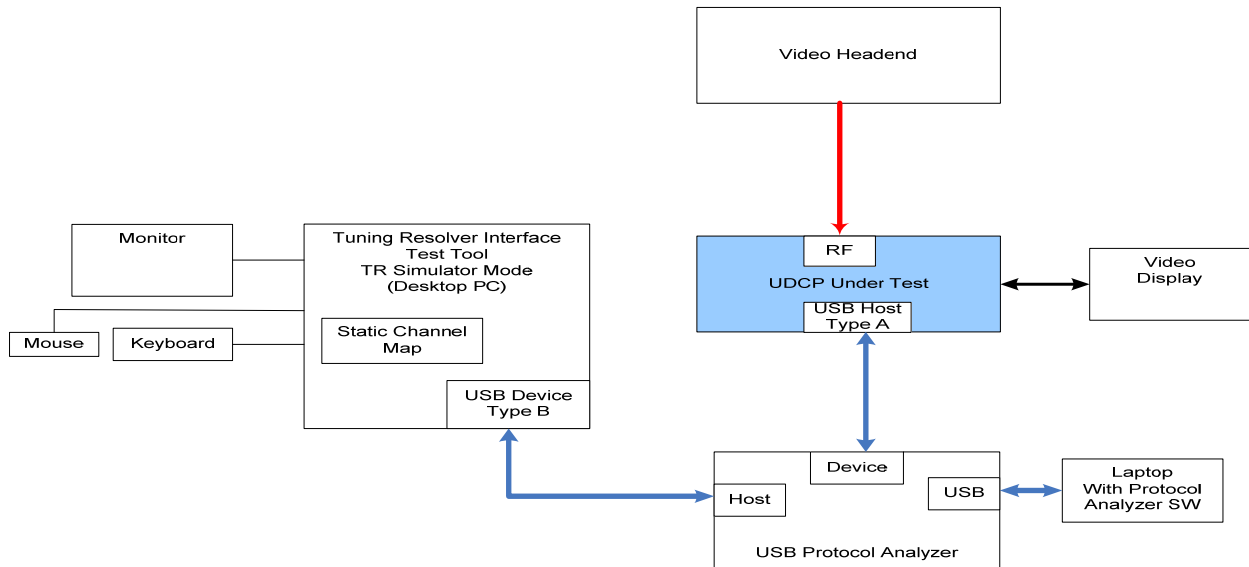


Figure 1 – Tuning Adapter UDCP test configuration

To test the TRIF messaging on the UDCP, the UDCP is connected to Video Headend (currently SA or Motorola), and connected via USB to the Tuning Resolver Interface Simulator (TRIS). The UDCP will be able to tune to video channels, and tune to encrypted video channels if a CableCARD is inserted and the TRIS is configured to provide the tuning information for those channels. The TRIS does not connect to the headend, and it does not interact with the SDV equipment in the headend. Actual switched channel testing is performed during interoperability testing.

3 TEST: TRIF INITIALIZATION

3.1 Positive tests

3.1.1 Testcase: TRIF init

Step#	Procedure	Pass/Fail	REQ
1.	From vendor documentation extract the following: <ul style="list-style-type: none"> - number of tuners (_____) - supported video codecs (_____) - supported audio codecs (_____) - upper frequency tuning range (_____) - hardware_version_num (_____) - str_byte (_____) 	N/A	
2.	Power-cycle the DUT and wait until it becomes operational.	N/A	
3.	Start TRIS application in TR simulation mode (t).	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	From TRIS trace window check that DUT performed TRIF initialization sequence.	N/A	
6.	From TRIS trace window verify that tr_init_req() APDU is well formed (no parse errors) and conforms to the following: <ul style="list-style-type: none"> - tr_init_req_tag is 0x0101 - trif_revision_code is 0x01 		REQ3754
7.	Verify that udcp_profile() part of tr_init_req() APDU is well formed (no parse errors).		REQ3754
8.	Verify that udcp_profile() contents correctly report DUT's features/interfaces as reported in vendor documentation (step #1). <ul style="list-style-type: none"> - number of tuners (_____) - supported video codecs (_____) - supported audio codecs (_____) upper frequency tuning range (_____) hardware_version_num (other than zero_____) str_byte (_____) 		REQ3754 REQ3878 REQ3879
9.	Verify that DUT did NOT send any challenge_req() APDUs.		REQ3832
10.	Check that TRIS sent challenge_req() APDU.	N/A	
11.	Verify that the DUT responds to a challenge_req() APDU with exactly one challenge_rsp() APDU within 5 seconds that provide values in the challenge_rsp() APDU for all datatype ids that are specified in the associated challenge_req() APDU.		REQ3763 REQ3764 REQ3766
12.	Verify that challenge_rsp() APDU does not include any datatypes other than datatype_id <u>7, 13, 15, and 17</u> . Note: These are Host_DevCert, Host_ManCert, DH_pubKey _H and SIGN _H .		REQ3765
13.	Verify that challenge_rsp() APDU(s) sent by DUT are well-formed (no parse errors) and conform to the following: <ul style="list-style-type: none"> - challenge_rsp_tag is 0x0106, - trif_revision_code is 0x01 		REQ3763 REQ3766

Step#	Procedure	Pass/Fail	REQ
14.	- Check in the TRIS log that the “Challenge response authenticated”.	N/A	
15.	Check that TRIS sends the HMAC key.	N/A	
16.	Verify that the DUT sent channel_table_req() APDU.		REQ3772
17.	Verify that channel_table_req()APDU is well formed (no parse errors) and conforms to the following: - channel_table_req_tag is 0x0107 - trif_revision_code is 0x01		REQ3772
18.	Check that TRIS replied with channel_table_rsp() APDU (actual status is not important at this time)	N/A	
19.	Verify that DUT sent udcp_status_update() APDU after receiving the channel table.		REQ3797
20.	Verify that udcp_status_update() APDU is well-formed (no parse errors) and conforms to the following: - udcp_status_update_tag is 0x0306 - trif_revision_code is 0x01		REQ3797
21.	Verify that udcp_status() part of udcp_status_update() APDU is well-formed (no parse errors). NOTE: semantics of udcp_status() are tested in section 3.1		REQ3797
22.	Exit back to the Main Menu on TRIS. Select Execute Script (e) This step configures TRIS to delay sending tr_init_rsp() APDU by 4.5 seconds. Execute the script: trInitRspDelay4500.tri.	N/A	
23.	Remove the USB cable from the DUT (UDCP) and plug it back into the UDCP to cause a USB reset and cause the UDCP to reinitialize.	N/A	
24.	Repeat steps #5 through #21 to confirm correct behavior of UDCP	N/A	
25.	Wait one minute and verify that the DUT never sent tr_status_req() APDU with a periodicity greater than one every five seconds.		REQ3791
26.	Exit the TRIS tool.	N/A	

3.1.2 Testcase: SCTE65 SI usage

Step#	Procedure	Pass/Fail	REQ
1.	Insert CableCARD into the DUT.	N/A	
2.	Power on the DUT and make it operational.	N/A	
3.	Verify that DUT can navigate channel map as received from CableCARD.		REQ3748
4.	Connect TRIS with the DUT.	N/A	
5.	Start TRIS application in TR mode.	N/A	
6.	Wait until DUT acquires channel map from TRIS.	N/A	
7.	Check that channel browsing is possible.	N/A	
8.	Disconnect USB cable from the DUT.	N/A	
9.	Verify that DUT reverted to channel map received from CableCARD.		REQ3749

3.2 Negative tests

3.2.1 Testcase: No tr_init_rsp() sent

Step#	Procedure	Pass/Fail	REQ
1.	Power-cycle the DUT and wait until it becomes operational (but do not connect USB cable to TRIS)	N/A	
2.	Start TRIS application in TR simulation mode (t).	N/A	
3.	Exit to Main Top menu, Select Execute Script (e) This step configures TRIS to suppress sending of any messages over TRIF. Execute the script trInitIgnore.tri. Note: This script will receive tr_init_req messages from the UDCP and just print out "Got TrInitReq Message" that they were received, but it will not respond.	N/A	
4.	Connect the DUT and TRIS as shown in figure 1.	N/A	
5.	From TRIS trace window verify that DUT attempted to initialize TRIF at least two times. Check that it performed USB reset and sent tr_init_req() APDU again. (Note the spec currently calls for a reset after each timeout of 5 seconds). The TRIS trace window will indicate when it receives a USB reset.		REQ3756
6.	Exit out of the TRIS application after this test is complete.	N/A	

3.2.2 Testcase: TR firmware upgrade reported in tr_init_rsp()

Step#	Procedure	Pass/Fail	REQ
1.	Power-cycle the DUT and wait until it becomes operational (but do not connect USB cable to TRIS)	N/A	
2.	Start the TRIS application in TR simulation mode.	N/A	
3.	Exit to main top menu in TRIS, execute the script: trStatusFWUpgrade.tri. Note: This step configures TRIS to report TR firmware upgrade tr_operational_status (0x02) (upgrade in progress) in tr_status_rsp() APDU.	N/A	
4.	Connect the DUT and TRIS as shown in figure 1. Note: The script allows 20 seconds to connect USB cable.	N/A	
5.	From TRIS trace window verify that DUT and TRIS goes through the initialization sequence, and TRIS reports firmware upgrade status in response to a tr_status_req() inquiry from the DUT. (Note it is not mandatory for the UDCP to send a tr_status_request during initialization, but recommended.)		REQ3789
6.	Verify that DUT did not attempt to proceed with initialization process after receiving tr_status_rsp() APDU.		REQ3821
7.	Verify that DUT can still navigate the channel map received from CableCARD.		REQ3821
8.	Verify that DUT never sent resolve_tuning_req() APDU.		REQ3821
9.	Exit out of the TRIS application after this test is complete.	N/A	

3.2.3 Testcase: TR not ready reported in tr_init_rsp()

Step#	Procedure	Pass/Fail	REQ
1.	Power-cycle the DUT (with CableCard included) and connected to the appropriate headend (i.e. SA or MOT depending upon CableCard used) and wait until it becomes operational (but do not connect USB cable to TRIS)	N/A	
2.	Start TRIS application.	N/A	
3.	From the top Main menu in TRIS, execute the script: trStatusNotReady.tri. This step configures TRIS to report TR not ready (tr_operational_status 0x01) (initializing) in tr_status_rsp() APDU.	N/A	
4.	Connect the DUT and TRIS as shown in figure 1.	N/A	
5.	Verify that DUT can still navigate the channel map received from CableCARD. From TRIS trace window verify that DUT attempted to initialize TRIF. Verify that DUT did not attempt to proceed with initialization process after receiving tr_status_rsp() APDU.		REQ3821
6.	Verify that DUT can still navigate the channel map received from CableCARD.		REQ3821
7.	Close down the TRIS application when the test is complete.	N/A	

3.2.4 Testcase: Authentication Failure Test

Step#	Procedure	Pass/Fail	REQ
1.	Power-cycle the DUT (with CableCard inserted) and wait until it becomes operational (but do not connect USB cable to TRIS)	N/A	
2.	Start TRIS application in TR simulation mode.	N/A	
3.	From the top Main menu in TRIS, execute the script: trInitRspBadAuth.tri. This step configures TRIS to report authentication failed in tr_status_update() APDU and to withhold sending the HMAC key.	N/A	
4.	Connect the DUT and TRIS as shown in figure 1.	N/A	
5.	From TRIS trace window verify that DUT attempted to initialize TRIF. Verify that DUT does not send a resolve_tuning_req() APDU after failing authentication.		REQ3869
6.	Verify that UDCP can still navigate the channel map received from CableCARD. Note: the UDCP may try to re-init – the exact behavior from the UDCP upon failing authentication is not defined. The TR may also send another challenge_req() if it fails – but again this is undefined.		REQ3821
7.	Close down the TRIS application when the test is complete.	N/A	

4 TEST: CHANNEL TABLE DELIVERY

4.1.1 Testcase: Channel table delivery and update

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with channel map for the Motorola plant. Do this by executing the script trChanMot.tri for the Motorola plant. (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational. The DUT should have an RF connection to the Motorola Cert Plant.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	From TRIS trace window check that DUT performed TRIF initialization sequence. Check for tr_init_req(), tr_init_rsp(), challenge_req(), challenge_rsp()	N/A	
6.	Verify that UDCP sent channel_table_req() APDU.		REQ3772
7.	From TRIS trace window verify that channel_table_req() APDU is well-formed (no parse errors) and conforms to the following: <ul style="list-style-type: none"> - channel_table_req_tag is 0x0107 - trif_revision_code is 0x01 		REQ3772
8.	Check that TRIS sent channel_table_rsp() APDU(s) with the channel map as specified in the script.	N/A	
9.	Verify that DUT sent resolve_tuning_req() APDU (if the DUT doesn't attempt to tune to a channel upon startup, may need to select a channel on the DUT using either the remote control or the front panel control – select a channel that is specified in the channel map loaded in the script (9, 25, 125, or 150).		REQ3776
10.	Verify that resolve_tuning_req() APDU is well-formed (no parse errors) and conforms to the following: <ul style="list-style-type: none"> - resolve_tuning_req_tag is 0x0201 - trif_revision_code is 0x01 - channel_source_type is 0 - channel_number is the same as presented by DUT on the screen - resolve_tuning_digest is valid 		REQ3771 REQ3776
11.	Check that TRIS sent resolve_tuning_rsp() APDU.	N/A	
12.	Check that DUT used the information from the resolve_tuning_rsp() APDU and tuned to the service as instructed. Specifically, check: <ul style="list-style-type: none"> - tune_frequency - tune_program_number (program_number to look up in PAT in transport defined by tune_frequency) - video and audio are present in the tuned service 	N/A	
13.	Verify that DUT sent resolve_tuning_cnf() APDU that is well-formed (no parse errors) and conforms to the following: <ul style="list-style-type: none"> - resolve_tuning_cnf_tag is 0x0204 - trif_revision_code is 0x01 		REQ3788
14.	Verify that udcp_status() part of resolve_tuning_cnf() APDU is well-formed (no parse errors) and conforms to the following: <ul style="list-style-type: none"> - udcp_status_revision is 0x01 - user_activity_detected is 1 - udcp_status is 0x01 		REQ3788

Step#	Procedure	Pass/Fail	REQ
15.	Verify that one of the tuner_channel_number fields of the udcp_status() matches currently tuned channel.		REQ3788
16.	Repeat steps #9 through #16 for two additional channels that are in the channel map loaded in the script.		
17.	Tune to channel 114. Verify that the guide and the program content show "HBO 2 w" as the channel name.	N/A	
18.	Tune to channel 115. Verify that the guide and the program content show "HBO Sig" as the channel name.		
19.	In TRIS, load a modified Motorola channel map. At the main menu in TRIS, execute the script "trChanMotTest.tri".	N/A	
20.	Verify that UDCP displays new channel name (TEST114) on channel 114 within five minutes. This can be verified by checking the guide data in the UDCP, tuning to channel 114 and verifying the channel name has changed. (Note that the tuning information is not changed – this is just a test to verify the name change in the table.		REQ3753
21.	Tune to channel 115.	N/A	
22.	Verify that UDCP displays new program content and new channel name on channel 115.		REQ3753
23.	Verify that the guide data shows channel 300. Tune to channel 300. Verify the TRIS receives a resolve_tuning_request, TRIS sends the response, and the UDCP successfully tunes to channel 300.		REQ3753 REQ3776

4.1.2 Testcase: Channel Table Update

1.	Start TRIS application.	N/A	
2.	Configure TRIS with channel map for the Motorola plant. Do this by executing the script trChanMot.tri for the Motorola plant. (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational. The DUT should have an RF connection to the Motorola Cert Plant.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	From TRIS trace window check that DUT performed TRIF initialization sequence. Check for tr_init_req(), tr_init_rsp(), challenge_req(), challenge_rsp()	N/A	
6.	Verify that UDCP sent channel_table_req() APDU.		REQ3772
7.	From TRIS trace window verify that channel_table_req() APDU is well-formed (no parse errors) and conforms to the following: - channel_table_req_tag is 0x0107 - trif_revision_code is 0x01		REQ3772
8.	Check that TRIS sent channel_table_rsp() APDU(s) with the channel map as specified in the script.	N/A	
9.	Check the guide data in the UDCP to verify it received and is using the channel table from the trChanMot.tri script. Note this should have valid names for channels 114 and 115, and it should not include channel 300.	N/A	
10.	Tune to channel 114. Verify that the guide and the program content show "HBO 2 w" as the channel name.	N/A	
11.	Tune to channel 115. Verify that the guide and the program content show "HBO Sig" as the channel name.	N/A	

12.	In TRIS, load a modified Motorola channel map. At the main menu in TRIS, execute the script "trChanMotTest.tri".	N/A	
13.	Verify that UDCP displays new program content and new channel name (TEST114) on channel 114. This can be verified by checking the guide data in the UDCP, tuning to channel 114 and verifying the channel name has changed. (Note that the tuning information is not changed – this is just a test to verify the name change in the table.		REQ3753
14.	Tune to channel 115.	N/A	
15.	Verify that UDCP displays new program content and new channel name on channel 115.		REQ3753
16.	Verify that the guide data shows channel 300. Tune to channel 300. Verify the TRIS receives a resolve_tuning_request, TRIS sends the response, and the UDCP successfully tunes to channel 300.		REQ3753 REQ3776

4.1.3 Testcase: Large Channel table delivery

The purpose of this test is to verify that the DUT can handle a channel table sent in more than one APDU.

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with the large channel table. Do this by executing the script trChanBig.tri. This will send the channel table to the DUT over a number of APDUs. (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational. The DUT should have an RF connection to the Motorola Cert Plant.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	From TRIS trace window check that DUT performed TRIF initialization sequence. Check for tr_init_req(), tr_init_rsp(), challenge_req(), challenge_rsp()	N/A	
6.	Verify that UDCP sent channel_table_req() APDU.		REQ3772
7.	From TRIS trace window verify that channel_table_req() APDU is well-formed (no parse errors) and conforms to the following: - channel_table_req_tag is 0x0107 - trif_revision_code is 0x01		REQ3772
8.	Check that TRIS sent channel_table_rsp() APDU(s) with the channel map as specified in the script.	N/A	
9.	Verify that the DUT received and processed the channel table and is using it within 5 minutes. The mechanism to verify this can differ from UDCP to UDCP. Check the guide data to see if it reflects the new table. Tune the UDCP to a few valid channels that are in the channel map: i.e. 125, 126, 150, and 300. The UDCP should successfully tune and view video on all these channels on the Motorola plant.		REQ3753

5 TEST: TUNING TESTS

5.1.1 Testcase: Tuner use status test

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with Motorola channel map by executing the script: "trChanMot.tri" (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	Wait until DUT and TRIS complete initialization. Depending upon what the last channel was in the UDCP, it may resolve to a valid channel or it may not.	N/A	
6.	Tune to a digital channel in the chanMot channel table (114, 115, 125, or 150). Note the channel number (_____) and its source id (_____).	N/A	
7.	Verify that DUT sent resolve_tuning_req() APDU with a tuner_use_status field equal to 0x00 Live full screen video (no HDD recording). Note the value of Itsid field: (_____).		REQ3776
8.	Verify that resolve_tuning_req() APDU contains a channel number or source_id as noted in step #6.		REQ3776
9.	Check that DUT tuned to channel requested in step #6.	N/A	
10.	From TRIS menu enter TR mode (t). Select (s) to enter "Send a message" menu. Select (s) to send udcp_status_req() APDU.	N/A	
11.	Verify that DUT sent udcp_status_rsp() APDU. Verify that the tuner_use_status value of the active tuner is reported at 0x00 (Live full screen video no HDD recording), or 0x02 (Live PIP or POP video) Verify that tuner_use_status values of all other tuners (Itsids) than the one noted in step #7 are equal to 0x05 (inactive) or 0x06 (speculative recording).		REQ3866
12.	If the DUT has more than one tuner, take the action on the UDCP to switch to another tuner (i.e. press the Live TV button or whatever is suitable for the DUT).	N/A	
13.	Tune to a digital channel that was not used in step 6 above (114, 115, 125, or 150). Note the channel number (_____).		
14.	Verify that DUT sent resolve_tuning_req() APDU with a tuner_use_status field equal to 0x00 Live full screen video (no HDD recording). Note the value of Itsid field: (_____). Verify that the Itsid is different from step 7 above.		REQ3784 REQ3817
15.	Check that the DUT tuned to the channel requested in step 13. Verify that the DUT sent a resolve_tuning_rsp() APDU with the current tuner tuner_use_status of 0x00, and the tuner in step 7 (the Itsid noted) reports a tuner_use_status of 0x06 (speculative recording).		REQ3783 REQ3828
16.	If the DUT has time-shift buffer capability, rewind the current viewed video back a few minutes and viewed the buffered video. Verify that the DUT sent a udcp_status_update() APDU with the tuner in focus reporting 0x00		REQ3784 REQ3829

Step#	Procedure	Pass/Fail	REQ
	(Live full screen video no HDD recording) or 0x02 (Live PIP or POP video no HDD recording). Note that the DUT should send a status update due to the remote control interaction, but if it does not, initiate a status request from the TRIS tool (in TR mode (t), TRMSG mode (s), enter (s) to request status).		
17.	If the DUT has a DVR functionality, perform steps #18 through #27.	N/A	
18.	Start the recording of current channel.	N/A	
19.	Verify that DUT sent a udcp_status_update() APDU with a tuner_use_status field equal to 0x01 - Live full screen video (HDD recording) for the tuner and channel that is being recorded. Note the ltsid value of the tuner in use: (_____) <p>Verify that the other tuners (if multi-tuner) is not recording and is reporting a tuner_use_status of 0x05 (inactive) or 0x06 (speculative recording).</p>		REQ3781 REQ3826
20.	If the DUT has more than one tuner, tune to a digital channel (do not stop the recording) – this should put the record tuner in the background. Note the channel number (_____).	N/A	
21.	Verify that the DUT sent resolve_tuning_req() APDU with a tuner_use_status field equal to 0x00 Live full screen video (no HDD recording)		REQ3784
22.	Verify that resolve_tuning_req() APDU contains a channel number or source_id as noted in step #13.		REQ3776
23.	Verify that the DUT sent another resolve_tuning_req() APDU with a tuner_use_status field equal to 0x04 (Recording only) OR updated the tuner_use_status in the resolve_tuning_cnf() APDU to 0x04 (Recording Only)		REQ3776
24.	Wait for the recording to finish without ever changing the “focus” back to the program being recorded. (Note this might not work if the DUT doesn’t have accurate guide data – you may need to stop the recording manually, in which case you need to put the focus back to the tuner being recorded.)	N/A	
25.	Verify that the DUT did not send resolve_tuning_req() APDU with a tuner_use_status field equal to 0x05 (Inactive). (Note: this value is only applicable to udcp_status_rsp() and udcp_status_update() APDUs.		REQ3778
26.	Verify that the DUT sent udcp_status_update() APDU.		REQ3863
27.	Verify that APDU received in previous step is well formed (no parse errors) and tuner_use_status field of tuner (ltsid) noted in step #11 is 0x05 (inactive) or 0x06 (speculative recording).		REQ3863
28.	INDEPENDENT OF MULTIPLE TUNERS OR DVR FUNCTIONALITY, verify that request_id fields have incremented for every transmission of resolve_tuning_req() APDU EVEN across multiple tuners		REQ3779

5.1.2 Testcase: Invalid HMAC Key Test

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with Motorola channel map by executing the script: “trChanMot.tri” (Note this also puts the TRIS in TR simulation mode.)	N/A	

Step#	Procedure	Pass/Fail	REQ
3.	Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	Wait until DUT and TRIS complete initialization. Depending upon what the last channel was in the UDCP, it may resolve to a valid channel or it may not.	N/A	
6.	Tune tuner 0 to channel 114.	N/A	
7.	Check that DUT sent a valid resolve_tuning_req() APDU and the DUT tunes to channel 114.	N/A	
8.	If the DUT is a multi-tuner device, tune the other tuners to channels other than channel 125.	N/A	
9.	In TRIS, execute the script "trtuneRsp06.tri". This will respond to a tune request indicating the HMAC key was invalid and send a new HMAC key.	N/A	
10.	Within 20 seconds of executing the script, tune the DUT tuner 0 to channel 125.	N/A	
11.	Check in the TRIS log window that the resolve_tuning_req() came in for ltsid 0 and channel 125.	N/A	
12.	Verify in the TRIS log window that TRIS responded with a resolve_tuning_rsp() with a tune_status of 0x06=Invalid resolve_tuning_digest, and sends a new HMAC key.		REQ3861
13.	Tune the DUT to another channel (that is not already acquired by other tuners in the DUT).	N/A	
14.	Verify that the resolve_tuning_req() comes in and is validated by TRIS. This validates that the DUT is using the new HMAC key.		REQ3771
15.	In TRIS, execute the script "trtuneRsp01.tri". This will respond to a tune request indicating that Tuning resolution cannot be determined at this time (Channel not available).	N/A	
16.	Within 20 seconds of executing the script, tune the DUT tuner 0 to channel 125.	N/A	
17.	Check in the TRIS log window that the resolve_tuning_req() came in for ltsid 0 and channel 125.	N/A	
18.	Check in the TRIS log window that the TRIS responded with a resolve_tuning_rsp() with a tune_status of 0x01= Tuning resolution cannot be determined at this time (Channel not available) Verify the DUT creates a visible message to the user indicating the channel is temporarily unavailable.		REQ3882 REQ3883
19.	Send an head-end system EAS message to the DUT while running script "trtuneRsp01.tri" again. Verify the DUT ends the visible tuning error message and processes the EAS message .		REQ3883
20.	Run script "trtuneRsp01.tri" again. When the DUT starts to display the visible message power cycle the DUT. Verify that the visible message ends.		REQ3882
21.	Tune the DUT to another channel (that is not already acquired by other tuners in the DUT).	N/A	
22.	In TRIS, execute the script "trtuneRsp02.tri". This will respond to a tune request indicating that Tuning resolution cannot be determined at this time (Network not available)	N/A	

Step#	Procedure	Pass/Fail	REQ
23.	Within 20 seconds of executing the script, tune the DUT tuner 0 to channel 125.	N/A	
24.	Check in the TRIS log window that the resolve_tuning_req() came in for ltsid 0 and channel 125.	N/A	
25.	Check in the TRIS log window that the TRIS responded with a resolve_tuning_rsp() with a tune_status of 0x02= Tuning resolution cannot be determined at this time (Network not available). Verify the DUT creates a visible message to the user indicating the channel is temporarily unavailable.		REQ3884 REQ3885
26.	Send an head-end system EAS message to the DUT while running script "trtuneRsp02.tri" again. Verify the DUT ends the visible tuning error message and processes the EAS message.		REQ3885
27.	Run script "trtuneRsp02.tri" again. When the DUT starts to display the visible message power cycle the DUT. Verify that the visible message ends.		REQ3884
28.	Close down the TRIS application when the test is complete.	N/A	

5.1.3 Testcase: Resolve Tuning Update Test

This test validates the UDCP responds appropriately to a resolve tuning update.

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application	N/A	
2.	Configure TRIS with Motorola channel map by executing the script: "trChanMot.tri" (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	Wait until DUT and TRIS complete initialization. Depending upon what the last channel was in the UDCP, it may resolve to a valid channel or it may not.	N/A	
6.	Tune tuner 0 to channel 125	N/A	
7.	Check that the DUT sent a valid resolve_tuning_req() APDU and the DUT tunes to channel 125.	N/A	
8.	If the DUT is a multi-tuner device, tune the other tuners to channels other than channel 125.	N/A	
9.	In TRIS, execute the script "trTuneUp125NewParms.tri". This will send the DUT a resolve_tuning_update() command indicating that channel 125 is at a new frequency, program number, source id, etc. It is sending parameters for another valid channel on the Motorola Cert plant.	N/A	
10.	Check that the UDCP changes to a new program (should occur in approximately 10 seconds from initiating the script). Verify that UDCP presented a visible message indicating the requested channel is temporarily unavailable and an alternative is available.		REQ3894
11.	In approximately 10 more seconds, check that the DUT changes back to the original programming information on channel 125.	N/A	

Step#	Procedure	Pass/Fail	REQ
12.	In TRIS, execute the script "trTuneUp125NewParams126.tri". This will send the DUT a resolve_tuning_update() command indicating that channel 125 is no longer available and an alternative (channel 126) is available.	N/A	
13.	Verify that the DUT creates a visible message to the user indicating the requested channel is temporarily unavailable and an alternative is available.		REQ3892 REQ3893
14.	Tune the DUT to another channel (that is not already acquired by other tuners in the DUT).	N/A	
15.	Tune to channel 125.	N/A	
16.	In TRIS, execute the script "trTuneUp125NewParams126.tri". This will send the DUT a resolve_tuning_update() command indicating that channel 125 is no longer available and an alternative (channel 126) is available. <i>The previous test was validating the behavior after a tune update, this test is validating after the UDCP received a tune response.</i>	N/A	
17.	Verify that the DUT creates a visible message to the user indicating the requested channel is temporarily unavailable and an alternative is available.		REQ3890 REQ3891
18.	Tune away and back to channel 125.		
19.	In TRIS, execute the script "trTuneUpdate125_1.tri". This will send the DUT a resolve_tuning_update() command indicating that channel 125 tuning resolution cannot be determined at this time (Channel not available).	N/A	
20.	Check in the TRIS log window that the TRIS sent a resolve_tuning_update() with a tune_status of 0x01= Tuning resolution cannot be determined at this time (Channel not available).	N/A	
21.	Verify the DUT creates a visible message to the user indicating that the Channel is temporarily unavailable.		REQ3886 REQ3887
22.	Tune away and tune back to 125 on ltsid 0.		
23.	Send a broadcast head-end system EAS message. Execute the trTuneUpdate_1.tri script again. Verify that the DUT does NOT impact the EAS message and does NOT display a message about the channel not available.		REQ3886
24.	Tune away to a different channel.	N/A	
25.	Tune back to channel 125.	N/A	
26.	Put the DUT in standby mode. In TRIS, execute the script "trTuneUpdate125_1.tri". This will send the DUT a resolve_tuning_update() command indicating that channel 125 tuning resolution cannot be determined at this time (Channel not available).	N/A	
27.	Check in the TRIS log window that theTRIS sent a resolve_tuning_update() with a tune_status of 0x01= Tuning resolution cannot be determined at this time (Channel not available).	N/A	
28.	Verify the DUT does not create a visible message to the user indicating that theChannel is temporarily unavailable.		REQ3886 REQ3887
29.	Tune away and tune back to channel 125 (on ltsid 0).	N/A	
30.	In TRIS, execute the script "trTuneUpdate125_2.tri".	N/A	

Step#	Procedure	Pass/Fail	REQ
	This will send the DUT a resolve_tuning_update() command indicating that channel 125 tuning resolution cannot be determined at this time (Network not available).		
31.	Check in the TRIS log window that the TRIS sent a resolve_tuning_update() with a tune_status of 0x02= Tuning resolution cannot be determined at this time (Network not available).	N/A	
32.	Verify the DUT creates a visible message to the user indicating that the Channel is temporarily unavailable.		REQ3888 REQ3889
33.	While the message from the previous step is still displayed, send a broadcast head-end system EAS message. Verify the DUT ends its presentation of messages from the previous step during an EAS message.		REQ3889
34.	Clear the EAS message. Tune away and tune back to channel 125 on ltsid 0.	N/A	
35.	Put DUT in Standby mode. Execute the "trTuneUpdate125_2.tri" script again. Verify that the DUT does NOT show any message about network not available (in standby mode it likely will not have any video output).		REQ3888

6 TEST: USER ACTIVITY TESTS

6.1.1 Testcase: Inactivity dialog test

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with Motorola channel map by executing the script: "trChanMot.tri" (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in figure 1.	N/A	
5.	Wait until DUT and TRIS complete initialization and a channel is presented. You may need to tune the DUT to a valid channel in the Motorola channel map (9,25,114,115,125, or 150). Select tuner 0 (ltsid 0) as the active tuner.	N/A	
6.	Wait at least 5 seconds and do not interact with the DUT (front panel, remote, TRIS tool).		
7.	In TRIS, send a tr_message for ltsid 0 by executing the script "trTrMessage00.tri"	N/A	
8.	Verify that DUT sent tr_message_rsp() APDU with message_response_code field equal to 0x00 (Message displayed) within 5 seconds of receiving tr_message_req().		REQ3806
9.	Verify that DUT presented a message on the screen related to a user inactivity message (i.e. Are you still watching?).		REQ3900 REQ3901
10.	Acknowledge presented message with a 'OK' or 'Select' RCU key and check that the message is removed from the display.	N/A	
11.	Put DUT in Standby mode.	N/A	
12.	Send the tr_message() APDU with ltsid field as noted in step #5 and message_type field equal to 0x00. In TRIS, execute the trTrMessage00.tri script.	N/A	
13.	Verify that UDCP sent tr_message_rsp() APDU with message_response_code field equal to 0x01 (Message not displayed, UDCP in standby) within 5 seconds of receiving tr_message req().		REQ3806
14.	Bring DUT out of Standby.	N/A	
15.	Verify that UDCP did NOT present any inactivity dialog.		REQ3900
16.	If the DUT is multi-tuner device send tr_message() APDU with ltsid field that corresponds to the other tuner. Execute the script "trTrMessage01.tri"	N/A	
17.	Verify that UDCP sent tr_message_rsp() APDU with message_response_code field equal to 0x02 (Message not displayed, no program present not in video display mode) within 5 seconds of receiving tr_message req().		REQ3806

6.1.2 Testcase: Unspecified Message Test

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with Motorola channel map by executing the script:	N/A	

Step#	Procedure	Pass/Fail	REQ
	"trChanMot.tri" (Note this also puts the TRIS in TR simulation mode.)		
3.	Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	Wait until DUT and TRIS complete initialization and a channel is presented. You may need to tune the DUT to a valid channel in the Motorola channel map (9,25,114,115,125, or 150). Select tuner 0 (ltsid 0) as the active tuner.	N/A	
6.	In TRIS, send a tr_message of type 1 (Unspecified Message) for ltsid 0 by executing the script "trTrMessage10.tri"	N/A	
7.	Verify that the DUT sent tr_message_rsp() APDU with message_response_code field equal to 0x00 (Message displayed) within 5 seconds of receiving tr_message req().		REQ3806 REQ3898 REQ3899
8.	Verify that the DUT presented the EXACT message on the screen that is in the script.		REQ3898 REQ3899
9.	Acknowledge presented message with a 'OK' or 'Select' RCU key and check that the message is removed from the display.	N/A	
10.	Put DUT in Standby mode. Verify the DUT sends a udcp_status_update() APDU with the udcp_status = 0x00 (UDCP in standby).	N/A	
11.	Send the tr_message() APDU with ltsid field as noted in step #5 and message_type field equal to 0x00. In TRIS, execute the trTrMessage10.tri script.	N/A	
12.	Verify that UDCP sent tr_message_rsp() APDU with message_response_code field equal to 0x01 (Message not displayed, UDCP in standby) within 5 seconds of receiving tr_message req().		REQ3806 REQ3898
13.	Bring DUT out of Standby.	N/A	
14.	Verify that UDCP did NOT present any inactivity dialog.		REQ3898
15.	If the DUT is multi-tuner device send tr_message() APDU with ltsid field that corresponds to the other tuner. Execute the script "trTrMessage11.tri"	N/A	
16.	Verify that UDCP sent tr_message_rsp() APDU with message_response_code field equal to 0x03 (Message not displayed, no program present) within 5 seconds of receiving tr_message req().		REQ3806
17.	Change tuners so that the second tuner is viewed. Note: to do that, start the recording of your current channel (which utilizes tuner 0) and perform a channel change.	N/A	
18.	Execute the script "trTrMessage11.tri".	N/A	
19.	Verify that the DUT sent tr_message_rsp() APDU with message_response_code field equal to 0x00 (Message displayed) within 5 seconds of receiving tr_message req().		REQ3806 REQ3898 REQ3899
20.	Verify that the UDCP presented the EXACT message on screen that is in the script.		REQ3898 REQ3899
21.	Acknowledge the message with an OK or Select RCU key and check that the message is removed from the display.	N/A	


6.1.3 Testcase: User activity test

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application.	N/A	
2.	Configure TRIS with Motorola channel map by executing the script: "trChanMot.tri" (Note this also puts the TRIS in TR simulation mode.)	N/A	
3.	Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in figure 1.	N/A	
5.	Wait until DUT and TRIS complete initialization and a channel is presented.	N/A	
6.	Change a channel.	N/A	
7.	Verify that UDCP sent <code>udcp_status_update()</code> APDU (within one second after RCU keypress) with <code>user_activity_detected</code> field is equal to 1.		REQ3798 REQ3823 REQ3824
8.	Change a channel.	N/A	
9.	Verify that as the UDCP receives updates of the <code>ca_allowed_status</code> from the CableCARD, it can send <code>udcp_status_update()</code> APDUs to the TR to reflect this updated status. Note: When the UDCP tunes, it may not immediately have <code>ca_allowed_status</code> from the CableCARD. In cases where the UDCP does not have <code>ca_allowed_status</code> , it may report <code>ca_allowed_status =0b1</code> .		REQ3825
10.	Send <code>udcp_status_req()</code> APDU. Either execute the script "trUdcpStatusReq.tri", or put the TRIS in tr mode, send a message mode, and send the status request from the menu.	N/A	
11.	Verify that UDCP replied with <code>udcp_status_rsp()</code> APDU with <code>user_activity_detected</code> field is equal to 0.		REQ3795
12.	Perform any kind of RCU activity.	N/A	
13.	Send <code>udcp_status_req()</code> APDU.	N/A	
14.	Verify that UDCP replied with <code>udcp_status_rsp()</code> APDU with <code>user_activity_detected</code> field is equal to 1.		REQ3795

7 TEST: EMERGENCY ALERT SYSTEM

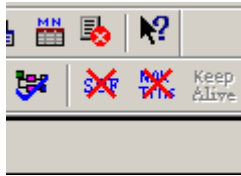

This test additionally requires an HPNx Pro test tool.

Step#	Procedure	Pass/Fail	REQ
1.	Bring up the HPNx Pro software on the given PC. Verify that the PC and HPNx Pro are on the same isolated network.	N/A	
2.	Note the last 4 (mac address) digits of HPNx Pro you are using. (Information is on the back side of card)	N/A	
3.	Under the Device tab, enter the 4 digits in the blank space labeled "Your HPNx Pro ID".	N/A	
4.	Insert the HPNx Pro extender card into DUT. From the HPNx Pro trace window, check that the status of the HPNx Pro goes ready and the resource manager opens its session.	N/A	
5.	Right click on the Trace window to select SPDU and Payload for full vision of all layers.	N/A	
6.	Select "Extended channel" tab, depending on the DUT you may have to change the resource version of the HPNX PRO to match that of the DUT. To change version, right click on the extended channel slot session. Select "Change resource version". In the explorer user prompt window enter the correct resource version needed to match the DUT. Click OK. On the "Extended channel" tab press the Play button to open the extended channel resource.	N/A	
7.	On the "Extended channel" tab press the Play button to open the extended channel resource.	N/A	
8.	Verify that the DUT issues a New_flow_req to Extended Channel requesting a service_type = MPEG_section with a PID = 0x1FFC. See example below. <pre> → A [90:49:27.834] New_flow_req to Extended Channel (note: description version and opened version differs) [9F 8E 00 03 00 FF FC] service_type = MPEG_section [0x00] MPEG_section_parameters PID = 0x1FFC ← A [90:49:27.904] New_flow_cnf from Extended Channel [9F 8E 01 06 00 05 00 00 01 00 E0] status_field = request granted [0x00] flows_remaining = 5 flow_info FLOW_ID = 0x000001 service_type = MPEG_section [0x00] </pre> Trace shows the HPNx Pro sending new_flow_cnf () APDU in response to DUT's request.	N/A	
9.	Under the "Extended channel" tab, expand the Flow Feed button.	N/A	

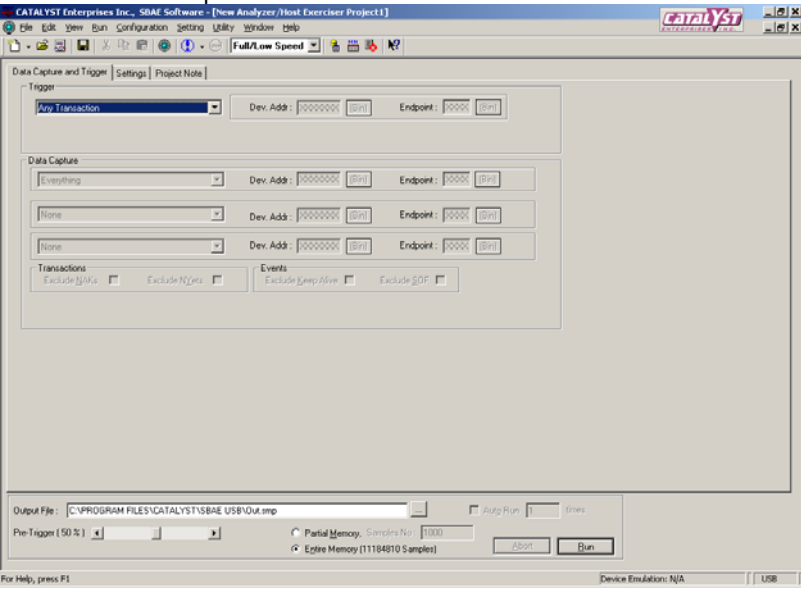

Step#	Procedure	Pass/Fail	REQ
10.	<p>Click “Current Time STT”, then click “Send” for sending current system time.</p> 	N/A	
11.	<p>Under same “Flow Feed, next to the “SI table file” click the browse button. Select the “Profile1.hex” file from wherever you have stored it on the HPNx Pro. Click “OPEN”.</p>	N/A	
12.	<p>On the DUT, using the channel up and down buttons or the remote control, check that the channel map has been successfully loaded and that the DUT can navigate the channel map.</p>	N/A	
15.	<p>Start TRIS application in TR simulation mode.</p>	N/A	
16.	<p>Configure TRIS with default channel map.</p>	N/A	
17.	<p>Connect DUT and TRIS as shown in figure 1.</p>	N/A	
18.	<p>Wait until DUT and TRIS complete initialization and a channel is presented.</p>	N/A	
19.	<p>Using HPNx Pro, under same “Flow Feed” next to the “SI table file” click the browse button. Select the “Maximum_priority_V2.hex” file from wherever you have stored it on the HPNx Pro. Click “OPEN”. Click send.</p>	N/A	
20.	<p>From TRIS trace verify that DUT sent resolve_tuning_req() APDU for the channel number (or source id) indicated in the EAS message.</p>		REQ3777
21.	<p>Check that TRIS sent resolve_tuning_rsp() APDU.</p>	N/A	
22.	<p>Verify that DUT force tuned to details channel as indicated by OOB EAS message.</p>		REQ3777
23.	<p>Optional: Using HPNx Pro, under same “Flow Feed” next to the “SI table file” click the browse button. Select the “Medium_priority_V2.hex” file from wherever you have stored it on the HPNx Pro. Click “OPEN”. Click send.</p>	N/A	
24.	<p>Verify that the DUT starts a scrolling message “EAS Blizzard Warning test message if you get this look outside and see if it is snowing”.</p>		

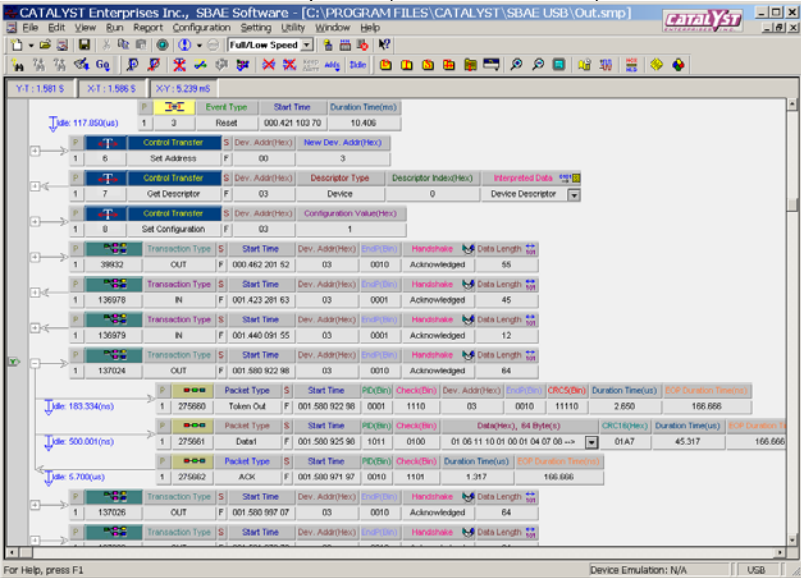
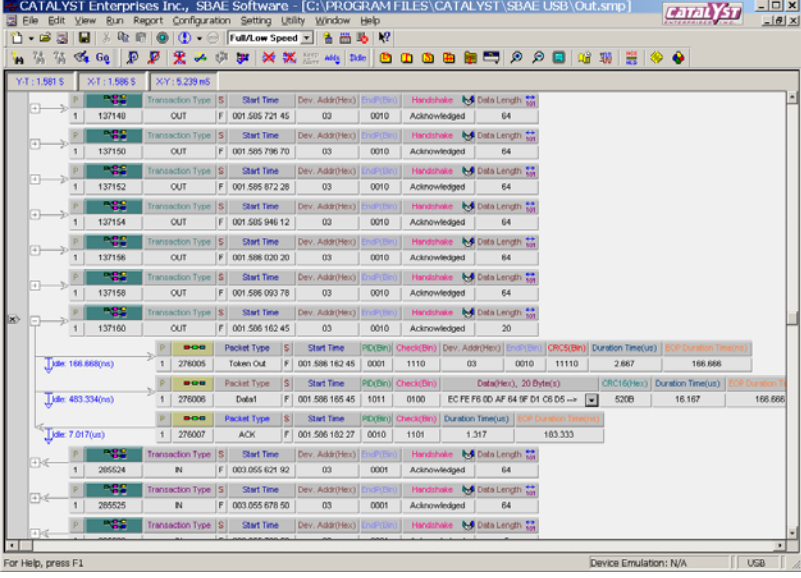
8 TEST: USB INTERFACE TESTS

8.1.1 Testcase: Device Descriptor Test

Step#	Procedure	Pass/Fail	REQ
1.	Connect UDCP with TRIS through Catalyst USB analyzer (use front panel Host and Device USB receptacles)	N/A	
2.	Start TRIS software but do not select operational mode yet.	N/A	
3.	Connect USB analyzer to the PC. Use back panel USB 'Host Interface' receptacle.	N/A	
4.	Start USB analyzer software. Select 'USB' in device selection dialog. Click 'OK'. Software should present list of connected analyzers. Select the only available analyzer and confirm.	N/A	
5.	From 'File' menu select 'New' -> 'Analyzer/Host Exerciser Project'. Project window should pop up. Make sure that <ul style="list-style-type: none"> - 'Trigger' is set to 'Any Transaction', - 'Data Capture' is set to 'Everything' - Transaction/Event filtering is not active. 	N/A	
6.	Set number of samples to at least 1'000'000 (one million).	N/A	
7.	Disconnect UDCP from USB analyzer.	N/A	
8.	Click 'Run'	N/A	
9.	Connect UDCP to 'Device' receptacle of the USB analyzer.	N/A	
10.	Once you do that, analyzer software should report a trigger (Balloon window in bottom-right corner of the screen) and start the capture.	N/A	
11.	Switch TRIS to TR simulation mode. Make sure it does not return an error afterwards.	N/A	
12.	Wait for the analyzer to finish the capture. Once capture is complete analyzer will display a trace window.	N/A	
13.	From analyzer software toolbar filter out all SOF and NAK frames. 	N/A	
14.	From analyzer software trace window verify that there were no protocol errors reported.		REQ3723
15.	Find the second received Device Descriptor in the trace window. Click Device Descriptor Menu tab to expand and check that values 0x07b2 (idVendor) and 0x6002 (idProduct) are included. Scroll down to the first Control Transfer "Set Configuration". Verify the DUT sent Configuration value 0x01 and then Transaction Type "SetUp" handshake acknowledgement showing no protocol errors. 		REQ3741 REQ3745

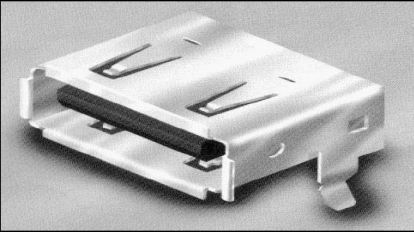
8.1.2 Testcase: Data Transfer Rate

Step#	Procedure	Pass/Fail	REQ
1.	Power-cycle the DUT and wait until it becomes operational. The DUT should have an RF connection to the Motorola Cert Plant.	N/A	
2.	Connect DUT and TRIS as shown in Figure 1.	N/A	
3.	Disconnect the USB cable running into the DUT		
4.	Start TRIS application.	N/A	
5.	Put the TRIS into Tuning Resolver mode by selecting (t)	N/A	
6.	Start the USB analyzer software.	N/A	
7.	From 'File' menu select 'New' -> 'Analyzer/Host Exerciser Project'. Project window should pop up. Make sure that <ul style="list-style-type: none"> - 'Trigger' is set to 'Any Transaction', - 'Data Capture' is set to 'Everything' - Transaction/Event filtering is not active. 	N/A	
8.	Set number of samples to the maximum. 	N/A	
9.	Note: perform the next three steps within a few seconds. Click 'Run' on the USB analyzer software window.	N/A	
10.	Connect the USB cable to the DUT.	N/A	
11.	Check that UDCP sent channel_table_req() APDU and the TRIS responded with the large channel table.	N/A	
12.	Observe the TRIS Log output. Once the DUT has sent the ChallengeDataRspMsg press Stop on the USB analyzer. Once the capture is complete the analyzer will display a trace window.	N/A	
13.	From the analyzer software toolbar filter out all SOF and NAK frames. 	N/A	

Step#	Procedure	Pass/Fail	REQ
14.	<p>Find the start of the ChallenDataRspMsg by looking for the pattern 01 06 in the first two bytes of the data portion of the captured packets. This should be the first packet with a length of 64 bytes.</p> <p>Record the start time for this packet (_____).</p> 	N/A	
15.	<p>Scroll down to the bottom of the packet capture screen and find the end of the ChallengeDataRspMsg. The last packet of the ChallengeDataRspMsg will be the first packet encountered with a length less than 64 bytes after a long string of 64 byte packets.</p> <p>Record the start time for the last ChallengeDataRspMsg (_____)</p> 	N/A	
16.	<p>Subtract the start time for the first packet from the start time for the last packet and record the result (_____). This is the transmission time (T).</p>	N/A	
17.	<p>Calculate the transmission speed by dividing the number of bits transmitted by the transmission time. The ChallengeDataRspMsg is 4370 bytes long. The transmission speed would be = $(4370 * 8) / T$ in bits per second.</p>	N/A	
18.	<p>Verify the the transmission speed is greater than or equal to 100 kb/s.</p>		REQ3733

9 TEST: MECHANICAL

9.1.1 Testcase: Visual Inspection of Keyed Connector

Step#	Procedure	Pass/Fail	REQ
1.	<p>Visually inspect the USB connection being used on the DUT.</p> <p>Verify the DUT USB interface type A receptacle [USB2.0]</p>  <p><i>(Downstream Output from the USB Host)</i></p>		<p>REQ3714 REQ3715</p>

10 TEST: DIAGNOSTIC MESSAGES

10.1 Testcase: Positive test – Diagnostic request

Step#	Procedure	Pass/Fail	REQ
1.	Start TRIS application. Note the next step puts the TRIS in TR simulation mode.	N/A	
2.	Configure TRIS with the large channel table. Do this by executing scripts and choosing the script trChanBig.tri.	N/A	
3.	Connect the DUT RF to the Motorola Cert Plant. Power-cycle the DUT and wait until it becomes operational.	N/A	
4.	Connect DUT and TRIS as shown in Figure 1.	N/A	
5.	From TRIS trace window check that DUT performed TRIF initialization sequence. Check for tr_init_req(), tr_init_rsp(), challenge_req(), challenge_rsp() authenticated.	N/A	
6.	Check that UDCP sent channel_table_req() APDU.	N/A	
7.	From TRIS trace window check that channel_table_req() APDU is well-formed (no parse errors) and conforms to the following: - channel_table_req_tag is 0x0107 trif_revision_code is 0x01	N/A	
8.	Using DUT remote, navigate menu to Tuning Adapter and request and view the TR diagnostics.	N/A	
9.	From TRIS trace window, verify the DUT send tr_diag_req() APDU well-formed (no parse errors) and conforms to the following: tr_diag_req_tag is 0x0401 trif_revision_code is 0x01 request_id (write down #)		REQ3812 REQ3807 REQ3871
10.	Verify the DUT displays the TR diagnostics that are received from the TR diagnostic query. Note: A user could cancel the request by exiting the mechanism.		REQ3872
11.	Send a broadcast head-end system EAS message. Verify the DUT DOES NOT display the TR diagnostics during the EAS message.		REQ3875
12.	Put DUT in Standby mode. Verify the diagnostics screen is interrupted and ends its display when going in standby. (Duh)		REQ3874
13.	Close down the TRIS application when the test is complete.	N/A	

11 TEST: DEVICE INTEROPERABILITY

11.1 Interoperability Tests Equipment Setup

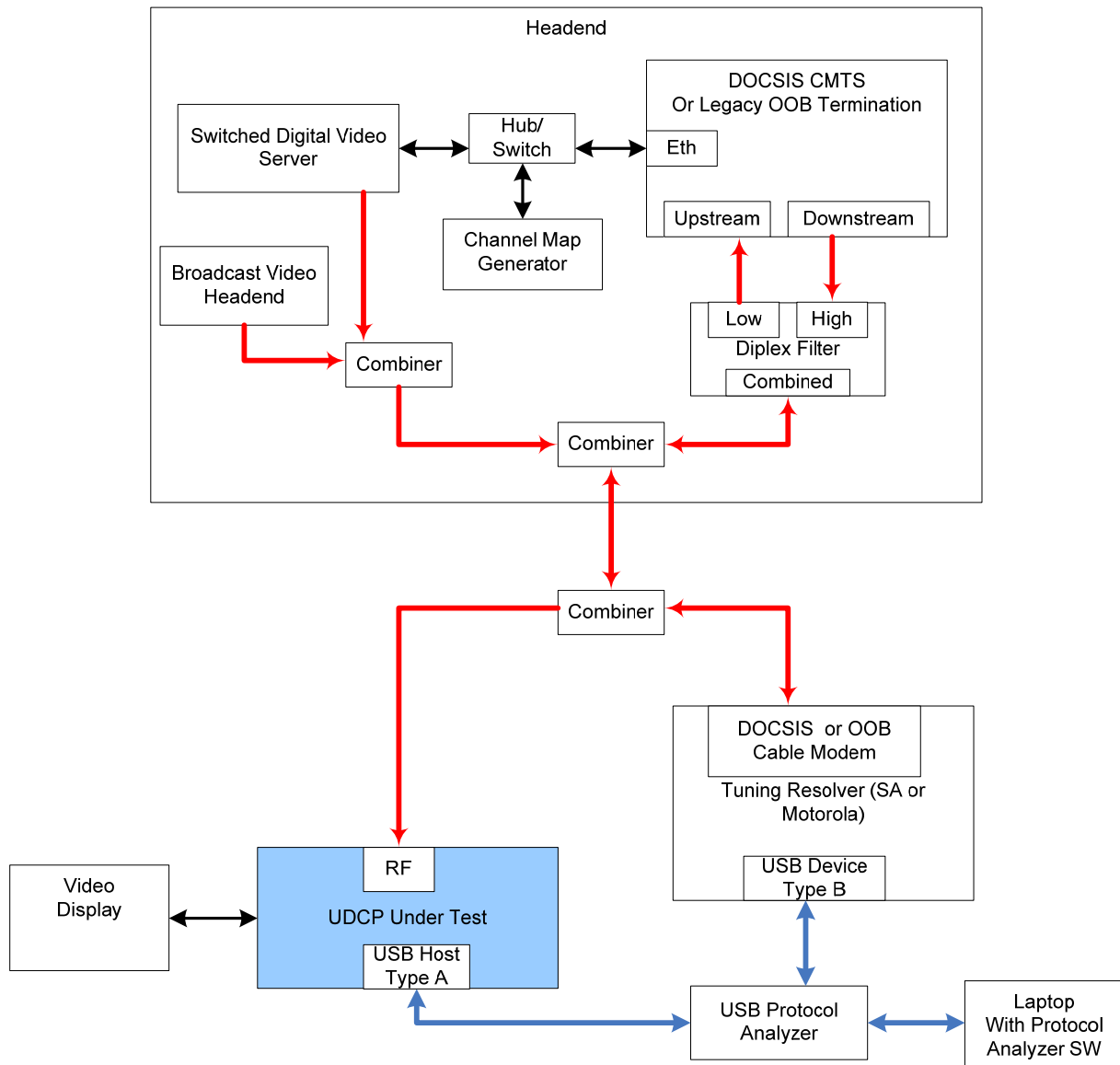


Figure 2 Interoperability Test Setup

11.2 Interoperability Tests

The TRIF-UDCP will be tested on cable system that utilizes SDV services and provides a Tuning Resolver solution. To date, both Motorola and SA have such systems.

CableLabs reserves the right to perform any type of Interoperability test that complies with TRIF specification.

Run these tests against:

Motorola TR device and SA TR device

11.2.1 Testcase: Power-up scenarios and channel map acquisition

Procedure 1: UDCP, then TR

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR (both powered-off).	N/A	
2.	Power up the DUT and make it operational.	N/A	
3.	After DUT becomes operational power-cycle the TR.	N/A	
4.	Verify that UDCP detects the TR and acquires channel map over TRIF.		
5.	Verify that UDCP can navigate through acquired channel map.		

Procedure 2: TR, then UDCP

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR (both powered-off).	N/A	
2.	Power-cycle the TR and wait until it becomes operational.	N/A	
3.	Power-cycle the DUT.	N/A	
4.	Verify that UDCP detects the TR and acquires channel map over TRIF		
5.	Verify that UDCP can navigate through acquired channel map		

11.2.2 Testcase: SDV manager interaction

Procedure 1: Basic test (applicable to all types of UDCPs)

Step#	Procedure	Pass/Fail	Comments
1.	Configure SDV manager to provide several switched digital channels. Set the inactivity period to 5 minutes. Reference annex's B & C.	N/A	
2.	Connect UDCP with TR and make both operational.	N/A	
3.	Wait until UDCP acquires channel map from the TR.	N/A	
4.	Verify that browsing switched digital channels is possible. From SDV manager console confirm that number of receiving devices changes accordingly.		
5.	Tune to a switched digital channel so there's only one device (the DUT) receiving the content. Make sure that only one tuner is in use (i.e. no DVR or PIP).		
6.	Wait five minutes. Do not perform any kind of RCU activity during that time.		
7.	Verify that UDCP shows user inactivity dialog. Do *not* confirm activity.		
8.	Confirm that SDV manager eventually removes the channel from the QAM after a defined "no-users" period.		
9.	Verify that UDCP shows a dialog informing of channel unavailability.		
10.	Tune to a different switched digital channel.		
11.	Wait five minutes. Do not perform any kind of RCU activity during that time.		
12.	Verify that UDCP shows user inactivity dialog.		
13.	Confirm that the channel is still being watched.		
14.	Confirm that SDV manager does not remove the channel from the QAM after a defined "no-users" period.		

Step#	Procedure	Pass/Fail	Comments
15.	From SDV manager console forcibly remove the channel currently being received by the DUT.		
16.	Verify that UDCP presents "channel temporarily unavailable" dialog.		
17.	Tune to a different switched digital channel.		
18.	This step requires forcing a SDV network outage. Do this according to head-end vendor instructions. If none are available, then the comments for this test should say "cannot be performed on: (Motorola or SA/Cisco)"		
19.	Verify that UDCP presents "network difficulties" dialog.		

Procedure 2: Multi-tuner test (applicable to multi-tuner UDCPs)

TBD

11.2.3 Testcase: TR diagnostics

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Enter a defined key sequence to invoke TR diagnostics.	N/A	
4.	Verify that UDCP presents TR diagnostic screen.		

11.2.4 Testcase: EAS messaging

Procedure 1: Maximum priority ("force tune") on analog channel

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Insert the CableCARD.	N/A	
4.	Tune to any analog channel.	N/A	
5.	Send an EAS force tune message from the head-end.	N/A	
6.	Once the Force tune has expired the DUT shall return to its original viewing channel when complete.		

Procedure 2: Maximum priority ("force tune") on digital channel

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Insert the CableCARD.	N/A	
4.	Tune to any digital channel.	N/A	
5.	Send an EAS force tune message from the head-end.	N/A	
6.	Once the Force tune has expired the DUT shall return to its original viewing channel when complete.		

Procedure 3: Medium priority (“scroll”) on analog channel

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Insert the CableCARD.	N/A	
4.	Tune to any analog channel.	N/A	
5.	Send an EAS force tune message from the head-end.	N/A	
6.	Verify that DUT shows the message text scrolling on the screen.		

Procedure 4: Medium priority (“scroll”) on digital channel

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Insert the CableCARD.	N/A	
4.	Tune to any digital channel.	N/A	
5.	Send an EAS scroll message from the head-end.	N/A	
6.	Verify that DUT shows the message text scrolling on the screen.		

11.2.5 Testcase: CableCard Firmware Upgrade

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Insert the CableCARD.	N/A	
4.	Tune to any digital channel.	N/A	
5.	From head-end console send the CableCARD firmware upgrade.	N/A	
6.	Verify that CableCARD firmware upgrade process completes successfully.		

11.2.6 Testcase: Head end Interoperability

Step#	Procedure	Pass/Fail	Comments
1.	Connect UDCP with TR and make both operational.	N/A	
2.	Wait until UDCP acquires channel map from the TR.	N/A	
3.	Insert the CableCARD.	N/A	
4.	Tune to any analog channel.	N/A	
5.	Verify audio and video on TV.		
6.	Tune to any unencrypted digital channel from QAM64 multiplex.	N/A	
7.	Verify audio and video on TV.		
8.	Tune to any unencrypted digital channel from QAM256 multiplex.	N/A	
9.	Verify audio and video on TV.		

Step#	Procedure	Pass/Fail	Comments
10.	Verify that the DUT's front panel LED or onscreen display, etc. on the correctly displays the channel number of the analog or digital service being displayed on the TV. (Verify a service at lower, middle and upper frequencies)		
11.	De-authorize the CableCARD	N/A	
12.	Verify that DUT presents an MMI dialog that asks the user to call in and register the CableCARD and host mating/addresses.		
13.	Tune to any "high value content" channel (refer to appropriate channel map).	N/A	
14.	Verify that CableCARD – DUT combination cannot view the high value content.		
15.	Authorize the CableCARD to receive high value content	N/A	
16.	Tune to any "high value content" channel (refer to appropriate channel map).	N/A	
17.	Verify audio and video on TV.		
18.	De-authorize the CableCARD.	N/A	
19.	Verify that CableCARD – DUT combination cannot view the high value content.		
20.	Verify the correct channel map is loaded into the DUT by using remote control to display each channel and comparing the channel number on DUT's front panel LED or onscreen display with the channel map and channel content.		
21.	Verify that DUT responds correctly to a "channel up" and "channel down" remote control commands		
22.	Tune to last channel in the channel map.	N/A	
23.	Verify that DUT tunes to first channel in the channel map upon receiving "channel up" remote control command.		
24.	Tune to unencrypted digital channel from QAM64 multiplex.	N/A	
25.	Verify audio and video on TV.		
26.	Physically remove signal cable from the back of the DUT. Wait 10 seconds and reconnect.	N/A	
27.	Verify that DUT's able to re-acquire QAM64 FAT channel.		
28.	Tune to unencrypted digital channel from QAM256 multiplex.	N/A	
29.	Verify audio and video on TV.		
30.	Physically remove signal cable from the back of the DUT. Wait 10 seconds and reconnect.	N/A	
31.	Verify that DUT's able to re-acquire QAM256 FAT channel.		
32.	Verify that DUT can display closed captioning on a TV from a digital multiplex. Tune to digital channel and verify the closed captioning on the TV. Note: Not all content is closed captioned, ie; some commercials do not have closed captioning.		
33.	Verify that DUT can display closed captioning on a TV from an analog service.		

Step#	Procedure	Pass/Fail	Comments
	Tune to analog channel and verify the closed captioning on the TV.		
34.	<p>Verify that the DUT can pass through content advisory information present in the VBI lines to a TV with a V-Chip present.</p> <p>To do this set up a TV with rating control via remote control for TV. Set up V-Chip to block any content over a rating "G". Channel through the channel map for analog and digital services and verify the TV has the ability to read the VBI lines and block the video with a rating above "G".</p>		

11.2.7 Testcase: Tuning Adapter Upgrade

[TBD]

12 TEST: VENDOR DOCUMENTATION

12.1.1 Testcase: Required documentation

The Tuning Adapter UDCP vendor must provide documentation affirming the compliance of the TA-UDCP to the requirements listed in the section.

REQ Tag	Requirement text	Type	Documentation Verified (Pass / Fail)
REQ3720	The UDCP USB interface SHALL meet the full USB power requirements as defined in [USB2.0].	Mandatory	
REQ3728	The UDCP SHALL implement full-speed bulk transfers.	Mandatory	
REQ3731	The UDCP SHALL send a single APDU in a single bulk transfer according to section 5.8 of [USB2.0].	Mandatory	
REQ3746	The UDCP SHALL respond to a USB Reset with a complete re-initialization, including re-authentication.	Mandatory	
REQ3751	The UDCP SHALL be able to interface and operate with a TR that is operating at a previous version of this specification.	Mandatory	
REQ3782	In the resolve_tuning_req() APDU, the UDCP SHALL report tuner_use_status values associated with opportunistic or speculative recording that was not user-initiated as 0x6.	Mandatory	
REQ3862	The UDCP SHALL NOT send a resolve_tuning_cnf() APDU with the tuner_use_status = 0x5 (inactive).	Mandatory	
REQ3804	If the UDCP chooses not to display the message in the APDU, the UDCP SHALL display an equivalent message.	Mandatory	
REQ3810	The UDCP SHALL only request URLs that were obtained from links found in MMI pages originating from a prior tr_diag_rsp() message.	Mandatory	
REQ3873	The UDCP SHALL NOT display the TR diagnostics if the request_id of the tr_diag_rsp() APDU does not equal the request_id of a previously transmitted tr_diag_req() APDU.	Mandatory	
REQ3817	The UDCP SHALL utilize a unique enumerated LTSID (local transport stream identifier) for each tuner in the UDCP.	Mandatory	
REQ3818	The UDCP SHALL NOT change the relationship between the LTSID and the tuner after reboot.	Mandatory	

REQ3864	The UDCP SHALL NOT use the tuning parameters obtained from a prior resolve_tuning_rsp() or resolve_tuning_update() APDU for rendering, recording, or transmitting further output from the associated tuner tuned according to those parameters after sending a udcp_status_update() with the tuner_use_status = 0x5 (inactive) for the corresponding LTSID.	Mandatory	
REQ3827	In the udcp_status function, the UDCP SHALL report tuner_use_status values associated with opportunistic or speculative recording that was not user-initiated as 0x6.	Mandatory	

12.1.2 Testcase: Optional Requirements

Requirements listed in this section are non-mandatory requirements that the Tuning Resolvers may or may not implement. They are listed here to provide traceability.

REQ Tag	Requirement text	Type
REQ3729	The UDCP MAY implement high-speed bulk transfers.	Optional
REQ3757	After two failures of receiving the tr_init_rsp() APDU, the UDCP MAY declare the TR inoperable and display a message stating such to the user.	Optional
REQ3770	The UDCP MAY conclude that authentication has failed if it does not receive a tr_hmac_key_send() APDU within five seconds.	Optional
REQ3790	Anytime after receiving the tr_init_rsp() APDU, the UDCP MAY send a tr_status_req() APDU.	Optional
REQ3809	When requesting the root page of diagnostic data provided by the TR, the UDCP MAY optionally specify a url_length of 0x0000 without specifying the actual locator in the url_byte loop that follows.	Optional
REQ3870	The UDCP MAY set the user_activity_detected_flag if the tuner_use_status is 'inactive'.	Optional

Annex A TRIS UDCP ATP scripts



scriptlist.tri



trChanBig.tri



trChanMotTest.tri



trInitIgnore.tri



trInitRspBadAuth.tri



trInitRspDelay4500.tri



trStatusFWUpgrade.tri



trStatusNotReady.tri



trTrMessage00.tri



trTrMessage01.tri



trTrMessage10.tri



trTrMessage11.tri



trTuneRsp01.tri



trTuneRsp02.tri



trTuneRsp03.tri



trTuneRsp06.tri



trTuneUp125NewParams.tri



trTuneUp125NewParams126.tri



trTuneUpdate125_1.tri



trTuneUpdate125_2.tri



trUdcpStatusReq.tri

Annex B Motorola SDV head-end Specifics

These instructions have been provided by Motorola Tuning Adapter group.

To set the inactivity timeout for required 5 minutes follow these steps.

On the SVM GUI, SDV Policy --> Channel Change [tab], change 'Last User Activity Report Interval' to 240 [sec]. Click 'Save' in lower left corner.

On the SVM GUI, SDV Policy --> Client Operations [tab], change 'Prompt User Interaction Timeout' to 60 [sec].

On the SVM GUI, SDV Policy --> Client Operations [tab], change 'Channel Inactivity Time' to 1 [minute]. Click 'Save' in lower left corner.

Procedure Detail:

The above steps will cause a barker to be displayed for one minute after four minutes of user inactivity. Then one minute later, the channel becomes a candidate for teardown. If the user set bandwidth utilization threshold is not exceeded, the channel will not get torn down. If the user set bandwidth utilization threshold is exceeded, the channel will get torn down.

Force an inactivity timeout

Don't do anything to the STB for a few minutes with the above settings in place.

There are two different ways to force a switched channel off the plant.

1. On the SVOM GUI, Channel Lists --> Detail [far right], highlight the switched channel you want to force off the plant, then click 'Out of Service' in the lower left of the window.
2. Simply delete the channel from the SVOM SDV Channel List, then go to 'Server Devices' and click 'Send Cfg' for the SVM application[s]. It will take a few minutes for the "mini_carousel" to pick up on the change. But you can always delete the service from the video source to speed up the process.

Procedure Detail:

For the completely seamless transition as each user leaves the switched version channel and then return to the switched channel, they will be watching the broadcast version of the program rather than the switched version.

Annex C SA/Cisco Head-End SDV Specifics

These instructions were provided by SA/Cisco Tuning Adapter group.

Note:

Currently, these instructions have not been successfully implemented at CableLabs.

To set the inactivity timeout for required 5 minutes follow these steps.

1. Log into the SDV server (admin, admin)
2. Go to Offered Programs and scroll all the way to the right
3. Under "Recapture Acknowledgement" time, change the time on each channel to 300 seconds.
(Currently its set at 600 seconds)
4. Don't forget to hit Commit changes.
5. Keep in mind that if the DNCS is rebooted or stopped/started the sdv server will revert back to its original settings.