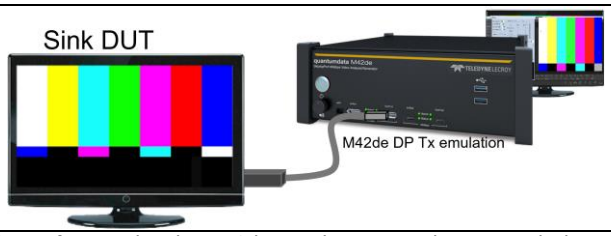
qdPrime™ Test Suite: Getting Started

1 qdPrime Overview

The qdPrime™ Test Suite is a software option for the quantumdata M42d/de analyzer generator platform that expands the scope of functional and interoperability testing for DisplayPort 2.1 devices. qdPrime utilizes a menu-based interface allowing users to define test parameters which are used to systematically verify interoperability with a diverse set of DisplayPort configurations. Each individual test generates a pass/fail result with detailed logs to provide a repeatable snapshot of device interoperability. There are separate test suites and license keys needed to run the qdPrime Source and Sink test cases.

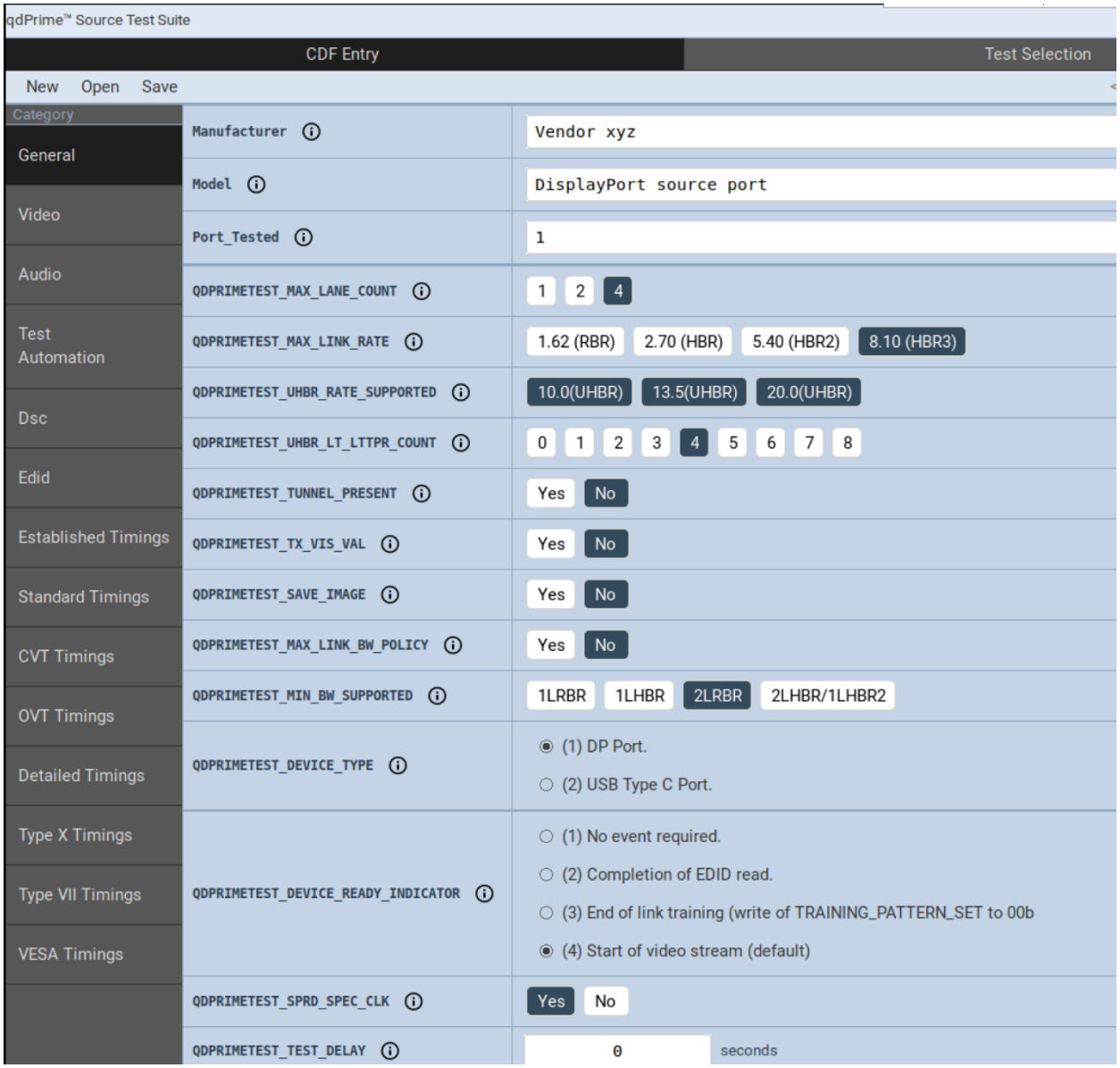
Role of CDF Files

Similar to the Capabilities Declaration Form (CDF) files used during VESA compliance testing, both the Source and Sink Tests require a CDF which is entered / saved in CDF tab. These entries describe the source or sink and primarily serve to identify specific test parameters that the M42de test platform will utilize when running qdPrime test cases.

	
Sink Testing – using the M42de as a known good source, qdPrime can extract the sink capabilities and systematically test each configuration the sink supports including every CVT, OVT and VESA format reported. Sink specific tests also utilize CDF entries in some tests to constrain the test conditions used by qdPrime.	**Source Testing** – qdPrime Test Suite for source verification requires a Capabilities Declaration Form (CDF) to constrain specific test parameters the tester will emulate. The M42de emulating a known-good-sink will use the CDF to dynamically create test parameters and verify specific test conditions on the GPU.

2 Creating CDF Files (Source DUT)

Upon opening qdPrime Test Suite, the first window displayed is the **CDF Entry** tab, as shown (below). Several tabs on the left allow users to input specific test parameters. During the Test Execution stage, the test platform will expose or request these parameters to ensure the desired test conditions are covered.

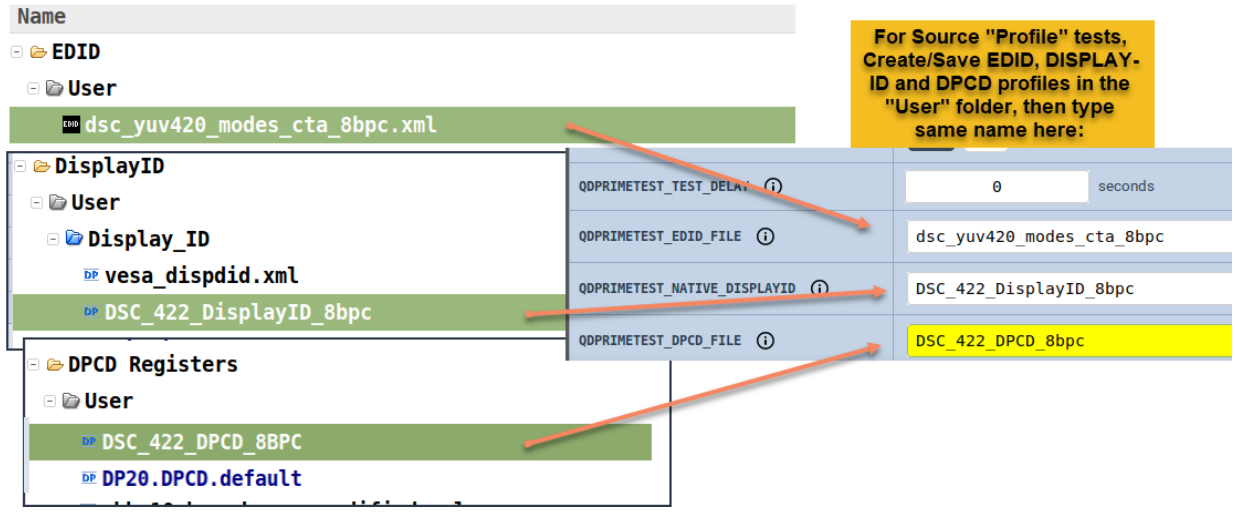


To configure for qdPrime Source Tests, it is necessary to input operating parameters summarized below:

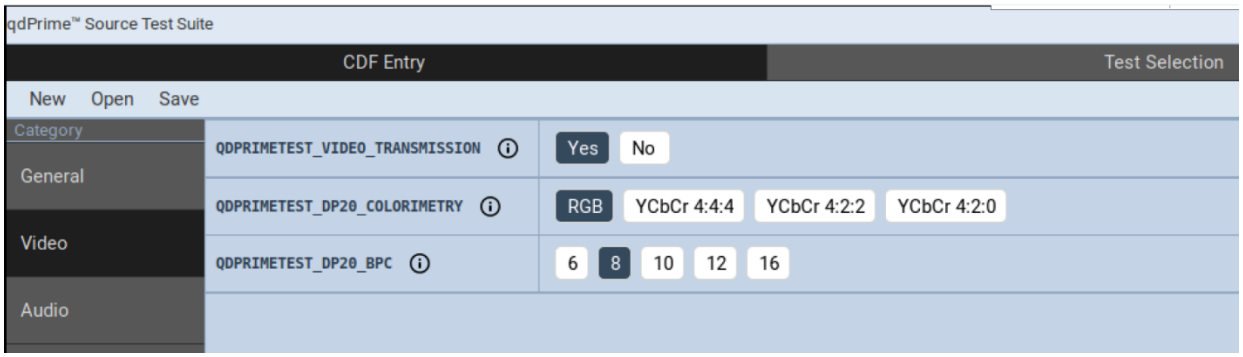
- General Tab – Basic operational parameters for the link When multiple items are selected, this instructs qdPrime to generate a separate iteration using each parameter for each of the defined timings.
    - Max_Lane Count: Selecting 4 lanes will cause all tests to repeat at 1, 2, and 4 lanes.
    - Max_Link_Rate: Selecting HBR3 rate will cause all tests to repeat at HBR3, HBR2, HBR & RBR rates
    - LT_LTTPR_Count: Defines the number of Retimers on-board the source DUT
    - Tunnel_Present: Defines if the source DUT is connecting through a DP Out Adapter

- TX_VIS_VAL: Enables **qdPrime to** prompt **the user** to perform visual verification **of the received** image**s after each test iteration.** See the section **Visual Verification Step**

- Save_Image: Saves the actual rendered Image file in the test result folder to facilitate troubleshooting or debug on the Source that generated the image.
- Max_Link_BW_Policy: Enables the tests to use Max Bandwidth policies for 8b10b link training.
- Min BW_Supported: Defines the minimum link bandwidth supported to prevent testing of unsupported link rates.
- Device_Type: Defines which port on the test platform (Type-C or DP) will be connected to the source DUT.
- Device_Ready_Indicator: Defines how and when the test platform (as sink) will initiate the test condition (helpful for early development devices where additional delay may be needed).
- Sprd_Spec_Clk: Defines if the source DUT will use Spread Spectrum Clocking on the video stream.
- Test_Delay: Allows user to specify an additional delay interval between tests to allow manual configuration or slow power-on DUTs (helpful for early development devices).
- EDID_File: Specifically for source profile tests: Create an EDID.xml file in the User folder, then type the name of the file here which will be used by qdPrime during the test (below).
- Native_DisplayID Specifically for source profile tests: Create a Native DisplayID file.xml file in the User folder, then type the name of the file here (below).
- DPCD_File: Specifically for source profile tests: Create an DPCD.xml file in the User folder, then type the name of the file which points qdPrime to the DPCD "profile" (below).

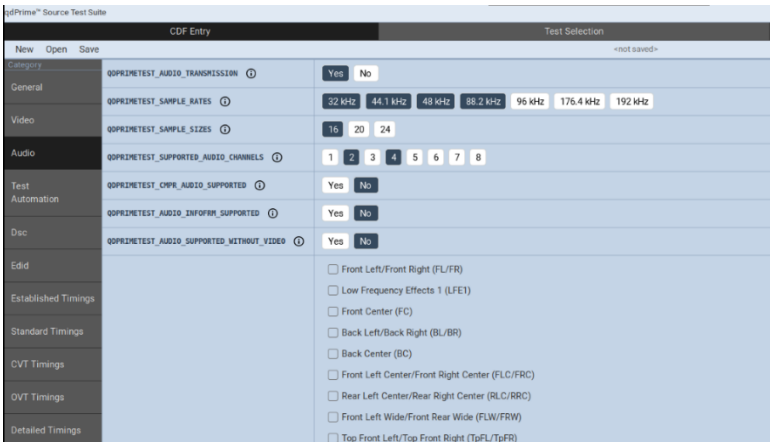


- Video Tab - Video specific parameters including color space and bits-per-component are entered here. When multiple items are selected, qdPrime will generate a separate test iteration using each parameter for all selected timings.



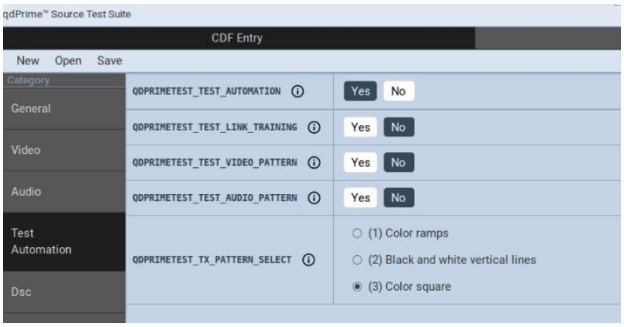
- Video_Transmission – Defines if source DUT supports video transmission; (disabling Video is only appropriate for audio-only testing).
- DP20_Colorimetry – Defines the color space that will be tested for the Source DUT
- DP20_Bpc – Defines the bits-per-color for UHBR modes

- Audio Tab – Audio specific parameters including sample rate and channels supported by the UUT are entered here. When multiple items are selected, this instructs qdPrime to generate a separate iteration using each parameter for all selected timings.



- Audio_Transmission – Defines if source DUT supports audio transmission; (disabling audio will disable audio checks).
- Sample_Rates – Defines the sample rates that will be tested for the Source DUT.
- Sample_Size – Defines the audio sample depth that will be tested for the Source DUT.
- Supported_Audio_Channels – Defines how many audio channels will be tested for the Source DUT.
- CMPR_Audio_Supported – Defines whether CMPR audio will be tested for the Source DUT.
- Audio_Info_Frame - Defines whether CMPR audio will be tested for the Source DUT.
- Audio_Supported_Without_Video: Defines whether audio-only generation will be tested for the Source DUT.
- Speaker_Allocation: Defines which speaker configurations will be tested for the Source DUT.

- Test Automation Tab – Automation specific parameters use the VESA automation framework defined for source DUTs. When enabled, qdPrime will use the automated testing register set to direct the source into specific test modes or to validate specific test steps.



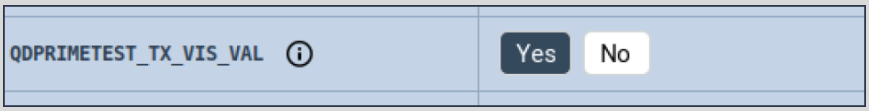
- Test_Automation – Defines if source DUT supports VESA test automation.
- Test_Link_Training – Defines if the source DUT supports the automation register set for “Link Training”.
- Test_Video_Pattern – Defines if the source DUT supports the automation register set for “video pattern checks”.
- Test_Audio_Pattern – Defines if the source DUT supports the automation register set for “Audio Pattern checks”.
- Test_Pattern_Select – Defines which if any of the common pre-defined patterns the source supports. The selection here assists qdPrime in image verification stage by defining which of the pre-defined image types the Source DUT will use on the next test run.

DSC Tab– If the source DUT supports DSC, the capabilities entered here communicate to qdPrime which parameters the source supports when generating compressed images. When running the DSC specific source tests, the Tester (as sink) will advertise and check these specific parameters (items that support multiple selection here will be individually tested with separate iterations).

- DSC_TX_Color – Specifies which DSC Color Space will be tested by qdPrime.
- DSC_TX_Color_Depth – Specifies which DSC Color Depth will be tested by qdPrime.
- DSC_DSC_Block_Prediction – Specifies if DSC Block prediction will be tested by qdPrime.
- DSC_Sink_Slice_Cap – Specifies the DSC slice width that will presented and verified by qdPrime.
- DSC_Sink_Buffer_Bit_Depth – Specifies the bit-depth that will be that will presented and verified by qdPrime.
- DSC_Sink_Bit_Pixel_Increment– Specifies the bit pixel increment that will presented and verified by qdPrime.

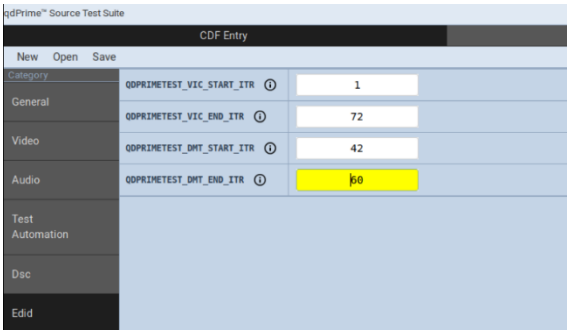
- **Visual Verification Step**

Under the General tab, the TX_VIS_VAL (Source DUTs) & TEST_VIS_VERIFY (Sink DUTs) parameter (below) should be enabled on the first run of source or sink tests. This will cause qdPrime to prompt the user to visually verify each image in the real-time display (M42de Rx port) or for sink DUT (M42de Tx port) upon completion of each test.

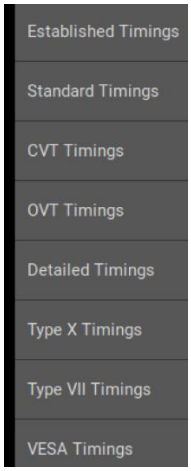


Successful visual verification of the image will be followed by a prompt to save the CRC for the image to the M42de internal database. This preserves the CRC for images that render correctly and any future test runs that use the same parameters and image will be automatically verified by qdPrime. The test suite will check the CRC value to verify the image has the same CRC regardless of whether the TX_VIS_VAL is enabled or not. The QDPRIMETEST_CACHE_CRC allows user to specify if he wants to always save the CRC in the database or not. Setting it to FALSE, will prompt the user everytime to save the CRC or not whereas setting it to TRUE, will automatically save the CRC by default in the database without prompting user.

- EDID tab - Allows the default range for VIC and DMT values to be defined for range iterator test runs. Experiment by selecting only a small range to assess the execution time needed for larger test runs.



  - VIC_Start_ITR – specifies starting VIC for ‘range iterator’ test cases.
  - VIC_End_ITR – specifies ending VIC for ‘range iterator’ test cases.
  - DMT_Start_ITR – specifies starting DMT for ‘range iterator’ test cases.
  - DMT_End_ITR – specifies ending DMT for ‘range iterator’ test cases.



Established timings includes all VESA timings that can be advertised and supported by the M42d/de platform.
Standard timings defines all Generalized Timing Formula (GTF) timings that can be advertised and supported by the M42d/de platform.
Coordinated Video Timings (CVT) for the latest VESA specification that can be advertised and supported by the M42d/de platform.
Optimized Video Timings (OVT) refers to the CTA-861 standard for Extended and supported by the M42d/de platform.
Detailed Timings provides a menu based interface to create custom timings including active lines and pixels. All detailed timings must be valid and adhere to the 18-byte Detailed Timing Descriptors (DTD).
Type X provides a menu based interface to create Extended Block Type Tag “X”. It must be a “valid extended block format” to be verified with qdPrime.
Type VII provides a menu based interface to create Extended Block Type Tag “VII”. It must be a “valid extended block format” to be verified with qdPrime.
VESA timings includes the most common timings defined by VESA that can be selected and supported by the M42d/de platform.

Entering Selections in the Timing Tabs:

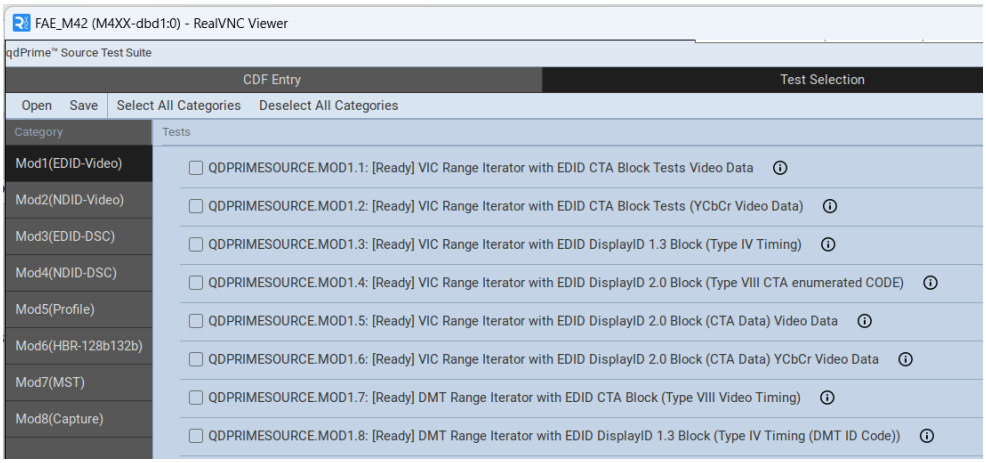
The timings tabs provide a menu based selection of timing specific parameters to be tested. Users can select standard timings common to VESA or GTF standards. The Detailed tab allows users to create “custom timings” that can be alternately selected for testing in the “Detailed Timings” Test Selection menu. It is only necessary to enter parameters in the timing tabs if users intend to test these specific timings in the Test Selection Menu (Source DUTs) step.

When selecting specific timings or adding custom timings in the tabs above, these parameters will instruct qdPrime to create a series of tests that systematically run each of the selected timings in combination with all other parameters (link width, lane rate, bpc, etc…). In cases where the CDF parameters exceed the bandwidth supported by the source, the test will be skipped. Please note, there can be hundreds of variations needed to cover even a subset of the various timing selections (right). Experiment by selecting only a single timing entry to assess the execution time needed for larger test runs.

Important Note: It is essential to save the CDF file after making any modifications.

3 Test Selection Menu (Source DUTs)

The Test Selection menu is organized around modules that contain specific tests. The modules are broadly organized around tests that use standard “EDID” (Mod1) or “Native DisplayID” (Mod2) Extended block formats. Notice there are also several modules (Mod3 & Mod4) that perform many of the same tests but operate with “DSC” enabled DUTs.



The modules 1 through 4 are the primary modules that should be used to deterministically test the different timings. The modules 5 through 8 are special test cases that provide specific test conditions described below: Profile tests (Mod5) are designed to use vendor provided EDID and DPCD files. These allow source DUTs to test against a known sink profile. The HBR-128b132b (Mod6) tests provide proprietary subset of the VESA Link Layer tests for DisplayPort 2.1 that run at 2.75Ghz to enable pre-silicon verification of UHBR encoding (128b/132b)

typically running in an FPGA. The MST tests (Mod7) are intended to provide the building blocks for validating DisplayPort 2.1 MST operation. The Capture (Mod8) tests will verify various proprietary test conditions.

Source Mod1 EDID-Video Tests:

The module 1 are the primary tests designed to verify the VIC, DMT, CVT, OVT, VESA and Custom ranges for Source DUTs when presented with various EDID formats:

QDPRIMESOURCE.MOD1.1 VIC Range Iterator with EDID CTA Block Tests Video Data Verifies the video according to the Video Data section under CTA Block.
QDPRIMESOURCE.MOD1.2 VIC Range Iterator with EDID CTA Block Tests (YCbCr Video Data) Verifies the video according to the YCbCr 4:2:0 Video Data section under CTA Block.
QDPRIMESOURCE.MOD1.3 VIC Range Iterator with EDID DisplayID 1.3 Block (Type IV Timing) Verifies the video according to the Type IV Timing (Enumerated Code CTA) section under DisplayID 1.3 Block.
QDPRIMESOURCE.MOD1.4 VIC Range Iterator with EDID DisplayID 2.0 Block (Type VIII CTA enumerated CODE) Verifies the video according to the Type VIII Timing (Enumerated Code CTA) section under DisplayID 2.0 Block.
QDPRIMESOURCE.MOD1.5 VIC Range Iterator with EDID DisplayID 2.0 Block (CTA Data) Video Data Verifies the video according to the Video Data section under CTA Data Section in DisplayID 2.0 Block.
QDPRIMESOURCE.MOD1.6 VIC Range Iterator with EDID DisplayID 2.0 Block (CTA Data) YCbCr Video Data Verifies the video according to the YCbCr 4:2:0 Video Data section under CTA Data Section in DisplayID 2.0 Block.
QDPRIMESOURCE.MOD1.7 DMT Range Iterator with EDID CTA Block (Type VIII Video Timing) Verifies the video according to the Type VIII Video Timing under CTA Block.
QDPRIMESOURCE.MOD1.8 DMT Range Iterator with EDID DisplayID 1.3 Block (Type IV Timing (DMT ID Code)) Verifies the video according to the Type IV Timing (Enumerated Code DMT) section under DisplayID 1.3 Block.
QDPRIMESOURCE.MOD1.9 DMT Range Iterator with EDID DisplayID 2.0 Block (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under DisplayID 2.0 Block.
QDPRIMESOURCE.MOD1.10 DMT Range Iterator with EDID DisplayID 2.0 Block (CTA Data) (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under CTA Data sectionof DisplayID 2.0 Block. The test intends to validate all the DMT IDs provided in CDF and ignore any interlaced formats by generating and applying the respective EDID's.
QDPRIMESOURCE.MOD1.11 Established Timings test with EDID Base Block Verifies the video according to the Established Timings by ignoring any interlaced formats while generating and applying EDIDs in the Base block.
QDPRIMESOURCE.MOD1.12 Standard Timings test with EDID Base Block Verifies the video according to the Standard Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD1.13 CVT Timings test with EDID Base Block Verifies the video according to the CVT Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD1.14 OVT Timings test with EDID CTA Block (Video Format Data) Verifies the video according to the OVT Timings provided in Video Format Data Section of CTA block.
QDPRIMESOURCE.MOD1.15 Detailed Timings test with EDID Base Block Verifies the video according to the Detailed Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD1.16 Type X Timings test with EDID CTA Block Verifies the video according to the Type X Timings provided in the CTA block.
QDPRIMESOURCE.MOD1.17 Type X Timings test with EDID DisplayID 2.0 Block Verifies the video according to the Type X Timings provided in the DisplayID 2.0 block.
QDPRIMESOURCE.MOD1.18 Vesa Timings test with EDID DisplayID 1.3 Block Verifies the video according to the Vesa Timings provided in the DisplayID 1.3 Block.
QDPRIMESOURCE.MOD1.19 Detailed timings with Edid CTA block (Detailed Timing Descriptors) Verifies the video according to the Detailed timings provided in the Edid CTA block.
QDPRIMESOURCE.MOD1.20 Detailed timings with Edid DisplayID 1.3 Ext block (Type I) Verifies the video according to the Detailed timings provided in the Edid DisplayID 1.3 Ext block (Type I).
QDPRIMESOURCE.MOD1.21 Detailed timings with Edid DisplayID 2.0 Ext Block (Type VII) Verifies the video according to the Detailed timings provided in the Edid DisplayID 2.0 Ext Block (Type VII).
QDPRIMESOURCE.MOD1.22 Detailed timings with Edid DisplayID 2.0 Ext Block (CTA Data Type VII)

Verifies the video according to the Detailed timings provided in the Edid DisplayID 2.0 Ext Block (CTA Data Type VII).
QDPRIMESOURCE.MOD1.23 Detailed timings with Edid DisplayID 1.3 Ext Block (Type II) Verifies the video according to the Detailed timings provided in the Edid DisplayID 1.3 Ext Block (Type II).
QDPRIMESOURCE.MOD1.24 Detailed timings with Edid DisplayID 1.3 Ext Block (Type VI) Verifies the video according to the Detailed timings provided in the Edid DisplayID 1.3 Ext Block (Type VI).

Source Mod2 DSC Tests:

The module 2 are the primary tests designed to verify the VIC, DMT, CVT, OVT, VESA and Custom ranges for Source DUTs when presented with Native Display-ID formats:

QDPRIMESOURCE.MOD2.1 VIC Range Iterator with Native DisplayID 1.3 (Type IV Timing) Verifies the video according to the Type IV Timing (Enumerated Code CTA) section under DisplayID 1.3.
QDPRIMESOURCE.MOD2.2 VIC Range Iterator with Native DisplayID 2.0 (Type VIII CTA enumerated CODE) Verifies the video according to the Type VIII Timing (Enumerated Code CTA) section under DisplayID 2.0.
QDPRIMESOURCE.MOD2.4 VIC Range Iterator with Native DisplayID 2.0 (CTA Data) YCbCr 4:2:0 Video Data Verifies the video according to the YCbCr 4:2:0 Video Data section under CTA Data Section in DisplayID 2.0.
QDPRIMESOURCE.MOD2.5 DMT Range Iterator with Native DisplayID 1.3 (Type IV Timing (DMT ID Code)) Verifies the video according to the Type IV Timing (Enumerated Code DMT) section under DisplayID 1.3.
QDPRIMESOURCE.MOD2.6 DMT Range Iterator with Native DisplayID 2.0 (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under DisplayID 2.0.
QDPRIMESOURCE.MOD2.7 DMT Range Iterator with Native DisplayID 2.0 (CTA Data) (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under CTA Data section of DisplayID 2.0.
QDPRIMESOURCE.MOD2.8 Type X Timings test with Native DisplayID 2.0 Verifies the video according to the Type X Timings provided in Native DisplayID 2.0.
QDPRIMESOURCE.MOD2.9 Type X Timings test with Native DisplayID 2.0 CTA Data Verifies the video according to the Type X Timings provided in Native DisplayID 2.0 CTA Data.
QDPRIMESOURCE.MOD2.10 Vesa Timings test with Native DisplayID 1.3 Verifies the video according to the Vesa Timings provided in Native DisplayID 1.3.
QDPRIMESOURCE.MOD2.11 OVT Timings test with Native DisplayID 2.0 CTA Data (Video Format Data) Verifies the video according to the OVT Timings provided in the Native DisplayID 2.0 CTA Data.
QDPRIMESOURCE.MOD2.12 Detailed timings with Native DisplayID 1.3 Ext block (Type I) Verifies the video according to the Type I Detailed Timings provided in the Native DisplayID 1.3 Ext block.
QDPRIMESOURCE.MOD2.13 Detailed timings with Native DisplayID 2.0 Ext Block (Type VII) Verifies the video according to the Type VII Detailed Timings provided in the Native DisplayID 2.0 Ext block.
QDPRIMESOURCE.MOD2.14 Detailed timings with Native DisplayID 2.0 Ext Block (CTA Data Type VII) Verifies the video according to the Type VII Detailed Timings provided in the Native DisplayID 2.0 CTA Ext block.
QDPRIMESOURCE.MOD2.15 Detailed timings with Native DisplayID 1.3 Ext block (Type II) Verifies the video according to the Type II Detailed Timings provided in the Native DisplayID 1.3 Ext block.
QDPRIMESOURCE.MOD2.16 Detailed timings with Native DisplayID 1.3 Ext block (Type VI) Verifies the video according to the Type VI Detailed Timings provided in the Native DisplayID 1.3 Ext block.

Source Mod3 DSC EDID-Video Tests:

The module 3 includes similar test cases as module 1 for Source DUTs that support DSC. This includes DSC operation for VIC, DMT, CVT, OVT, VESA and Custom ranges for Source DUTs when presented with various EDID formats:

QDPRIMESOURCE.MOD3.1 [DSC] VIC Range Iterator with EDID CTA Block Tests Video Data Verifies the DSC video according to the Video Data section under CTA Block.
QDPRIMESOURCE.MOD3.2 [DSC] VIC Range Iterator with EDID CTA Block Tests (YCbCr Video Data) Verifies the video according to the YCbCr 4:2:0 Video Data section under CTA Block.
QDPRIMESOURCE.MOD3.3 [DSC] VIC Range Iterator with EDID DisplayID 1.3 Block (Type IV Timing) Verifies the video according to the Type IV Timing (Enumerated Code CTA) section under DisplayID 1.3 Block.

QDPRIMESOURCE.MOD3.5 [DSC] VIC Range Iterator with EDID DisplayID 2.0 Block (CTA Data) Video Data Verifies the video according to the Video Data section under CTA Data Section in DisplayID 2.0 Block.
QDPRIMESOURCE.MOD3.6 [DSC] VIC Range Iterator w/ EDID DisplayID 2.0 Block (CTA Data) YCbCr Video Verifies the video according to the YCbCr 4:2:0 Video Data section under CTA Data Section in DisplayID 2.0 Block.
QDPRIMESOURCE.MOD3.7 [DSC] DMT Range Iterator with EDID CTA Block (Type VIII Video Timing) Verifies the video according to the Type VIII Video Timing under CTA Block.
QDPRIMESOURCE.MOD3.8 [DSC] DMT Range Iterator with EDID DisplayID 1.3 Block (Type IV Timing (DMT ID Code)) Verifies the video according to the Type IV Timing under EDID DisplayID 1.3 Block.
QDPRIMESOURCE.MOD3.9 [DSC] DMT Range Iterator with EDID DisplayID 2.0 Block (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing under CTA Block.
QDPRIMESOURCE.MOD3.10 [DSC] DMT Range Iterator with EDID DisplayID 2.0 Block (CTA Data) (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under CTA Data section of DisplayID 2.0 Block.
QDPRIMESOURCE.MOD3.11 [DSC] Established Timings test with EDID Base Block Verifies the video according to the Established Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD3.12 [DSC] Standard Timings test with EDID Base Block Verifies the video according to the Standard Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD3.13 [DSC] CVT Timings test with EDID Base Block Verifies the video according to the CVT Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD3.14 [DSC] OVT Timings test with EDID CTA Block (Video Format Data) Verifies the video according to the OVT Timings provided in Video Format Data Section of CTA block.
QDPRIMESOURCE.MOD3.15 [DSC] Detailed Timings test with EDID Base Block Verifies the video according to the Detailed Timings provided in the Base block of EDID.
QDPRIMESOURCE.MOD3.16 [DSC] Type X Timings test with EDID CTA Block Verifies the video according to the Type X Timings provided in the CTA block.
QDPRIMESOURCE.MOD3.17 [DSC] Type X Timings test with EDID DisplayID 2.0 Block Verifies the video according to the Type X Timings provided in the DisplayID 2.0 block.
QDPRIMESOURCE.MOD3.18 [DSC] Vesa Timings test with EDID DisplayID 1.3 Block Verifies the video according to the Vesa Timings provided in the DisplayID 1.3 Block.
QDPRIMESOURCE.MOD3.19 [DSC] Detailed timings with Edid CTA block (Detailed Timing Descriptors) Verifies the video according to the Detailed Timings provided in the CTA block.
QDPRIMESOURCE.MOD3.20 [DSC] Detailed timings with Edid DisplayID 1.3 Ext block (Type I) Verifies the video according to the Type I Timings provided in the DisplayID 1.3 Block.
QDPRIMESOURCE.MOD3.21 [DSC] Detailed timings with Edid DisplayID 2.0 Ext Block (Type VII) Verifies the video according to the Type VII Timings provided in the DisplayID 2.0 Block.
QDPRIMESOURCE.MOD3.22 [DSC] Detailed timings with Edid DisplayID 2.0 Ext Block (CTA Data Type VII) Verifies the video according to the Type VII Timings provided in the CTA block of DisplayID 2.0.
QDPRIMESOURCE.MOD3.23 [DSC] Detailed timings with Edid DisplayID 1.3 Ext block (Type II) Verifies the video according to the Type II Timings provided in the DisplayID 1.3 Block.
QDPRIMESOURCE.MOD3.24 [DSC] Detailed timings with Edid DisplayID 1.3 Ext block (Type VI) Verifies the video according to the Type VI Timings provided in the DisplayID 1.3 Block.

Source Mod4 DSC Native Display-ID Video Tests:

The module 4 includes similar tests as module 2 for Source DUTs that support DSC. This includes DSC operation for VIC, DMT, CVT, OVT, VESA and Custom ranges for Source DUTs when presented with Native Display-ID formats:

QDPRIMESOURCE.MOD4.1 [DSC] VIC Range Iterator with Native DisplayID 1.3 (Type IV Timing) Verifies the video according to the Type IV Timing (Enumerated Code CTA) section under DisplayID 1.3.
QDPRIMESOURCE.MOD4.2 [DSC] VIC Range Iterator with Native DisplayID 2.0 (Type VIII CTA enumerated CODE) Verifies the video according to the Type VIII Timing (Enumerated Code CTA) section under DisplayID 2.0
QDPRIMESOURCE.MOD4.3 [DSC] VIC Range Iterator with Native DisplayID 2.0 (CTA Data) Video Data

Verifies the video according to the Video Data section under CTA Data Section in DisplayID 2.0.
QDPRIMESOURCE.MOD4.4 [DSC] VIC Range Iterator with Native DisplayID 2.0 (CTA Data) YCbCr 4:2:0 Video Data Verifies the video according to the YCbCr 4:2:0 Video Data section under CTA Data Section in DisplayID 2.0.
QDPRIMESOURCE.MOD4.5 [DSC] DMT Range Iterator with Native DisplayID 1.3 (Type IV Timing (DMT ID Code)) Verifies the video according to the Type IV Timing (Enumerated Code DMT) section under DisplayID 1.3.
QDPRIMESOURCE.MOD4.6 [DSC] DMT Range Iterator with Native DisplayID 2.0 (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under DisplayID 2.0.
QDPRIMESOURCE.MOD4.7 [DSC] DMT Range Iterator with Native DisplayID 2.0 (CTA Data) (Type VIII Timing (DMT Code)) Verifies the video according to the Type VIII Timing (Enumerated Code DMT) section under CTA Data section of DisplayID 2.0.
QDPRIMESOURCE.MOD4.8 [DSC] Type X Timings test with Native DisplayID 2.0 Block Verifies the video according to the Type X Timings provided in Native DisplayID 2.0.
QDPRIMESOURCE.MOD4.9 [DSC] Type X Timings test with Native DisplayID 2.0 CTA Data Verifies the video according to the Type X Timings provided in Native DisplayID 2.0 CTA Data.
QDPRIMESOURCE.MOD4.10 [DSC] Vesa Timings test with Native DisplayID 1.3 Verifies the video according to the Vesa Timings provided in Native DisplayID 1.3.
QDPRIMESOURCE.MOD4.11 [DSC] OVT Timings test with Native DisplayID 2.0 CTA Data (Video Format Data) Verifies the video according to the OVT Timings provided in the Native DisplayID 2.0 CTA Data.
QDPRIMESOURCE.MOD4.12 [DSC] Detailed timings with Native DisplayID 1.3 Ext block (Type I) Verifies the video according to the Type I Timings provided in Native DisplayID 1.3.
QDPRIMESOURCE.MOD4.13 [DSC] Detailed timings with Native DisplayID 2.0 Ext Block (Type VII) Verifies the video according to the Type VII Timings provided in the Native DisplayID 2.0 Ext Block.
QDPRIMESOURCE.MOD4.14 [DSC] Detailed timings with Native DisplayID 2.0 Ext Block (CTA Data Type VII) Verifies the video according to the Type VII Timings provided in Native DisplayID 2.0 CTA Data.
QDPRIMESOURCE.MOD4.15 [DSC] Detailed timings with Native DisplayID 1.3 Ext block (Type II) Verifies the video according to the Type II Timings provided in Native DisplayID 1.3 using detailed Timings defined in CDF and ignoring any interlaced formats by generating and applying the respective NDID's using DSC max link rate, max lane count, and repeats for all colorimetry and BPC.
QDPRIMESOURCE.MOD4.16 [DSC] Detailed timings with Native DisplayID 1.3 Ext block (Type VI) Verifies the video according to the Type VI Timings provided in Native DisplayID 1.3.

Source Mod5 Profile Tests:

The module 5 tests are designed to use externally created EDID and DPCD .XML files. The M42de as sink will emulate or mimic these “real” or custom device “profiles”. These files can be saved from a real device or generated/edited using the M42de EDID and DPCD Editor utility. Separate tests are provided that use CTA-861 EDID or Native Display-ID as well as with or without DSC enabled.

QDPRIMESOURCE.MOD5.1 Sink Profiling Test with user EDID & DPCD files. This test profiles sinks for a source with user provided EDID & DPCD files.
QDPRIMESOURCE.MOD5.2 Sink Profiling Test with user native DisplayID & DPCD files. This test profiles sinks for a source with user provided native DisplayID & DPCD files.
QDPRIMESOURCE.MOD5.3 [DSC] Sink Profiling Test with user EDID & DPCD files. This test profiles sinks for a source with user provided EDID & DPCD files.
QDPRIMESOURCE.MOD5.4 [DSC] Sink Profiling Test with user native DisplayID & DPCD files. This test profiles sinks for a source with user provided native DisplayID & DPCD files.

Source Mod6 HBR 128b/132b VESA Link Layer tests:

The module 6 contains HBR-128b132b which is a proprietary subset of the VESA Link Layer tests for DisplayPort 2.1 that run at 2.75Ghz to enable pre-silicon verification of UHBR encoding (128b/132b).

QDPRIMESOURCE.MOD6.1 4.3.1.14 Successful Link Training at All Supported Lane Counts and UHBR Link Speeds (HBR 128b/132b);Please refer to the test description of 5.3.1.14.
QDPRIMESOURCE.MOD6.2 4.3.1.15 Successful Link Training Upon HPD Plug Event for UHBR speed (HBR 128b/132b); Please refer to the test description of 5.3.1.15.
QDPRIMESOURCE.MOD6.3 4.3.1.16 Successful Link Training when EQ done at 20th loop during channel EQ phase (HBR 128b/132b); Please refer to the test description of 5.3.1.16.
QDPRIMESOURCE.MOD6.4 4.4.1.4 UHBR Rate Pixel Data Packing and Steering (HBR 128b/132b); Please refer to the test description of 4.4.1.4.
QDPRIMESOURCE.MOD6.5 4.4.2.2 UHBR Rate Main Video Stream Format Change Handling (HBR 128b/132b); Please refer to the test description of 4.4.2.2.
QDPRIMESOURCE.MOD6.6 4.4.4.1 Configuring Video and Audio Parameters (HBR 128b/132b); Please refer to the test description of 4.4.4.1.
QDPRIMESOURCE.MOD6.7 4.4.4.2 Audio Stream Header Synchronization (HBR 128b/132b); Please refer to the test description of 4.4.4.2.
QDPRIMESOURCE.MOD6.8 4.4.4.3 Audio Time Stamp Generation (HBR 128b/132b); Please refer to the test description of 4.4.4.3.
QDPRIMESOURCE.MOD6.9 4.4.4.4 Audio INFOFRAME Packet (HBR 128b/132b); Please refer to the test description of 4.4.4.4.
QDPRIMESOURCE.MOD6.10 4.4.4.5 Audio Stream Transmission (HBR 128b/132b); Please refer to the test description of 4.4.4.5.
QDPRIMESOURCE.MOD6.11 4.4.4.6 Audio Start Sequence (HBR 128b/132b); Please refer to the test description of 4.4.4.6.
QDPRIMESOURCE.MOD6.12 4.6.1.3 DSC PPS convert RGB flag verification (HBR 128b/132b); Please refer to the test description of 4.6.1.3.
QDPRIMESOURCE.MOD6.13 4.6.1.4 DSC PPS (YCbCr 4:4:4 convert RGB = 0) flag verification (HBR 128b/132b); Please refer to the test description of 4.6.1.4.
QDPRIMESOURCE.MOD6.14 4.6.1.5 DSC PPS Simple 4:2:2 flag verification (HBR 128b/132b); Please refer to the test description of 4.6.1.5.
QDPRIMESOURCE.MOD6.15 4.6.1.6 DSC PPS Native 4:2:2 flag verification (HBR 128b/132b); Please refer to the test description of 4.6.1.6.
QDPRIMESOURCE.MOD6.16 4.6.1.7 DSC PPS Native 4:2:0 flag verification (HBR 128b/132b); Please refer to the test description of 4.6.1.7.
QDPRIMESOURCE.MOD6.17 4.6.1.8 DSC PPS convert RGB flag verification for DSC Algorithm (HBR 128b/132b); Please refer to the test description of 4.6.1.8.
QDPRIMESOURCE.MOD6.18 4.6.1.9 DSC PPS (YCbCr 4:4:4 convert RGB = 0) flag verification for DSC Algorithm (HBR 128b/132b); Please refer to the test description of 4.6.1.9.
QDPRIMESOURCE.MOD6.19 4.8.1.1 Fixed-Average VTotal Mode Support over the Declared Frame Rate Range (HBR 128b/132b); Please refer to the test description of 4.8.1.1.
QDPRIMESOURCE.MOD6.20 4.8.1.2 Fixed-Average VTotal Mode Duration Increase and Decrease Constraint Value Support (HBR 128b/132b); Please refer to the test description of 4.8.1.2.

Source Mod7 MST Tests:

The module 7 provides MST tests and are intended to provide the building blocks for validating DisplayPort 2.1 MST operation. These tests are in early draft stage and will be updated in future releases.

QDPRIMESOURCE.MOD7.1 MST Source: Reads MSTM_CAP (DPCD 00021h[0]); Verifies if the MST capable source DUT reads the MSTM_CAP register of the downstream device
QDPRIMESOURCE.MOD7.2 MST Source: Sets MSTM_CTRL (DPCD 00111h) register appropriately. Verifies that source DUT sets MSTM_CTRL only when it sees an MST capable branch/sink
QDPRIMESOURCE.MOD7.3 MST Source: Supports SST mode as well Verifies that an MST capable source should shall work well with an SST only sink
QDPRIMESOURCE.MOD7.4 [Under Development] MST Source should be able to perform transition to SST mode

Verifies that Source DUT transitions to SST mode as per the spec.
QDPRIMESOURCE.MOD7.5 [Under Development] MST Source: Clears the Message sequence number, i.e. message transactions are atomic Verify that no two consequetive sideband transactions have same message sequence number. Message sequence number can only repeat in case of a timeout or when retries are exhausted.
QDPRIMESOURCE.MOD7.6 [Under Development] MST Source: Max number of sideband transaction request retires when there is no response for a sideband request Verifies retry behavior when there is no response for a sideband request
QDPRIMESOURCE.MOD7.7[Under Development] MST Source: Max timeout for sideband transaction request timout when there is no response. Verifies timeout behavior when there is no response for a sideband request
QDPRIMESOURCE.MOD7.8 MST Source: Link Address Request upon detecting a Branch (1 level down) Verifies that Source issues Link Address request upon detecting any branch 1 level down; test is repeated for different topologies.
QDPRIMESOURCE.MOD7.9 MST Source: Link Address Request is not a path message and not a broadcast message. Verifies that Link Address Request is a directed message replied only by the targetted device. Repeated for different topologies.
QDPRIMESOURCE.MOD7.10 MST Source: Link Address Request upon detecting a Branch (2 levels down) Verifies that Source issues Link Address request upon detecting any branch 2 levels down
QDPRIMESOURCE.MOD7.11 MST Source: Link Address Request upon detecting a Branch (3 levels down) Verifies that Source issues Link Address request upon detecting any branch 3 levels down
QDPRIMESOURCE.MOD7.12 MST Source: Link Address Request upon detecting a Branch (4 levels down) Verifies that Source issues Link Address request upon detecting any branch 4 levels down; Rrepeated for different topologies.
QDPRIMESOURCE.MOD7.13 MST Source: Link Address Request upon detecting a Branch (5 levels down) Verifies that Source issues Link Address request upon detecting any branch 5 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.14 MST Source: Link Address Request upon detecting a Branch (6 levels down) Verifies that Source issues Link Address request upon detecting any branch 6 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.15 MST Source: Link Address Request upon detecting a Branch (7 levels down) Verifies that Source issues Link Address request upon detecting any branch 7 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.16 MST Source: Enum Path Resources Request upon detecting a Sink (1 level down) Verifies that Source issues Enum Path Resources Request upon detecting any branch 1 level down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.17 MST Source: Enum Path Resources Request is not a path message and not a broadcast message Verifies that Enum Path Resources Request is a directed message replied only by the targetted device; Repeated for different topologies.
QDPRIMESOURCE.MOD7.18 MST Source: Enum Path Resources Request upon detecting a Sink (2 levels down) Verifies that Source issues Enum Path Resources Request upon detecting any branch 2 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.19 MST Source: Enum Path Resources Request upon detecting a Sink (3 levels down) Verifies that Source issues Enum Path Resources request upon detecting any sink 3 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.20 MST Source: Enum Path Resources Request upon detecting a Sink (4 levels down) Verifies that Source issues Enum Path Resources request upon detecting any sink 4 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.21 MST Source: Enum Path Resources Request upon detecting a Sink (5 levels down); ; Verifies that Source issues Enum Path Resources request upon detecting any sink 5 levels down; Repeated for different topologies.

QDPRIMESOURCE.MOD7.22 MST Source: Enum Path Resources Request upon detecting a Sink (6 levels down) Verifies that Source issues Enum Path Resources request upon detecting any sink 6 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.23 MST Source: Enum Path Resources Request upon detecting a Sink (7 levels down) Verifies that Source issues Enum Path Resources request upon detecting any sink 7 levels down; Repeated for different topologies.
QDPRIMESOURCE.MOD7.24 [Under Development] MST Source: Allocate Payload Sequence is initiated for 1 Sink
QDPRIMESOURCE.MOD7.25 [Under Development] MST Source: Clear Payload table request for the first time, after HPD
QDPRIMESOURCE.MOD7.26 [Under Development] MST Source: Clear Payload table is a path message and a broadcast message
QDPRIMESOURCE.MOD7.27 [Under Development] MST Source: VC Payload Table update before Allocate Payload request (1 level down)
QDPRIMESOURCE.MOD7.28 [Under Development] MST Source: ACT Sequence after VC Payload Table update for the first branch
QDPRIMESOURCE.MOD7.29 [Under Development] MST Source: Payload table from TimeSlot 0 for UHBR and TimeSlot 1 for 8b/10b
QDPRIMESOURCE.MOD7.30 [Under Development] MST Source: ACT sequence is not inserted in:
QDPRIMESOURCE.MOD7.31 [Under Development] [Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (1 level down)
QDPRIMESOURCE.MOD7.32 [Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (2 level down)
QDPRIMESOURCE.MOD7.33[Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (3 level down)
QDPRIMESOURCE.MOD7.34 [Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (4 level down)
QDPRIMESOURCE.MOD7.35 [Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (5 level down)
QDPRIMESOURCE.MOD7.36 [Under Development] [Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (6 level down)
QDPRIMESOURCE.MOD7.37 [Under Development] MST Source: Allocate Payload Request with 0 PBN upon NACK (7 level down)
QDPRIMESOURCE.MOD7.38 [Under Development] [Under Development] MST Source: Allocate Payload sequence is initiated when 2 sinks are present
QDPRIMESOURCE.MOD7.39 [Under Development] MST Source: Allocate Payload sequence is initiated when 3 sinks are present
QDPRIMESOURCE.MOD7.40 [Under Development] MST Source: Allocate Payload sequence is initiated when 4 sinks are present
QDPRIMESOURCE.MOD7.41 [Under Development] MST Source: Allocate Payload sequence is initiated when 5 sinks are present
QDPRIMESOURCE.MOD7.42 [Under Development] MST Source: Allocate Payload sequence is initiated when 6 sinks are present
QDPRIMESOURCE.MOD7.43 [Under Development] MST Source: Allocate Payload sequence is initiated when 7 sinks are present
QDPRIMESOURCE.MOD7.44 [Under Development] MST Source: Allocate Payload sequence is initiated when 8 sinks are present
QDPRIMESOURCE.MOD7.45 [Under Development] MST Source: Allocate Payload sequence is initiated when 9 sinks are present
QDPRIMESOURCE.MOD7.46 [Under Development] MST Source: Allocate Payload sequence is initiated when 10 sinks are present
QDPRIMESOURCE.MOD7.47 [Under Development] MST Source: Allocate Payload sequence is initiated when 11 sinks are present
QDPRIMESOURCE.MOD7.48 [Under Development]

[Under Development] MST Source: Allocate Payload sequence is initiated when 12 sinks are present
QDPRIMESOURCE.MOD7.49 [Under Development] MST Source: Allocate Payload sequence is initiated when 13 sinks are present
QDPRIMESOURCE.MOD7.50 [Under Development] MST Source: Allocate Payload sequence is initiated when 14 sinks are present
QDPRIMESOURCE.MOD7.51 [Under Development] MST Source: Allocate Payload sequence is initiated when 15 sinks are present
QDPRIMESOURCE.MOD7.52 [Under Development] MST Source: Allocate Payload sequence is initiated when 16 sinks are present
QDPRIMESOURCE.MOD7.53 [Under Development] MST Source: CSN ACK response
QDPRIMESOURCE.MOD7.54 [Under Development] MST Source: RSN ACK response
QDPRIMESOURCE.MOD7.55 [Under Development] MST Source: Delete Payload upon CSN with a sink device disconnect (2 levels down)
QDPRIMESOURCE.MOD7.56 [Under Development] MST Source: Allocate Payload request (PBN = 0) upon CSN with sink device disconnect (2 level down)
QDPRIMESOURCE.MOD7.57 [Under Development] MST Source: Delete Payload upon CSN with a sink device disconnect (3 levels down)
QDPRIMESOURCE.MOD7.58 [Under Development] MST Source: Allocate Payload request (PBN = 0) upon CSN with sink device disconnect (3 level down)
QDPRIMESOURCE.MOD7.59 [Under Development] MST Source: Delete Payload upon CSN with a sink device disconnect (4 levels down)
QDPRIMESOURCE.MOD7.60 [Under Development] MST Source: Allocate Payload request (PBN = 0) upon CSN with sink device disconnect (4 level down)
QDPRIMESOURCE.MOD7.61 [Under Development] MST Source: Delete Payload upon CSN with a sink device disconnect (5 levels down)
QDPRIMESOURCE.MOD7.62 [Under Development] MST Source: Allocate Payload request (PBN = 0) upon CSN with sink device disconnect (5 level down)
QDPRIMESOURCE.MOD7.63 [Under Development] MST Source: Delete Payload upon CSN with a sink device disconnect (6 levels down)
QDPRIMESOURCE.MOD7.64 [Under Development] MST Source: Allocate Payload request (PBN = 0) upon CSN with sink device disconnect (6 level down)
QDPRIMESOURCE.MOD7.65 [Under Development] MST Source: Delete Payload upon CSN with a sink device disconnect (7 levels down)
QDPRIMESOURCE.MOD7.66 [Under Development] MST Source: Allocate Payload request (PBN = 0) upon CSN with sink device disconnect (7 level down)
QDPRIMESOURCE.MOD7.67 [Under Development] MST Source: Delete Payloads or Clear Payload Table upon CSN with a branch device disconnect with multiple sinks (2 levels down)
QDPRIMESOURCE.MOD7.68 [Under Development] MST Source: Delete Payloads & allocate payload sequence(s)(PBN = 0) upon CSN with a branch device disconnect with multiple sinks (2 levels down)
QDPRIMESOURCE.MOD7.69 [Under Development] MST Source: Delete Payloads & allocate payload sequence(s)(PBN = 0) upon CSN with a branch device disconnect with multiple sinks (3 levels down)
QDPRIMESOURCE.MOD7.70 [Under Development] MST Source: Delete Payloads & allocate payload sequence(s)(PBN = 0) upon CSN with a branch device disconnect with multiple sinks (4 levels down)
QDPRIMESOURCE.MOD7.71 [Under Development] MST Source: Delete Payloads & allocate payload sequence(s)(PBN = 0) upon CSN with a branch device disconnect with multiple sinks (5 levels down)
QDPRIMESOURCE.MOD7.72 [Under Development] MST Source: Delete Payloads & allocate payload sequence(s)(PBN = 0) upon CSN with a branch device disconnect with multiple sinks (6 levels down)
QDPRIMESOURCE.MOD7.73 [Under Development]

MST Source: Delete Payloads & allocate payload sequence(s)(PBN = 0) upon CSN with a branch device disconnect with multiple sinks (7 levels down)
QDPRIMESOURCE.MOD7.74[Under Development] MST Source: Allocate Payload with reduced PBN upon RSN with reduced b/w (2 levels down)
QDPRIMESOURCE.MOD7.75 [Under Development] MST Source: Reduced Throttled_VCP_Size before Allocate Payload with reduced PBN
QDPRIMESOURCE.MOD7.76 [Under Development] MST Source: Allocate Payload with reduced PBN upon RSN with reduced b/w (3 levels down)
QDPRIMESOURCE.MOD7.77 [Under Development] MST Source: Allocate Payload with reduced PBN upon RSN with reduced b/w (4 levels down)
QDPRIMESOURCE.MOD7.78 [Under Development] MST Source: Allocate Payload with reduced PBN upon RSN with reduced b/w (5 levels down)
QDPRIMESOURCE.MOD7.79 [Under Development] MST Source: Allocate Payload with reduced PBN upon RSN with reduced b/w (6 levels down)
QDPRIMESOURCE.MOD7.80 [Under Development] MST Source: Allocate Payload with reduced PBN upon RSN with reduced b/w (7 levels down)
QDPRIMESOURCE.MOD7.81 [Under Development] MST Source: Waits for ACK response from Allocate payload sequence before
QDPRIMESOURCE.MOD7.82 [Under Development] MST Source: Source DUT inserts and transmits VCPF control link sequence when no stream symbols are present
QDPRIMESOURCE.MOD7.83 [Under Development] MST Source: Allocate Payload ACK before Increasing Throttled_VCP_Size upon RSN with increased PBN
QDPRIMESOURCE.MOD7.84 [Under Development] MST Source: FEC_PARITY and FEC_PM link symbols insertion when DFP has FEC enabled
QDPRIMESOURCE.MOD7.85 [Under Development] MST Source: SDP Data rate should be greater than Peak Data rate of stream packed into SDPs
QDPRIMESOURCE.MOD7.86 [Under Development] MST Source: Even distribution of additional data across timeslots for FEC_PARITY_PH and FEC_OM link symbols
QDPRIMESOURCE.MOD7.87 [Under Development] MST Source: Average of Stream Symbol time slots per MTP = TARGET_Average_StreamSymbolTimeSlotsPerMTP
QDPRIMESOURCE.MOD7.88 [Under Development] MST Source: Audio_Stream SDP transmission rate <= HBLNK_ML_SYM_CYC_CNT per HBlank Period
QDPRIMESOURCE.MOD7.89 [Under Development] MST Source: 2^16 symbol interval of SR control link symbols after LT
QDPRIMESOURCE.MOD7.90 [Under Development] MST Source: Data symbols transported in the time slots for ECFs are not encrypted
QDPRIMESOURCE.MOD7.91 [Under Development] MST Source: When FEC enabled, 2.4% transport overhead for PBN calculation of AV stream
QDPRIMESOURCE.MOD7.92 [Under Development] MST Source: FEC is not enabled until LT is completed
QDPRIMESOURCE.MOD7.93 [Under Development] MST Source: Video Detected on x Sinks

Source Mod8 Capture Tests:

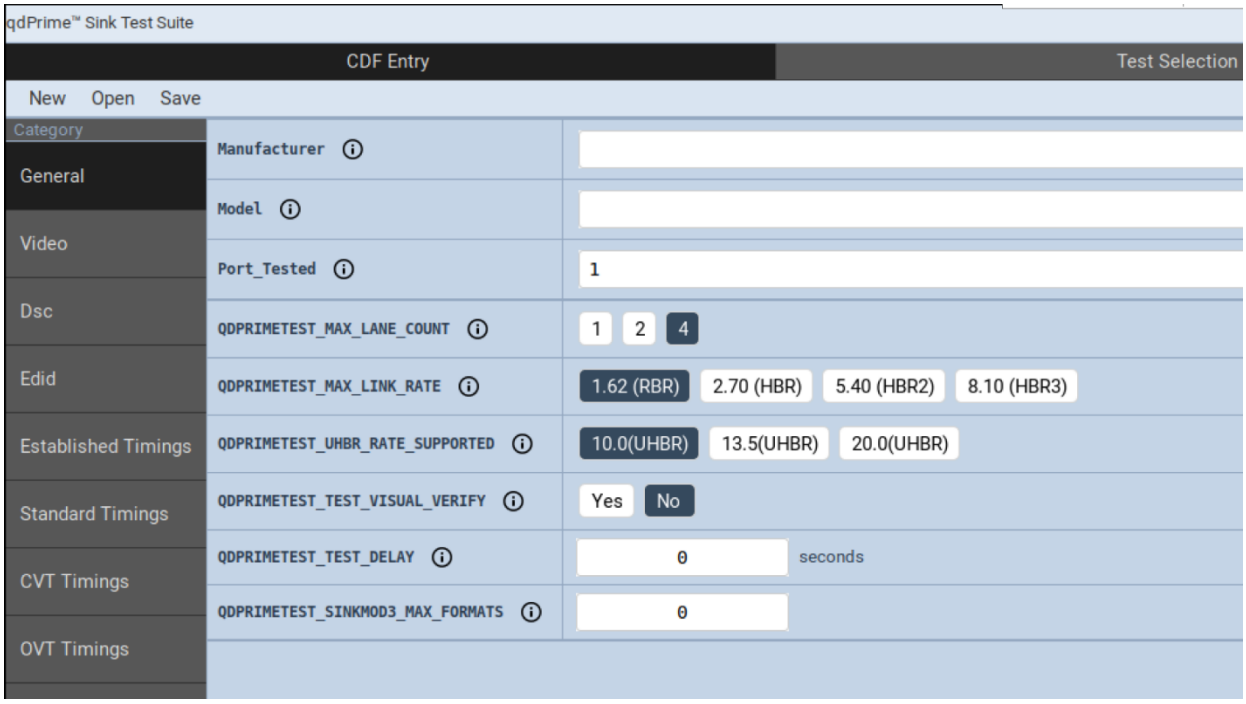
The module 8 provides miscellaneous proprietary tests with additional test cases expected in future releases.

QDPRIMESOURCE.MOD8.1 Minimum Horizontal Blank Time validation This test will capture the data for 5 sec and generate a report for each HBlank time.

4 Creating CDF Files (Sink DUT)

Upon opening qdPrime Sink Test Suite, the first window displayed is the **CDF Entry** tab, as shown (below). There are several tabs on the left that are used to input specific parameters. Sink test operation is similar to source test

in that the entries in the CDF will bypass the EDID and DPCD settings on the sink and instead direct the sink using the parameters defined in these tabs. During the test execution stage, qdPrime will deterministically request these parameters to ensure the desired test conditions are covered.
Important Note: It is essential to save the CDF file after making any modifications.

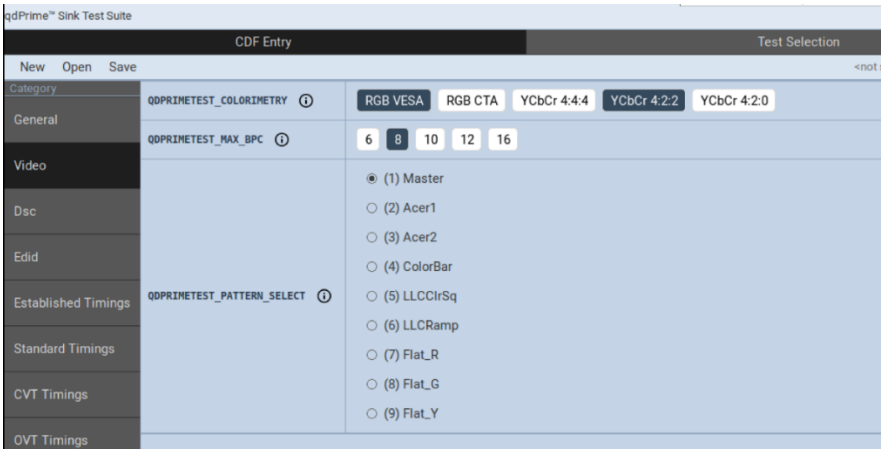


To configure qdPrime for Sink Tests, it is necessary to input operating parameters summarized below:

- General Tab – Basic operational parameters for the link. When multiple items are selected, this instructs qdPrime to generate a separate iteration using each parameter for all selected timings.
    - Max_Lane Count: Selecting 4 lanes will cause all tests to repeat at 1, 2, and 4 lanes.
    - Max_Link_Rate: Selecting HBR3 rate will cause all tests to repeat at HBR3, HBR2, HBR & RBR rates.
    - UHBR_Rate_Supported: Defines the UHBR rates that will be tested. Users should select Each lane rate that they intend to test.

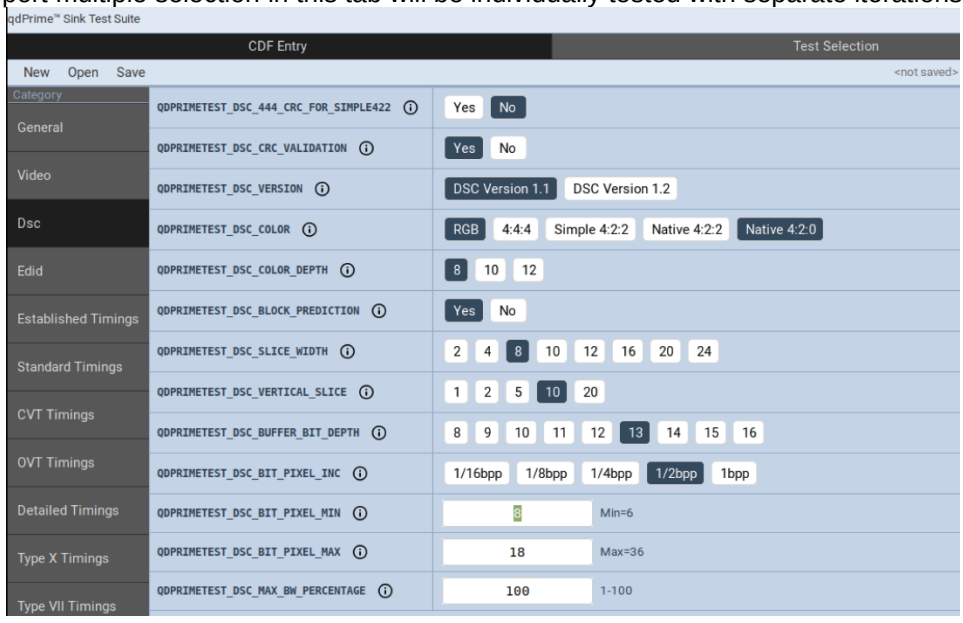
- **TX_VIS_VAL**: Enables qdPrime to prompt the user to perform visual verification of the received images after each test iteration. See the section **Visual Verification Step**
    - QDPRIMETEST_CACHE_CRC: allows user to specify to always save the CRC in the database or not. Setting it to FALSE, will *prompt* the user every time to save the CRC; setting it to TRUE, will automatically save the CRC by default in the database without prompting user.
    - Test_Delay: Allows user to specify an additional delay interval between tests (helpful for early development devices).
    - SinkMod3_Max_Formats Allows user to specify the maximum number of formats the Tester as Source will generate for each timing (allows constraining the number of test iterations for Sink DUTs).

- Video Tab - Video specific parameters including color space and bits-per-component are entered here. When multiple items are selected, qdPrime will generate a separate test iteration using each parameter for all of the defined timings.



  - DP20_Colorimetry – Defines the color space that will be tested for the Source DUT
  - DP20_Bpc – Defines the bits-per-color for UHBR modes
  - Pattern_Select – Defines which pattern the Tester (as source) will transmit to the Rx port. Visual verification and updating the internal qdPrime CRC database may be necessary if users change patterns between test runs.

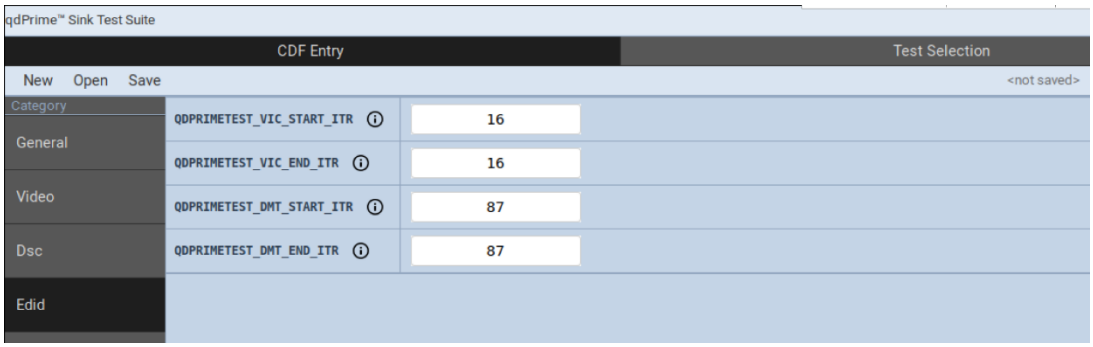
DSC tab– If the sink DUT supports DSC, the capabilities entered here communicate to qdPrime which parameters the Tester (as source) will use when generating compressed images. When running the DSC specific sink tests, qdPrime will check these specific parameters bypassing the EDID and DPCD entries. Items that support multiple selection in this tab will be individually tested with separate iterations.



  - DSC_444_CRC_FOR_SIMPLE422 – Specifies if (Sink DUT) supports 4:4:4 CRC when operating in 4:2:2 DSC mode

- DSC_CRC_VALIDATION – Specifies if CRC values from the cached database should be used by qdPrime to perform image verification.
- DSC_VERSION – Specifies which DSC version (Sink DUT) uses and will be tested.
- DSC_COLOR – Specifies which DSC Color Space (Sink DUT) supports and will be tested.
- DSC_COLOR_DEPTH – Specifies which DSC Color Depth will be tested.
- DSC_BLOCK_PREDICTION – Specifies if DSC Block prediction (Sink DUT) supports and will be tested.
- DSC_SLICE_WIDTH – Specifies the number of DSC slice count that will be tested.
- DSC_VERTICAL_SLICE – Specifies DSC slice width sink supports and will be tested.
- DSC_BUFFER_BIT_DEPTH – Specifies the buffer bit-depth that will be tested by qdPrime
- DSC_BIT_PIXEL_INC – Specifies the bit pixel increment that will be tested by qdPrime
- DSC_BIT_PIXEL_MIN – Specifies the bit per pixel min that will be tested by qdPrime
- DSC_BIT_PIXEL_MAX – Specifies the bit per pixel max that will be tested by qdPrime
- DSC_MAX_BW_PERCENTAGE – Specifies the max bandwidth percent that can be allocated when tester is configuring video stream for higher resolution test cases.

- EDID tab - Allows the default range for VIC and DMT values to be defined for range interator test runs. Experiment by selecting only a small range to assess the execution time needed for larger test runs.



- VIC_Start_ITR – specifies starting VIC for ‘range iterator’ test cases
- VIC_End_ITR – specifies ending VIC for ‘range iterator’ test cases
- DMT_Start_ITR – specifies starting DMT for ‘range iterator’ test cases
- DMT_End_ITR – specifies ending DMT for ‘range iterator’ test cases

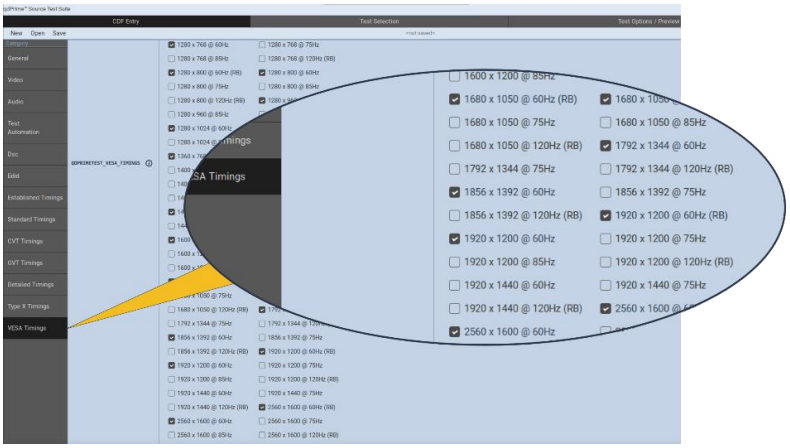
Entering Selections in the Timing Tabs:
The timings tabs provide a menu based selection of timing specific parameters to be tested. Users can select standard timings common to VESA or GTF standards. The Detailed tab allows users to create “custom” timings that can alternately be selected for testing in the Test Selection menu. It is only necessary to enter parameters in the timing tabs if users intend to test these specific timings in the Test Selection menu.

Established timings includes all VESA timings that can be advertised and supported by the M42d/de platform.

Established Timings	Standard timings defines all Generalized Timing Formula (GTF) timings that can be advertised and supported by the M42d/de platform.
Standard Timings	Coordinated Video Timings (CVT) for the latest VESA specification that can be advertised and supported by the M42d/de platform.
CVT Timings	Optimized Video Timings (OVT) refers to the CTA-861 standard for Extended and supported by the M42d/de platform.
OVT Timings	Detailed Timings provides a menu based interface to create custom timings including active lines and pixels. All detailed timings must adhere to the 18-byte Detailed Timing Descriptors (DTD) and must be a “valid DTD format”.
Detailed Timings	Type X provides a menu based interface to create Extended Block Type Tag “X”. It must be a “valid extended block format” to be verified with qdPrime.
Type X Timings	Type VII provides a menu based interface to create Extended Block Type Tag “VII”. It must be a valid “Type VII block format” to be verified with qdPrime.
Type VII Timings	VESA timings includes the most common timings defined by VESA that can be generated by the M42d/de platform.
VESA Timings	

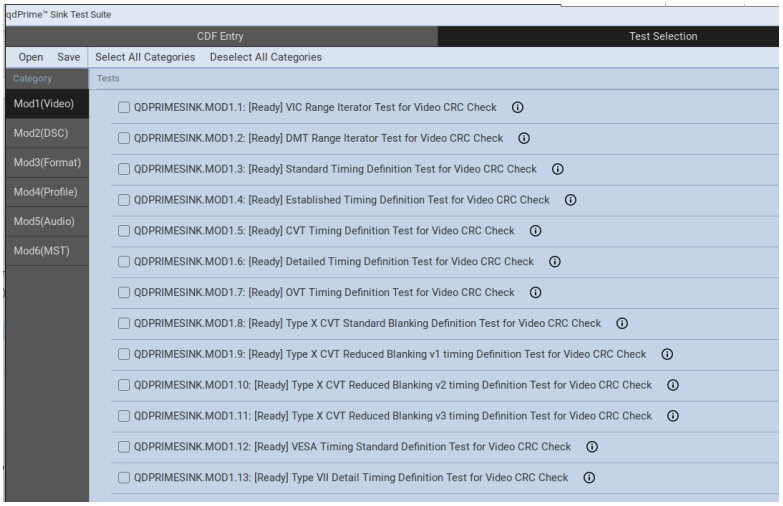
When selecting specific timings or adding custom timings, these parameters will instruct qdPrime to create a series of tests that systematically run each of the selected timings in combination with other variables (link width, lane rate, bpc, etc…). There can be more than 100 variations needed to cover a single timing selection (below). Experiment by selecting only a single timing entry to assess the execution time needed for larger test runs.

Important Note: It is essential to save the CDF file after making any modifications.



5 Test Selection Menu (Sink DUTs)

The Test Selection menu for sinks is broadly organized around tests that verify specific formats defined in the CDF. The modules 1 through 3 are the primary modules that should be used to deterministically test the different timings. They specifically bypass the EDID/DPCD on the DUT to deterministically test all the VIC, DMT, CVT, OVT, VESA, or Custom timings entered in the Sink CDF. The module 2 performs similar tests as module 1, but specifically with “DSC” enabled.



The modules 5 through 7 are special test cases that provide specific test conditions described below: The “Profile” tests (Mod4) are special tests designed to use the current installed EDID and DPCD entries on the Sink DUT. These tests do not use the CDF selections but rather default to the actual lane-rates, link widths, colorimtry, etc… defined in the sink’s EDID/DPCD. To control which Mod4 tests are performed may require installing a custom EDID and DPCD profile on the sink UUT. The “Audio” (Mod5) tests provide proprietary tests that use entries in the audio CDF tab for validating audio operation. The “MST” tests (Mod6) are intended to provide the building blocks for validating DisplayPort 2.1 MST operation for branch devices. The “MST” tests (Mod7) are intended to provide the building blocks for validating MST operation for sink devices.

Sink Mod1 Video Tests:

The module 1 are the primary tests designed to verify that Sink DUTs can operate with a source that deterministically tests all predefined VIC, DMT, CVT, OVT, VESA, or Custom timings entered in the Sink CDF (while remaining within the allowed bandwidth).

QDPRIMESINK.MOD1.1 VIC Range Iterator Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for the whole range of VIC IDs selected from the CDF.
QDPRIMESINK.MOD1.2 DMT Range Iterator Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for the whole range of DMT IDs selected from the CDF.
QDPRIMESINK.MOD1.3 Standard Timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for standard timing definitions specified in the CDF.
QDPRIMESINK.MOD1.4 Established Timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for established timings definitions specified in the CDF.
QDPRIMESINK.MOD1.5 CVT Timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for CVT timing definitions specified in the CDF.
QDPRIMESINK.MOD1.6 Detailed Timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for detailed timing definitions specified in the CDF.
QDPRIMESINK.MOD1.7 OVT Timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for OVT timing definitions specified in the CDF.
QDPRIMESINK.MOD1.8 Type X CVT Standard Blanking Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for Type X CVT Standard Blanking timing definitions specified in the CDF.
QDPRIMESINK.MOD1.9 Type X CVT Reduced Blanking v1 timing Definition Test for Video CRC Check

Verifies the image generated by matching the sink's CRC with the source CRC for Type X CVT Reduced Blanking v1 timing definitions specified in the CDF.
QDPRIMESINK.MOD1.10 Type X CVT Reduced Blanking v2 timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for Type X CVT
QDPRIMESINK.MOD1.11 Type X CVT Reduced Blanking v3 timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for Type X
QDPRIMESINK.MOD1.12 VESA Timing Standard Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for VESA timing definitions specified in the CDF.
QDPRIMESINK.MOD1.13 Type VII Detail Timing Definition Test for Video CRC Check Verifies the image generated by matching the sink's CRC with the source CRC for Type VII detailed timing definitions specified in the CDF.

Sink Mod2 DSC Tests:

The module 2 performs similar tests as module 1 with DSC enabled. This includes verifying that Sink DUTs can operate with a source that deterministically tests all predefined VIC, DMT, CVT, OVT, VESA, or Custom timings entered in the Sink CDF (while remaining within the allowed bandwidth).

QDPRIMESINK.MOD2.1 VIC Range Iterator Test for DSC Video CRC Check Verifies the DSC image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for the whole range of VIC IDs selected from the CDF.
QDPRIMESINK.MOD2.2 DMT Range Iterator Test for DSC Video CRC Check Verifies the DSC image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for the whole range of DMT IDs selected from the CDF.
QDPRIMESINK.MOD2.3 Standard Timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC standard timing definitions specified in the CDF.
QDPRIMESINK.MOD2.4 Established Timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC established timing definitions specified in the CDF.
QDPRIMESINK.MOD2.5 CVT Timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC CVT timing definitions specified in the CDF
QDPRIMESINK.MOD2.6 Detailed Timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC detailed timing definitions specified in the CDF
QDPRIMESINK.MOD2.7 OVT Timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC OVT timing definitions specified in the CDF.
QDPRIMESINK.MOD2.8 Type X CVT Standard Blanking Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC Type X CVT Standard Blanking timing definitions specified in the CDF.
QDPRIMESINK.MOD2.9 Type X CVT Reduced Blanking v1 timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC Type X CVT Reduced Blanking v1 timing definitions specified in the CDF.
QDPRIMESINK.MOD2.10 Type X CVT Reduced Blanking v2 timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC Type X CVT Reduced Blanking v2 timing definitions specified in the CDF.
QDPRIMESINK.MOD2.11 Type X CVT Reduced Blanking v3 timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC Type X CVT Reduced Blanking v3 timing definitions specified in the CDF.
QDPRIMESINK.MOD2.12 VESA Timing Standard Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for DSC VESA timing definitions specified in the CDF.
QDPRIMESINK.MOD2.13 Type VII Detail Timing Definition Test for DSC Video CRC Check Verifies the image generated by either asking user to visually verify or matching the sink's CRC with the CRC database for Type VII DSC detailed timing definitions specified in the CDF.

Sink Mod3 Format Tests:

The module 3 verifies that Sink DUTs can operate for all predefined VIC, DMT, DMR, CVT, CR2, and OVT timings at all BPC and colorimetry parameters (while remaining within the maximum defined bandwidth).

QDPRIMESINK.MOD3.1 All predefined formats iteration Verifies the image generated by matching the sink's CRC with the source CRC for all the formats in the generator's library.
QDPRIMESINK.MOD3.3 Format iteration for all the OVT formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of OVT category formats in the generator's library.
QDPRIMESINK.MOD3.4 Format iteration for all the APP formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of APP category formats in the generator's library.
QDPRIMESINK.MOD3.5 Format iteration for all the C2R formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of C2R category formats in the generator's library.
QDPRIMESINK.MOD3.6 Format iteration for all the CVR formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of CVR category formats in the generator's library.
QDPRIMESINK.MOD3.7 Format iteration for all the CVT formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of CVT category formats in the generator's library.
QDPRIMESINK.MOD3.8 Format iteration for all the DMR formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of DMR category formats in the generator's library.
QDPRIMESINK.MOD3.9 Format iteration for all the DMT formats Verifies the image generated by matching the sink's CRC with the source CRC for all the formats of DMT category formats in the generator's library.

Sink Mod4 Profile Tests:

The module 4 tests are designed to use the current installed EDID and DPCD entries on the Sink DUT. These tests do not use the CDF selections. Separate tests are provided that use CTA-861 EDID or Native Display-ID as well as with or without DSC enabled. For the “MST” specific tests, the Tester (as source) will test formats defined in the “profile” using MST VC mode. Several early development tests for validating Panel Replay, DSC Toggling, and Adaptive Sync will be added in future releases.

QDPRIMESINK.MOD4.1 Video Test (SST Mode) for all the formats supported in EDID. Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's EDID in single stream transport(SST) mode.
QDPRIMESINK.MOD4.2 DSC Test (SST Mode) for all the formats supported in EDID. Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's EDID in single stream transport(SST) mode.
QDPRIMESINK.MOD4.3 Video Test (SST Mode) for all the formats supported in NDID. Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's NDID in single stream transport(SST) mode.
QDPRIMESINK.MOD4.4 DSC Test (SST Mode) for all the formats supported in NDID. Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's NDID in single stream transport(SST) mode.
QDPRIMESINK.MOD4.6 DSC Test (MST Mode) for all the formats supported in EDID. Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's EDID in multi stream transport(MST) mode.
QDPRIMESINK.MOD4.7 Video Test (MST Mode) for all the formats supported in NDID. Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's NDID in multi stream transport(MST) mode.
QDPRIMESINK.MOD4.8 DSC Test (MST Mode) for all the formats supported in NDID.

Verifies the image generated by matching the sink's CRC with the source CRC for all the formats from the sink's NDID in multi stream transport(MST) mode.
QDPRIMESINK.MOD4.9 Toggle between multiple BPP DSCs and non-DSC image of same format video test. Verifies that the sink device can switch between different DSC BPPs and non-DSC image of the same format without a glitch on sink.
QDPRIMESINK.MOD4.10 [Under development] Validate Panel Reply state transition. Verifies that the sink device can switch to panel replay states.
QDPRIMESINK.MOD4.11 [Under development] Panel Reply Full Frame Video test. Verifies that sink device can display Videos with full frame update.
QDPRIMESINK.MOD4.12 [Under development] Panel Reply Full Frame DSC test. Verifies that sink device can display Videos with full frame DSC update.
QDPRIMESINK.MOD4.13 [Under development] Panel Reply Selective Frame Update Video test. Verifies that sink device can display Videos with selective frame update.
QDPRIMESINK.MOD4.14[Under development] Adaptive Sync square pattern tests. Verifies that sink device can handle at given refresh rate min to max for both EDID and NDID.
QDPRIMESINK.MOD4.15 [Under development] Adaptive Sync increment decrement frame rate tests. Verifies that sink device can increment and decrement of refresh rates for both EDID and NDID.
QDPRIMESINK.MOD4.16 Audio test using min and max sink supported capability. Verifies the audio generated by asking the user to hear the audio stream from the SINK DUT.

Sink Mod5 Audio Tests:

The module 5 are audio specific tests designed to verify that Sink DUTs can correctly extract and process audio using the parameters entered in the CDF.

QDPRIMESINK.MOD5.1 Audio stream channel test (single channel only). Verifies that the sink device can handle audio at every selected channel for the maximum bits/sample and maximum sample rate selected in the CDF.
QDPRIMESINK.MOD5.2 Audio stream sample rate test (single channel only). Verifies that the sink device can handle audio at every sample rate for the maximum bits / sample and maximum channel count selected in the CDF. (require the user to hear the audio and respond to complete the test.)
QDPRIMESINK.MOD5.3 Audio stream Bits per Sample Test (single channel only). Verifies that the sink device can handle audio at every bits / sample for the maximum sample rate and maximum channel count selected in the CDF.
QDPRIMESINK.MOD5.4 Audio stream increase/decrease level (dB) (single channel only). Verifies that the sink device can handle audio streams in a range of levels (dB) for maximum sample rate, maximum channel count, and maximum bits / sample as selected in the CDF and default values for frequency.
QDPRIMESINK.MOD5.5 Audio stream increase/decrease frequency (single channel only). Verifies that the sink device can handle audio streams on a range of frequencies (Hz) for maximum sample rate, maximum channel count, maximum bits / sample as selected in the CDF, and default values for level (dB).

Branch Mod6 MST Tests:

Mod6 provides MST tests and are intended to serve as building blocks for validating DisplayPort 2.1 MST operation on branch devices. These tests are in early draft stage and will be updated in future releases.

QDPRIMESINK.MOD6.1 [Under Development] MST Branch: MSTM_CAP.
QDPRIMESINK.MOD6.2 [Under Development] MST Branch: Verify Update of MST control fields based on DFP Capabilities.
QDPRIMESINK.MOD6.3 [Under Development] MST Branch: SINK_COUNT, DOWNSTREAM_PORT_STATUS_CHANGED update verification upon Sink Plug in.
QDPRIMESINK.MOD6.4 [Under Development] MST Branch: SINK_COUNT, DOWNSTREAM_PORT_STATUS_CHANGED update verification upon Sink Plug out.
QDPRIMESINK.MOD6.5 [Under Development]

MST Branch: SINK_COUNT verification.
QDPRIMESINK.MOD6.6 [Under Development] MST Branch: Link Trains DFP to Maximum link bandwidth.
QDPRIMESINK.MOD6.7 [Under Development] MST Branch: Set MST_EN in MSTM_CTRL before Link Training DFP.
QDPRIMESINK.MOD6.8 [Under Development] MST Branch: ACK Response to LINK_ADDRESS Request.
QDPRIMESINK.MOD6.9 [Under Development] MST Branch: At least one input and output port present.
QDPRIMESINK.MOD6.10 [Under Development] MST Branch: Physical Ports' number range from 0x0 to 0x7.
QDPRIMESINK.MOD6.11 [Under Development] MST Branch: Logical Ports' number range from 0x8 to 0xF.
QDPRIMESINK.MOD6.12 [Under Development] MST Branch: Input Port numbers less than output port numbers.
QDPRIMESINK.MOD6.13 [Under Development] MST Branch: LINK_ADDRESS ACK Response is appropriate as per the connected DFP.
QDPRIMESINK.MOD6.14 [Under Development] MST Branch: Successful Forward of LINK_ADDRESS Request.
QDPRIMESINK.MOD6.15 [Under Development] MST Branch: Successful Forward of LINK_ADDRESS Response.
QDPRIMESINK.MOD6.16 [Under Development] MST Branch: Successful Response to ENUM_PATH_RESOURCES Request.
QDPRIMESINK.MOD6.17 [Under Development] MST Branch: Successful Forward of ENUM_PATH_RESOURCES Request.
QDPRIMESINK.MOD6.18 [Under Development] MST Branch: Successful Forward of ENUM_PATH_RESOURCES Response without modification.
QDPRIMESINK.MOD6.19 [Under Development] MST Branch: Successful Forward of ENUM_PATH_RESOURCES Response with modification.
QDPRIMESINK.MOD6.20 [Under Development] MST Branch: Successful Link Training of DFP before forwarding ENUM_PATH_RESOURCES Request.
QDPRIMESINK.MOD6.21 [Under Development] MST Branch: Responds with correct values w.r.t DFP Sink for ENUM_PATH_RESOURCES Request.
QDPRIMESINK.MOD6.22 [Under Development] MST Branch: Upon hot plug, verify checking DFP for replies to AUX transactions before issue of CSN.
QDPRIMESINK.MOD6.23 [Under Development] MST Branch: Upon hot plug, if DFP does not reply to AUX, verify periodic AUX transactions by Branch DUT.
QDPRIMESINK.MOD6.24 [Under Development] MST Branch: Successful CSN detection upon hot plug.
QDPRIMESINK.MOD6.25 [Under Development] MST Branch: Successful CSN detection upon hot unplug.
QDPRIMESINK.MOD6.26 [Under Development] MST Branch: Upon hot unplug and plug, 2nd CSN only after 1st CSN Ack or 2 seconds, followed by ALLOCATE_PAYLOAD with PBN 0.
QDPRIMESINK.MOD6.27 [Under Development] MST Branch: No CSN upon DFP HPD input level change.
QDPRIMESINK.MOD6.28 [Under Development] MST Branch: On event, removal of unallocated time slots before VC payloads.
QDPRIMESINK.MOD6.29 [Under Development] MST Branch: Upon verification of successful handling of ACT, waits 16 MTPs after new VC payload is established.
QDPRIMESINK.MOD6.30 [Under Development] MST Branch: When ACT sequence received without changes to DPCD 001C0h, no update in payload table.
QDPRIMESINK.MOD6.31 [Under Development] MST Branch: Capable of reassembling nested SDP.
QDPRIMESINK.MOD6.32 [Under Development] MST Branch: SDP splitting supported in SST mode.
QDPRIMESINK.MOD6.33 [Under Development]

MST Branch: SDP splitting supported in MST mode.
QDPRIMESINK.MOD6.34 [Under Development] MST Branch: SDP data rate is greater than peak data rate of stream packed into SDPs.
QDPRIMESINK.MOD6.35 [Under Development] MST Branch: Capable of HBlank reduction sets HBLANK_REDUCTION_CAPABLE in RECEIVE_PORT0_CAP registers.
QDPRIMESINK.MOD6.36 [Under Development] MST Branch test
QDPRIMESINK.MOD6.37 [Under Development] MST Branch test
QDPRIMESINK.MOD6.38 [Under Development] MST Branch test
QDPRIMESINK.MOD6.39 [Under Development] MST Branch test
QDPRIMESINK.MOD6.40 [Under Development] MST Branch test
QDPRIMESINK.MOD6.41 [Under Development] MST Branch test
QDPRIMESINK.MOD6.42 [Under Development] MST Branch test
QDPRIMESINK.MOD6.43 [Under Development] MST Branch test
QDPRIMESINK.MOD6.44 [Under Development] MST Branch test
QDPRIMESINK.MOD6.45 [Under Development] MST Branch test
QDPRIMESINK.MOD6.46 [Under Development] MST Branch test
QDPRIMESINK.MOD6.47 [Under Development] MST Branch test
QDPRIMESINK.MOD6.48 [Under Development] MST Branch test
QDPRIMESINK.MOD6.49 [Under Development] MST Branch test
QDPRIMESINK.MOD6.50 [Under Development] MST Branch test

Sink Mod7 MST Tests:

Mod7 provides MST tests and are intended to serve as building blocks for validating DisplayPort 2.1 MST operation on sink devices. These tests are in early draft stage and will be updated in future releases.

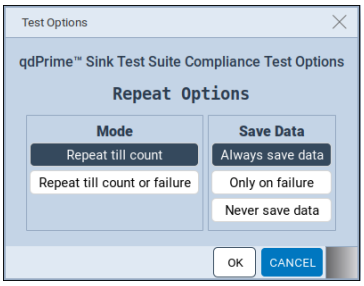
QDPRIMESINK.MOD7.1 [Under Development] MST Sink: SST Mode support
QDPRIMESINK.MOD7.2 [Under Development] MST Sink: DFP port present = 0
QDPRIMESINK.MOD7.3 [Under Development] MST Sink: MST_CAP (DPCD 00021h [0]) = 1
QDPRIMESINK.MOD7.4 [Under Development] MST Sink: Continuous implementation of phase error correction from loss of 4-symbol sequence phase lock
QDPRIMESINK.MOD7.5 [Under Development] MST Sink: Able to handle 8b/10b MST HBLANK ML symbol cycle count variation
QDPRIMESINK.MOD7.6 [Under Development] MST Sink: DUT shall wait for 4 consecutive SR control link symbol with 2^16 time slot intervals before switching to an SR control link symbol location that is different from the original location
QDPRIMESINK.MOD7.7 [Under Development] MST Sink: Verify SR interval's consistency before using the SR control link symbol as the link timing reference

QDPRIMESINK.MOD7.8 [Under Development] MST Sink: Disregard the appearance of SR control link symbol at an unexpected time slot
QDPRIMESINK.MOD7.9 [Under Development] MST Sink: Always use 4 symbol sequence redundancy to detect trigger control sequence
QDPRIMESINK.MOD7.10 [Under Development] MST Sink: Configure to SST
QDPRIMESINK.MOD7.11 [Under Development] MST Sink: Transition to SST
QDPRIMESINK.MOD7.12 [Under Development] MST Sink: Disable the DPRX UFP's Main Link when SET_POWER_STATE = 010b (DPCD 00600h [2:0])
QDPRIMESINK.MOD7.13 [Under Development] MST Sink: VC Payload Mapping Table is intact when switched to low power mode.
QDPRIMESINK.MOD7.14 [Under Development] MST Sink: Clear the VC Payload table upon Clear Payload ID Table.
QDPRIMESINK.MOD7.15 [Under Development] MST Sink: Ability to regenerate stream clock and audio clock without Mvid/Nvid and Maud/Naud values
QDPRIMESINK.MOD7.16 [Under Development] MST Sink: Ability to handle interleaved messages.
QDPRIMESINK.MOD7.17 [Under Development] MST Sink test
QDPRIMESINK.MOD7.18 [Under Development] MST Sink test
QDPRIMESINK.MOD7.19 [Under Development] MST Sink test
QDPRIMESINK.MOD7.20 [Under Development] MST Sink test
QDPRIMESINK.MOD7.21 [Under Development] MST Sink test
QDPRIMESINK.MOD7.22 [Under Development] MST Sink test
QDPRIMESINK.MOD7.23 [Under Development] MST Sink test
QDPRIMESINK.MOD7.24 [Under Development] MST Sink test
QDPRIMESINK.MOD7.25 [Under Development] MST Sink test
QDPRIMESINK.MOD7.26 [Under Development] MST Sink test
QDPRIMESINK.MOD7.27 [Under Development] MST Sink test
QDPRIMESINK.MOD7.28 [Under Development] MST Sink test
QDPRIMESINK.MOD7.29 [Under Development] MST Sink test
QDPRIMESINK.MOD7.30 [Under Development] MST Sink test
QDPRIMESINK.MOD7.31 [Under Development] MST Sink test
QDPRIMESINK.MOD7.32 [Under Development] MST Sink test
QDPRIMESINK.MOD7.33 [Under Development] MST Sink test
QDPRIMESINK.MOD7.34 [Under Development] MST Sink test
QDPRIMESINK.MOD7.35 [Under Development] MST Sink test
QDPRIMESINK.MOD7.36 [Under Development]

MST Sink test
QDPRIMESINK.MOD7.37 [Under Development] MST Sink test
QDPRIMESINK.MOD7.38 [Under Development] MST Sink test
QDPRIMESINK.MOD7.39 [Under Development] MST Sink test
QDPRIMESINK.MOD7.40 [Under Development] MST Sink test
QDPRIMESINK.MOD7.41 [Under Development] MST Sink test
QDPRIMESINK.MOD7.42 [Under Development] MST Sink test
QDPRIMESINK.MOD7.43 [Under Development] MST Sink test
QDPRIMESINK.MOD7.44 [Under Development] MST Sink test
QDPRIMESINK.MOD7.45 [Under Development] MST Sink test
QDPRIMESINK.MOD7.46 [Under Development] MST Sink test
QDPRIMESINK.MOD7.47 [Under Development] MST Sink test
QDPRIMESINK.MOD7.48 [Under Development] MST Sink test
QDPRIMESINK.MOD7.49 [Under Development] MST Sink test
QDPRIMESINK.MOD7.50 [Under Development] MST Sink test

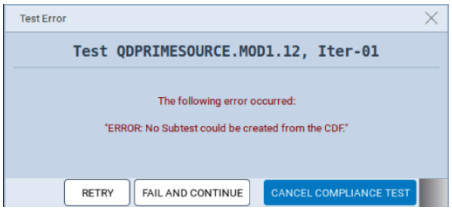
6 qdPrime Test Execution

After selecting specific tests to run, the third tab (**Test Options / Preview**) will automatically populate with all selected tests. Several configurable options are available with the "**Count**" and "**Options**" button. This includes:



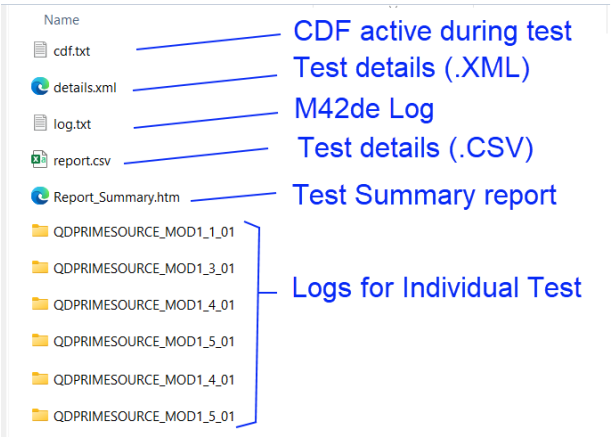
- Setting the repeat count for individual tests.
- Setting the option to automatically end "Repeat for any test that fails".
- Setting the option to "Always Save Data", "Only on failures", or "Never save".

Clicking the "**Execute Tests**" button will initiate tests. User will be prompted to name the test result folder before testing begins. Tests will run sequentially, and the test summary screen will update with test result after all iterations are completed. In some cases qdPrime may report that specific tests are not valid for the current CDF. The message below indicates users have selected tests with no valid corresponding entry in the CDF. The test suite allows users to select Retry, Fail (skip) or Cancel the test run. To avoid this, for example, do not select Standard timing test (Mod1.12) if there are no entries in the Standard timing tab of the CDF.



7 qdPrime Test Reports

Each test run automatically saves a test report in several formats including HTML, XML and CSV. The test output files are are saved in the test output folder on the M42de platform (right). Upon completion of the test run, the test window provides a **Compliance Test Result Viewer** (below left). This contains the same details as the XML report and can be expanded to reveal which assertions passed and failed. ACA and test logs for each individual test will be saved in the same sub-folder. Test output files can be moved/transferred from the M42de platform according to Error! Reference source not found. section. Clicking the **HTML report** button will open the qdPrime Test Summary.htm report (below right).



Compliance Test Result Viewer

qdPrime Test Summary (HTML) Report