

Universal Serial Bus  
Device Class Definition  
for  
Video Devices  
H.264 Payload

Compliance Test Specification

Revision 1.00

April 27, 2011

## Contributors

Stephen Cooper	Microsoft Corp.
Richard Webb	Microsoft Corp.
Andrei Jefremov	Skype
Remy Zimmermann	Logitech Inc.
Chandrashekhar Rao	Logitech Inc.

Copyright © 2011, USB Implementers Forum, Inc.

All rights reserved.

A LICENSE IS HEREBY GRANTED TO REPRODUCE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, IS GRANTED OR INTENDED HEREBY.

USB-IF AND THE AUTHORS OF THIS SPECIFICATION EXPRESSLY DISCLAIM ALL LIABILITY FOR INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. USB-IF AND THE AUTHORS OF THIS SPECIFICATION ALSO DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE THE INTELLECTUAL PROPERTY RIGHTS OF OTHERS.

THIS SPECIFICATION IS PROVIDED "AS IS" AND WITH NO WARRANTIES, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE. ALL WARRANTIES ARE EXPRESSLY DISCLAIMED. NO WARRANTY OF MERCHANTABILITY, NO WARRANTY OF NON-INFRINGEMENT, NO WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE, AND NO WARRANTY ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.

IN NO EVENT WILL USB-IF OR USB-IF MEMBERS BE LIABLE TO ANOTHER FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA OR ANY INCIDENTAL, CONSEQUENTIAL, INDIRECT, OR SPECIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SPECIFICATION, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

All product names are trademarks, registered trademarks, or service marks of their respective owners.

#### **AVC/H.264 Disclaimer**

Any implementation of the specification described herein would require a MPEG LA AVC/H.264 Patent Portfolio license to essential patent rights for the AVC/H.264 (MPEG-4 Part 10) digital video coding standard. See <http://www.MPEGLA.com>.

## Revision History

Version	Date	Description
0.10	February 22, 2011	Initial template
0.20	March 14, 2011	Added test cases for compliance. More test cases to be added and It is draft version.
0.30	March 29, 2011	Added Dynamic test and sync up with latest spec. The LTR related XUControl will be added after LTR review.
0.90	April 4, 2011	Added LTR, added MIN/MAX check, updated max bit check,
0.91	April 13, 2011	Updated based on Review and comments. Added combination of parameters test.
0.92	April 19, 2011	Updated based on Review and comments.
0.93	April 20, 2011	Updated after review.
0.94	April 22, 2011	Updated based on changes in the specification.
1.00	April 26, 2011	Updated based on XU control order

## Table of Contents

1	Scope.....	1
2	Related Documents.....	1
3	Terms and Abbreviations .....	2
4	Test philosophy .....	4
4.1	Implementation .....	4
5	Assertions.....	4
5.1	Extension Unit Control related Assertions.....	4
5.1.1	XU Basic Static Assertions.....	4
5.1.2	XU Basic Dynamic Assertions.....	5
5.1.3	XU Control MIN/MAX test Assertions .....	7
5.1.4	XU Control Combination test Assertions.....	7
6	Description of tests .....	7
6.1	General Procedures .....	7
6.1.1	Init procedure .....	7
6.1.2	Reset Endpoint procedure.....	8
6.1.3	“Unfreeze” device procedure.....	8
6.2	Test details.....	8
6.2.1	Static XU Control tests.....	8
6.2.1.1	Device XU Control Basic Tests.....	8
6.2.1.2	Device XU Control dwFrameInterval Tests .....	9
6.2.1.3	Device XU Control wWidth and wHeight Tests.....	9
6.2.1.4	Device XU Control wLeakyBucketSize Tests.....	10
6.2.1.5	Device XU Control dwBitRate Tests .....	11
6.2.1.6	Device XU Control Slice Mode Tests .....	11
6.2.1.7	Device XU Control wIFramePeriod Tests .....	12
6.2.1.8	Device XU Control device reported delay Tests.....	13
6.2.1.9	Device XU Control device reported bRateControlModeTests .....	14
6.2.1.10	Device XU Control device reported SVC Scale ModeTests .....	14
6.2.1.11	Device XU Control device reported bStreamMuxOption Tests .....	15
6.2.1.12	Device XU Control device reported bStreamFormat Tests .....	17
6.2.1.13	Device XU Control device reported bEntropyCABAC Tests.....	17
6.2.1.14	Device XU Control device reported bTimestampTests .....	18
6.2.1.15	Device XU Control device reported bNumOfReorderFramesTests.....	19
6.2.1.16	Device XU Control device reported bPreviewFlipped Tests .....	19
6.2.1.17	Device XU Control device reported bView parameter Tests .....	20
6.2.1.18	Device XU Control device reported bStreamID parameter Tests .....	21
6.2.1.19	Device XU Control device reported bSpatialLayerRatio parameter Tests .....	22
6.2.2	Dynamic XU Control tests.....	22
6.2.2.1	Device XU Control device reported bRateControlMode parameter Tests .....	23
6.2.2.2	Device XU Control device reported bTemporalScaleMode parameter Tests.....	23
6.2.2.3	Device XU Control device reported bSpatialScaleMode parameter Tests .....	25
6.2.2.4	Device XU Control device reported bSNRScaleMode parameter Tests.....	26
6.2.2.5	Device XU Control device reported UVCX_BITRATE_LAYERS parameter Tests .....	27
6.2.2.6	Device XU Control device reported UVCX_QP_STEPS_LAYERS parameter Tests .....	29
6.2.2.7	Device XU Control device reported UVCX_PICTURE_TYPE_CONTROL parameter Tests..	32

6.2.2.8	Device XU Control device reported UVCX_FRAMERATE_CONFIG parameter Tests.....	32
6.2.2.9	Device XU Control device reported UVCX_VIDEO_ADVANCE_CONFIG parameter Tests	33
6.2.2.10	Device XU Control device reported UVCX_ENCODER_RESET parameter Tests.....	34
6.2.2.11	Device XU Control device reported UVCX_LTR_BUFFER_SIZE_CONTROL Test .....	35
6.2.2.12	Device XU Control device reported UVCX_LTR_PICTURE_CONTROL Tests .....	35
6.2.3	Static XU Control MIN/MAX tests.....	36
6.2.3.1	Device XU Control device reported UVCX_VIDEO_CONFIG_PROBE's MIN/MAX Tests...	36
6.2.4	Compliance test with the combination of parameters settings.....	40

## 1 Scope

This document specifies assertions and test procedures for use with Video Class devices supporting H.264 Payload.

This testing is intended to be in addition to standard USB Compliance testing; assertions covered by the USBCV test document, for instance, are not covered here.

This testing applies to one configuration and all VIC at a time. This testing covers the validation of the Video class-specific Descriptors (including the possible dependencies between them) and of the Class-specific control requests (except those explicitly excluded above).

## 2 Related Documents

[1] USB Video Class 1.1 ([http://www.usb.org/developers/devclass\\_docs#approved](http://www.usb.org/developers/devclass_docs#approved))

[2] USB\_Video\_Payload\_Frame\_Based\_1.1

[3] USB\_Video\_Payload\_Stream\_Based\_1.1

[4] USB\_Video\_Payload\_MJPEG\_1.1

[5] RTP Payload for H.264 (<http://tools.ietf.org/html/rfc3914>)

[6] ITU H.241 (<http://www.itu.int/itu-t/recommendations/index.aspx?ser=H>)

[7] ITU T.81 (<http://www.itu.int/itu-t/recommendations/index.aspx?ser=T>)

[8] The H.264/MPEG-4 AVC standard (<http://www.itu.int/rec/T-REC-H.264>) (referred to hereafter simply as H.264) is specified in the following document:

- a. ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced video coding for generic audiovisual services. The standard is available at. Unless otherwise specified, this document refers to the edition approved by ITU-T in March 2010 (posted at the ITU-T web site link above).
- b. The Scalable Video Coding (SVC) extensions to the H.264/MPEG-4 AVC standard (referred to hereafter simply as SVC) are specified in Annex G of the above document.
- c. The Multiview Video Coding (MVC) extensions to the H.264/MPEG-4 AVC standard (referred to hereafter simply as MVC) are specified in Annex H of the above document.

[9] When supported, the use of SVC and simulcast of multiple streams in the context of this specification shall additionally conform to the following specification:

- a. Unified Communication Specification and Interfaces for H.264/MPEG-4 AVC and SVC Encoder Implementation

- b. The specification is available at <http://technet.microsoft.com/en-us/lync> . Unless otherwise specified, this document refers to the edition of version 1.01 (posted at the Microsoft web site link above).

### 3 Terms and Abbreviations

Term	Definition
AVC	Advanced Video Coding (see H.264)
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CBR	Constant Bit Rate
CPB	Coded Picture Buffer
DPB	Decoded Picture Buffer
H.264	ISO/IEC 14496 Part 10
IDR	Instantaneous Decoder Refresh. Intraframe with no past reference.
MB	Macroblock
MJPG	Motion JPEG. See UVC standard reference payload specification.
MPF	Multiplexed Payload Format
MVC	Multiview Video Coding
NAL	Network Abstract Layer
NALU	Network Access Layer Unit
NV12	Planar 4:2:0 format with Y-plane followed by plane of interleaved U/V (see <a href="http://www.fourcc.org/yuv.php#NV12">http://www.fourcc.org/yuv.php#NV12</a> )
PPS	Picture Parameter Set
QP	Quantization Parameter
SCR	Source Clock Reference
SEI	Supplemental Enhancement Information
SPS	Sequence Parameter Set
SVC	Scalable Video Coding



USB	Universal Serial Bus
UVC	USB Video Class
VBR	Variable Bit Rate
VC	Video Control
VS	Video Streaming
VUI	Video Usability Information
XU	Extension Unit
YUY2	Interleaved 16-bit YUV data. Y, U, Y, V.

## 4 Test philosophy

### 4.1 Implementation

The test description specified by this document will be part of USBH264 tool available from the USB-IF. It is intended that all devices that report a UVC H.264 Payload will be required to pass this test in order to receive the logo certification.

## 5 Assertions

### 5.1 Extension Unit Control related Assertions

#### 5.1.1 XU Basic Static Assertions

Extension Unit (XU) Control Basic Assertion

Num	Assertions
5.1.1.1.	Device UVCX_VIDEO_CONFIG_PROBE does not provide default configuration. <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD11</a>
5.1.1.2.	Device UVCX_VIDEO_CONFIG_PROBE does not change dwFrameInterval <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD12</a>
5.1.1.3.	Device UVCX_VIDEO_CONFIG_PROBE does not change wWidth and wHeight <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD13</a>
5.1.1.4.	Device UVCX_VIDEO_CONFIG_PROBE does not change wLeakyBucketSize <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD14</a>
5.1.1.5.	Device UVCX_VIDEO_CONFIG_PROBE does not change dwBitRate <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD15</a>
5.1.1.6.	Device UVCX_VIDEO_CONFIG_PROBE does not change wSliceMode <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD16</a>
5.1.1.7.	Device UVCX_VIDEO_CONFIG_PROBE does not change wFramePeriod <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD17</a>
5.1.1.8.	Device UVCX_VIDEO_CONFIG_PROBE does not have valid wEstimatedVideoDelay and wEstimatedMaxConfigDelay <b>Specification Reference:</b> USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1 <b>Test Description:</b> <a href="#">TD18</a>

- 5.1.1.9. Device UVCX\_VIDEO\_CONFIG\_PROBE does not change bRateControlMode.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD19](#)
- 5.1.1.10. Device UVCX\_VIDEO\_CONFIG\_PROBE does not change bTemporalScaleMode, bSpatialScaleMode and bSNRScaleMode.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD20](#)
- 5.1.1.11. Device UVCX\_VIDEO\_CONFIG\_PROBE does not change bStreamMuxOption mode.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD21](#)
- 5.1.1.12. Device UVCX\_VIDEO\_CONFIG\_PROBE does not have valid bStreamFormat.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD22](#)
- 5.1.1.13. Device UVCX\_VIDEO\_CONFIG\_PROBE bEntropyCABAC does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD23](#)
- 5.1.1.14. Device UVCX\_VIDEO\_CONFIG\_PROBE bTimestamp does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD24](#)
- 5.1.1.15. Device UVCX\_VIDEO\_CONFIG\_PROBE bNumOfReorderFrames does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD25](#)
- 5.1.1.16. Device UVCX\_VIDEO\_CONFIG\_PROBE bPreviewFlipped does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD26](#)
- 5.1.1.17. Device UVCX\_VIDEO\_CONFIG\_PROBE bView does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD27](#)
- 5.1.1.18. Device UVCX\_VIDEO\_CONFIG\_PROBE bStreamID does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD28](#)
- 5.1.1.19. Device UVCX\_VIDEO\_CONFIG\_PROBE bSpatialLayerRatio does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.1  
**Test Description:** [TD29](#)

## 5.1.2 XU Basic Dynamic Assertions

Extension Unit (XU) Control Basic Assertion

Num	Assertions
-----	------------

- 5.1.1.20. Device UVCX\_RATE\_CONTROL\_MODE bRateControlMode does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.3  
**Test Description:** [TD30](#)
- 5.1.1.21. Device UVCX\_TEMPORAL\_SCALE\_MODE bTemporalScaleMode does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.4  
**Test Description:** [TD31](#)
- 5.1.1.22. Device UVCX\_SPATIAL\_SCALE\_MODE bSpatialScaleMode does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.5  
**Test Description:** [TD32](#)
- 5.1.1.23. Device UVCX\_SNR\_SCALE\_MODE bSNRScaleMode does not work  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.6  
**Test Description:** [TD33](#)
- 5.1.1.24. Device XU Control UVCX\_BITRATE\_LAYERS of dwPeakBitrate and dwAverageBitrate do not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.14  
**Test Description:** [TD34](#)
- 5.1.1.25. Device XU Control UVCX\_QP\_STEPS\_LAYERS of bFrameType, bMinQP and bMaxQP do not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.15  
**Test Description:** [TD35](#)
- 5.1.1.26. Device XU Control UVCX\_PICTURE\_TYPE\_CONTROL of wPicType does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.9  
**Test Description:** [TD36](#)
- 5.1.1.27. Device XU Control UVCX\_FRAMERATE\_CONFIG of dwFrameInterval does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.12  
**Test Description:** [TD37](#)
- 5.1.1.28. Device XU Control UVCX\_VIDEO\_ADVANCE\_CONFIG of dwMb\_max and blevel\_idc do not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.13  
**Test Description:** [TD38](#)
- 5.1.1.29. Device XU Control UVCX\_ENCODER\_RESET does not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.11  
**Test Description:** [TD39](#)
- 5.1.1.30. Device XU Control UVCX\_LTR\_BUFFER\_SIZE\_CONTROL of bLTRBufferSize and bLTREncoderControl do not work.  
**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.7

**Test Description:** [TD40](#)

- 5.1.1.31. Device XU Control UVCX\_LTR\_PICTURE\_CONTROL bPutAtPositionInLTRBuffer and bEncodeUsingLTR do not work.

**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.8

**Test Description:** [TD41](#)

### 5.1.3 XU Control MIN/MAX test Assertions

Extension Unit (XU) Control MIN/MAX Assertion

Num	Assertions
-----	------------

- 5.1.1.32. Device XU Control UVCX\_VIDEO\_CONFIG\_PROBE MIN and MAX limit does not work.

**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.2

**Test Description:** [TD42](#)

### 5.1.4 XU Control Combination test Assertions

Extension Unit (XU) Control Combination Assertion

Num	Assertions
-----	------------

- 5.1.1.33 Device XU Control UVCX\_VIDEO\_CONFIG\_PROBE set with a good combination does not work.

**Specification Reference:** USB Device Class Definition for Video Devices: H.264 Payload Section 3.3.2

**Test Description:** [TD43](#)

## 6 Description of tests

The Every test for the XU control test is designed to run independently. This allows user to run specific test to validate the device XU control configuration and the device responses.

### 6.1 General Procedures

The initialization procedure is always called before starting the test.

#### 6.1.1 Init procedure

The followings are Init procedure for the device under test.

1. The device should be enumerated and set to default configuration mode.
2. Check the Exposed XU control. The exposed XU control shall contain all XU controls defined in the specification USB for Video devices: H.264 Payload.
3. Check the device by reading the XU control GET\_DEF for UVCX\_VERSION.
4. If device is not alive, execute the “Unfreeze” device procedure.

### **6.1.2 Reset Endpoint procedure**

If an Endpoint is stalled, do the following:

1. Send a reset command to the stalled Endpoint
2. Check Endpoint state to see if it is now functional
3. If not, execute the Init procedure and fail the current test

### **6.1.3 “Unfreeze” device procedure**

If device under test is frozen and does not respond to the Init procedure, do the following:

1. Power cycle of the device under test by reconnecting the USB port.
2. Re-enumerate the device under test.
3. Check the device by reading the XU control GET\_DEF for UVCX\_VERSION.
4. If device does not respond turn off the power of the device and exit the test with a failure log.

## **6.2 Test details**

### **6.2.1 Static XU Control tests**

#### **6.2.1.1 Device XU Control Basic Tests**

##### **TD 1.1 Device XU Control Basic Test**

This test verifies the device XU Static Control configuration for the device’s default settings.

#### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Check for the following parameters.
  - dwFrameInterval has valid value. Not equal to zero.

- dwBitRate has valid value. Not equal to zero.
  - wWidth has valid value. Not equal to zero.
  - wHeight has valid value. Not equal to zero.
  - wProfile has valid value. Not equal to zero.
6. If step (5) fails for any parameters, fail the test and throw related assertion ([5.1.1.1](#)).

#### **6.2.1.2 Device XU Control dwFrameInterval Tests**

##### **TD 1.2 Device XU dwFrameInterval Test**

This test verifies the XU control of dwFrameInterval parameters.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MIN for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Save the value of dwFrameInterval in test tool. (e.g. dwFrameInterval=333333)
6. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
7. Change the dwFrameInterval to higher (lower frame rate) e.g. 666667
8. SET\_CUR with new dwFrameInterval values.
9. GET\_CUR for UVCX\_VIDEO\_CONFIG\_PROBE and check with configured value of dwFrameInterval or valid supported by the device.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.2](#) )

#### **6.2.1.3 Device XU Control wWidth and wHeight Tests**

##### **TD 1.3 Device XU Control wWidth and wHeight Test**

This test verifies the XU control of Control wWidth and wHeight parameters.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Save the value of Control wWidth and wHeight in test tool. (e.g. Control wWidth = 1920 and wHeight=1080)
6. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
7. Change the wWidth and wHeight. (e.g. Control wWidth = 1280 and wHeight = 720).
8. SET\_CUR with new wWidth and wHeight values.
9. GET\_CUR for UVCX\_VIDEO\_CONFIG\_PROBE and check for set value of wWidth and wHeight. The device shall return the set value or valid supported value for the device.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.3](#))

#### **6.2.1.4 Device XU Control wLeakyBucketSize Tests**

##### **TD 1.4 Device XU wLeakyBucketSize Test**

This test verifies the XU control of wLeakyBucketSize parameters.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Save the value of wLeakyBucketSize in test tool. (e.g . wLeakyBucketSize =1000)
6. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
7. Change the wLeakyBucketSize to 500.



8. SET\_CUR with new wLeakyBucketSize values.
9. GET\_CUR for UVCX\_VIDEO\_CONFIG\_PROBE and check for the set value of wLeakyBucketSize.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.4](#) )

#### **6.2.1.5 Device XU Control dwBitRate Tests**

##### **TD 1.5 Device XU dwBitRate Test**

This test verifies the XU control of dwBitRate parameters.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Save the value of dwBitRate in test tool. (e.g. dwBitRate = 3000000)
6. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
7. Change the dwBitRate to the device supported settings. (e.g. dwBitRate = 512000)
8. SET\_CUR with new dwBitRate values.
9. GET\_CUR for UVCX\_VIDEO\_CONFIG\_PROBE and check for the set value.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.5](#) )

#### **6.2.1.6 Device XU Control Slice Mode Tests**

##### **TD 1.6 Device XU Slice Mode Test**

This test verifies the XU control of wSliceUnits and wSliceMode parameters for the minimum required value (one slice per frame).

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Change the wSliceUnits and wSliceMode parameters.
  - SET\_CUR with wSliceMode = 0 and verify by GET\_CUR
6. If step (5) fails, fail the test and throw related assertion ([5.1.1.6](#))
7. Change the wSliceUnits and wSliceMode parameters.
  - SET\_CUR with wSliceMode = device supported value (e.g. 0x0001) and wSliceUnits = device supported value (e.g. 0x1F40 for 8 Kb)
8. GET\_CUR to verify the settings. Device may return wSliceMode = 0x0000, which means device does not support wSliceMode = 0x0001 option.
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.6](#))
10. Change the wSliceUnits and wSliceMode parameters.
  - SET\_CUR with wSliceMode = device supported value (e.g. 0x0002) and wSliceUnits = device supported value (e.g. 0x0078 for 120 Macroblocks)
11. GET\_CUR to verify the settings. Device may return wSliceMode = 0x0000, which means device does not support wSliceMode = 0x0002 option.
12. If step (11) fails, fail the test and throw related assertion ([5.1.1.6](#))
13. Change the wSliceUnits and wSliceMode parameters.
  - SET\_CUR with wSliceMode = device supported value (e.g. 0x0003) and wSliceUnits = device supported value (e.g. 0x0004 for 4 slices per frame)
14. GET\_CUR to verify the settings. Device may return wSliceMode = 0x0001 and wSliceUnits = 0x0001 (one slice per frame).
15. If step (14) fails, fail the test and throw related assertion ([5.1.1.6](#))

#### **6.2.1.7 Device XU Control wFramePeriod Tests**

##### **TD 1.7 Device XU wFramePeriod Test**

This test verifies the XU control of wIFramePeriod. This is to set and verify the I frame intervals in units in ms.

### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Change the wIFramePeriod to 1000. (I frame every 1000 ms)
6. SET\_CUR with new wIFramePeriod values.
7. GET\_CUR for control UVCX\_VIDEO\_CONFIG\_PROBE and check the set value of wIFramePeriod.
8. If step (7) fails, fail the test and throw related assertion ([5.1.1.7](#) )

### 6.2.1.8 Device XU Control device reported delay Tests

#### TD 1.8 Device XU wEstimatedVideoDelay and wEstimatedMaxConfigDelay Test

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Check the following parameters.
  - wEstimatedVideoDelay has valid value. Not equal to zero.
  - wEstimatedMaxConfigDelay has valid value. Not equal to zero.
6. If step (5) fails for any parameters, fail the test and throw related assertion ([5.1.1.8](#))

### **6.2.1.9 Device XU Control device reported bRateControlModeTests**

#### **TD 1.9 Device XU bRateControlMode Test**

This test verifies the XU control of bRateControlMode.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the valid value of bRateControlMode.
6. If step (5) fails, fail the test and throw related assertion ([5.1.1.9](#)).
7. Change the bRateControlMode to valid values 1 to 3.
8. SET\_CUR with new bRateControlMode values.
9. GET\_CUR for UVCX\_VIDEO\_CONFIG\_PROBE and check the set value of bRateControlMode. The device may return the valid different value, as device may not support requested value.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.9](#)).

### **6.2.1.10 Device XU Control device reported SVC Scale ModeTests**

#### **TD 2.0 Device XU Control bTemporalScaleMode, bSpatialScaleMode and bSNRScaleMode Test**

The test verifies the XU control bTemporalScaleMode, bSpatialScaleMode and bSNRScaleMode.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.

3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the value for non zero.  

bTemporalScaleMode

bSpatialScaleMode

bSNRScaleMode
6. If step (5) has all zero, then log and exit the test as parameters are not supported by the device.
7. Change the value of bTemporalScaleMode by decrementing the value of GET\_MAX by one.
8. Set the new value of bTemporalScaleMode with SET\_CUR.
9. Verifies the GET\_CUR value.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.10](#)).
11. Change the value of bSpatialScaleMode by decrementing the value of GET\_MAX by one.
12. Set the new value of bSpatialScaleMode with SET\_CUR.
13. Verifies the GET\_CUR value.
14. If step (13) fails, fail the test and throw related assertion ([5.1.1.10](#)).
15. Change the value of bSNRScaleMode by decrementing the value of GET\_MAX by one.
16. Set the new value of bSNRScaleMode with SET\_CUR.
17. Verifies the GET\_CUR value.
18. If step (17) fails, fail the test and throw related assertion ([5.1.1.10](#)).

#### **6.2.1.11 Device XU Control device reported bStreamMuxOption Tests**

##### **TD 2.1** Device XU Control device reported bStreamMuxOption Tests

This test verifies the XU control of bStreamMuxOption.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify Supported stream type.
6. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
7. SET\_CUR with Bit 0 set to 1 for auxiliary stream and Bit1 set to 1 for H.264 stream.
8. GET\_CUR and verify the Bit 0 and Bit 1 as set.
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.11](#))
10. Check the resolution (wWidth or/and wHeight). If resolution is set to zero by the device. (The resolution equal to zero implies that the device cannot support)
11. Check the Bit2 from GET\_MAX
  - a. If step (11) is supported, then SET\_CUR with Bit 0 set to 1 for auxiliary stream, Bit1 set to 1 for H.264 stream and Bit2 for YUY2.
  - b. GET\_CUR and verify the Bit 0, Bit 1 and Bit 2 as set.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.11](#)).
  - d. Check the resolution (wWidth or/and wHeight). If resolution is set to zero by the device. (The resolution equal to zero implies that the device cannot support).
  - e. Go to step (12).
12. Check the Bit3 from GET\_MAX
  - a. If step (12) is supported, then SET\_CUR with Bit 0 set to 1 for auxiliary stream, Bit1 set to 1 for H.264 stream and Bit3 for NV12.
  - b. GET\_CUR and verify the Bit 0, Bit 1 and Bit 2 as set.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.11](#)).
  - d. Check the resolution (wWidth or/and wHeight). If resolution is set to zero by the device. (The resolution equal to zero implies that the device cannot support).
  - e. Go to step (13).
13. If resolutions are not set to zero and GET\_MAX has Bit 2 and 3 supported.

- a. If step (13) is supported, then SET\_CUR with Bit 0 set to 1 for auxiliary stream, Bit1 set to 1 for H.264 stream, Bit2 for YUY2 and Bit3 for NV12.
- b. GET\_CUR and verify the Bit 0, Bit 1, Bit 2 and Bit 3 as set.
- c. If step (b) fails, fail the test and throw related assertion ([5.1.1.11](#)).
- d. Check the resolution (wWidth or/and wHeight). If resolution is set to zero by the device. (The resolution equal to zero implies that the device cannot support).

#### **6.2.1.12 Device XU Control device reported bStreamFormat Tests**

##### **TD 2.2 Device XU bStreamFormat Test**

This test verifies the XU control of bStreamFormat.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the bStreamFormat for valid values 0 and 1.
6. If step (5) fails, fail the test and throw related assertion ([5.1.1.12](#)).
7. SET\_CUR with bStreamFormat = 0.
8. GET\_CUR and verify value for valid values 0 or 1.
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.12](#)).
10. SET\_CUR with bStreamFormat = 1.
11. GET\_CUR and verify valid values 0 or 1.
12. If step (11) fails, fail the test and throw related assertion ([5.1.1.12](#)).

#### **6.2.1.13 Device XU Control device reported bEntropyCABAC Tests**

##### **TD 2.3 Device XU bEntropyCABAC Test**

This test verifies the XU control of bEntropyCABAC.

#### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the bEntropyCABAC valid value.
6. If step (5) fails, fail the test and throw related assertion ([5.1.1.13](#)).
7. SET\_CUR with bEntropyCABAC = 1.
8. GET\_CUR and verify the valid value. The device may return with requested value or supported value. In this case the value can be 0 or 1.
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.13](#)).

#### **6.2.1.14 Device XU Control device reported bTimestampTests**

##### **TD 2.4 Device XU bTimestamp Test**

This test verifies the XU control of bTimestamp.

#### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the valid value returned by the device.
6. If step (5) fails, fail the test and throw related assertion ([5.1.1.14](#)).



7. SET\_CUR with bTimestamp =1.
8. GET\_CUR and verify valid values bTimestamp = 1 or bTimestamp = 0.
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.14](#)).

#### **6.2.1.15 Device XU Control device reported bNumOfReorderFramesTests**

##### **TD 2.5 Device XU bNumOfReorderFrames Test**

This test verifies the XU control of bNumOfReorderFrames.

#### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. If the value of bNumOfReorderFrames is not zero.
  - a. SET\_CUR bNumOfReorderFrames = 0.
  - b. GET\_CUR and verify the value for zero.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.15](#)).
  - d. Log and exit the test.
6. SET\_CUR bNumOfReorderFrames = 0x04.
7. GET\_CUR verify value. The device may return the set value or zero or device supported value.

#### **6.2.1.16 Device XU Control device reported bPreviewFlipped Tests**

## TD 2.6 Device XU bPreviewFlipped Test

This test verifies the XU control of bPreviewFlipped.

### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the valid value.
6. If step (5) fails, fail the test and throw related assertion ([5.1.1.16](#)).
7. If the value of bPreviewFlipped is not zero.
  - a. SET\_CUR bPreviewFlipped = 0.
  - b. GET\_CUR and verify the value for zero.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.16](#)).
  - d. Log and exit the test.
8. If the value of bPreviewFlipped is zero.
  - a. SET\_CUR bPreviewFlipped = 1.
  - b. GET\_CUR and verify the value for 1 or 0.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.16](#)).

## 6.2.1.17 Device XU Control device reported bView parameter Tests

### TD 2.7 Device XU bView parameter Test

This test verifies the XU control of bView parameter.

### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. If bView is equal to zero.
  - a. Log and exit the test. (Feature not supported)
6. SET\_CUR with lower than or equal to the GET\_MAX value of bView.
7. GET\_CUR and Verify bView value with set value.
8. If step (7) fails, fail the test and throw related assertion ([5.1.1.17](#)).

#### **6.2.1.18 Device XU Control device reported bStreamID parameter Tests**

TD 2.8 Device XU bStreamID parameter Test

This test verifies the XU control of bStreamID parameter.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify the bStreamID.
  - a. If bStreamID == 0, then log and exit the test. (only one stream supported)
6. Verify the bStreamID for valid values. (0x0000 to 0x0006).
7. If step (6) fails, fail the test and throw related assertion ([5.1.1.18](#)).

8. SET\_CUR with bStreamMuxOption enable bit 0, bit 1, and bStreamID less than or equal to the GET\_MAX value.
9. GET\_CUR and verify the value of bStreamID.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.18](#)).

#### **6.2.1.19 Device XU Control device reported bSpatialLayerRatio parameter Tests**

##### **TD 2.9 Device XU bSpatialLayerRatio parameter Test**

This test verifies the XU control of bSpatialLayerRatio parameter.

#### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Verify bSpatialScaleMode value
  - a. If bSpatialScaleMode == 0, then log and exit the test. (SVC Spatial Layer not supported).
6. If the value of bSpatialScaleMode is more than zero.
7. SET\_CUR bSpatialLayerRatio with 0x18.
8. GET\_CUR and the device should return set value or its supported value.
9. SET\_CUR bSpatialLayerRatio with 0x20.
10. GET\_CUR and the device should return set value or its supported value.

#### **6.2.2 Dynamic XU Control tests**

### **6.2.2.1 Device XU Control device reported bRateControlMode parameter Tests**

#### **TD 3.0 Device UVCX\_RATE\_CONTROL\_MODE bRateControlMode parameter Test**

This test verifies the UVCX\_RATE\_CONTROL\_MODE control of bRateControlMode parameter.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_CUR for XU Control UVCX\_RATE\_CONTROL\_MODE defined in specification section 3.3.3.
8. Verify the valid value of bRateControlMode.
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.20](#)).
10. Change the bRateControlMode to valid values 1 or 2 or 3.
11. SET\_CUR with new bRateControlMode values and set the StreamID (part of wLayerID) with the value of bStreamID used in step (6).
12. GET\_CUR for UVCX\_RATE\_CONTROL\_MODE and check for set value of bRateControlMode. The device may return the valid different value, as the device may not support set value.
13. If step (10) fails, fail the test and throw related assertion ([5.1.1.20](#)).

### **6.2.2.2 Device XU Control device reported bTemporalScaleMode parameter Tests**

TD 3.1 This test verifies the UVCX\_TEMPORAL\_SCALE\_MODE control of bTemporalScaleMode parameter.

### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU Control UVCX\_TEMPORAL\_SCALE\_MODE defined in specification section 3.3.4.
8. Verify the value of bTemporalScaleMode for valid range (0-6 range).
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.21](#)).
10. If bTemporalScaleMode is equal to zero.
  - a. Log and exit the test as device does not support requested temporal scalability.
11. Check StreamID of wLayerID.
12. If StreamID of wLayerID is equal to zero.
  - a. SET\_CUR with less than max value of bTemporalScaleMode and StreamID equal to zero.
  - b. Verify the value.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.21](#)).
13. If StreamID of wLayerID is greater than 0 and equal to n.
  - a. SET\_CUR with less than max value of bTemporalScaleMode and StreamID equal to 1 to n.
  - b. Verify the value.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.21](#)).

- d. Go to step (a) and repeat the test for the supported remaining StreamIDs (up to n).

### 6.2.2.3 Device XU Control device reported bSpatialScaleMode parameter Tests

TD 3.2 This test verifies the UVCX\_ SPATIAL \_SCALE\_ MODE control of bSpatialScaleMode parameter.

#### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU Control UVCX\_SPATIAL\_SCALE\_MODE defined in specification section 3.3.5.
8. Verify the value of bSpatialScaleMode for the valid range (0-7 range).
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.22](#))
10. If bSpatialScaleMode is equal to zero.
  - a. Log and exit the test as the device do not support requested spatial scalability.
11. Check StreamID of wLayerID.
12. If StreamID of wLayerID is equal to or greater than zero.
  - a. SET\_CUR with less than max value of bSpatialScaleMode and StreamID equal to zero.
  - b. Verify the value.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.22](#))
13. If StreamID of wLayerID is greater than 0 and equal to n.

- a. SET\_CUR with less than max value of bSpatialScaleMode and StreamID equal to 1 to n.
- b. Verify the value.
- c. If step (b) fails, fail the test and throw related assertion ([5.1.1.22](#)).
- d. Go to step (a) and repeat the test for the supported remaining StreamIDs (up to n).

#### 6.2.2.4 Device XU Control device reported bSNRScaleMode parameter Tests

TD 3.3 This test verifies the UVCX\_SNR\_SCALE\_MODE control of bSNRScaleMode parameter.

##### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU Control UVCX\_SNR\_SCALE\_MODE defined in specification section 3.3.6.
8. Verify the value of bSNRScaleMode for the valid range (0-6 range).
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.23](#)).
10. If bSpatialSNRMode is equal to zero.
  - a. Log and exit the test as the device do not support requested spatial scalability.
11. Check StreamID of wLayerID.
12. If StreamID of wLayerID is equal to or greater than zero.
  - a. SET\_CUR with less than max value of bSNRScaleMode and StreamID equal to zero.



- b. Verify the value.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.23](#))
- 13. If StreamID of wLayerID is greater than 0 and less than or equal to n.
  - a. SET\_CUR with less than max value of bSNRScaleMode and StreamID equal to 1 to n.
  - b. Verify the value.
  - c. If step (b) fails, fail the test and throw related assertion ([5.1.1.23](#)).
  - d. Go to step (a) and repeat the test for the supported remaining StreamIDs (up to n).

#### **6.2.2.5 Device XU Control device reported UVCX\_BITRATE\_LAYERS parameter Tests**

TD 3.4 This test verifies the UVCX\_BITRATE\_LAYERS control of dwPeakBitrate and dwAverageBitrate parameter.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU Control UVCX\_BITRATE\_LAYERS defined in specification section 3.3.14.
8. Verify the wLayerID returned valid value.
9. If wLayerID value is not valid, fail the test and throw related assertion ([5.1.1.24](#)).
10. If wLayerID is equal to zero.

- a. SET\_CUR with dwPeakBitrate and dwAverageBitrate for wLayerID zero. The dwPeakBitrate and dwAverageBitrate value should be in the range from MIN to MAX value as supported by the device.
- b. GET\_CUR to dwPeakBitrate and dwAverageBitrate for wLayerID zero.
- c. Verify the value of dwPeakBitrate and dwAverageBitrate as configured value.
- d. If step (c) fails, fail the test and throw related assertion ([5.1.1.24](#)).
- e. Log and exit the test, as no other layers are supported.

11. If wLayerID is equal to non zero value. (test for wLayerID zero)

- a. SET\_CUR with dwPeakBitrate and dwAverageBitrate for wLayerID zero. The dwPeakBitrate and dwAverageBitrate value should be in the range from MIN to MAX value as supported by the device.
- b. GET\_CUR to dwPeakBitrate and dwAverageBitrate for wLayerID zero.
- c. Verify the value of dwPeakBitrate and dwAverageBitrate as configured value.
- d. If step (c) fails, fail the test and throw related assertion ([5.1.1.24](#)).

12. If wLayerID is equal to valid non zero value. (test for lower than Max layers).

- a. The GET\_MAX values supported by the device for StreamID, QualityID, DependencyID, TemporalID, dwPeakBitrate and dwAverageBitrate.
- b. SET\_CUR with lower than the max valid value of StreamID, QualityID, DependencyID, TemporalID with valid dwPeakBitrate and dwAverageBitrate. If max value is zero for ID, then use zero for that ID. The dwPeakBitrate and dwAverageBitrate value should be in the range from MIN to MAX value as supported by the device.
- c. GET\_CUR for dwPeakBitrate and dwAverageBitrate using StreamID only. The StreamID value used shall be same as used in SET\_CUR.
- d. Verify the value of dwPeakBitrate and dwAverageBitrate as set in step (b).
- e. If step (d) fails, fail the test and throw related assertion ([5.1.1.24](#)).
- f. GET\_CUR for dwPeakBitrate and dwAverageBitrate using QualityID only. The QualityID value used shall be same as used in SET\_CUR.
- g. Verify the value of dwPeakBitrate and dwAverageBitrate as set in step (b).
- h. If step (g) fails, fail the test and throw related assertion ([5.1.1.24](#)).
- i. GET\_CUR for dwPeakBitrate and dwAverageBitrate using DependencyID only. The DependencyID value used shall be same as used in SET\_CUR.

- j. Verify the value of dwPeakBitrate and dwAverageBitrate as set in step (b).
- k. If step (j) fails, fail the test and throw related assertion ([5.1.1.24](#)).
- l. GET\_CUR for dwPeakBitrate and dwAverageBitrate using TemporalID only. The TemporalID value used shall be same as used in SET\_CUR.
- m. Verify the value of dwPeakBitrate and dwAverageBitrate as set in step (b).
- n. If step (m) fails, fail the test and throw related assertion ([5.1.1.24](#)).

#### **6.2.2.6 Device XU Control device reported UVCX\_QP\_STEPS\_LAYERS parameter Tests**

TD 3.5 This test verifies the UVCX\_QP\_STEPS\_LAYERS control of bFrameType, bMinQP and bMaxQP parameters.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU Control UVCX\_QP\_STEPS\_LAYERS defined in specification section 3.3.15.
8. Verify the wLayerID returns the valid value.
9. If wLayerID value is not in the valid range, fail the test and throw related assertion ([5.1.1.25](#)).
10. If wLayerID is equal to zero
  - a. If step (9) is true.

- b. SET\_CUR with bFrameType (bFrameType=0x03, I and P frames), bMinQp = 0x18 and bMaxQp = 0x2A for wLayerID zero.
  - c. GET\_CUR to bMinQp and bMaxQp for wLayerID zero.
  - d. Verify the value of bMinQp and bMaxQp as configured value.
  - e. If step (d) fails, fail the test and throw related assertion ([5.1.1.25](#)).
  - f. Verify the support of B frame by GET\_MAX bFrameType.
  - g. If B frame is not supported, then go to step (11).
  - h. SET\_CUR with bFrameType (bFrameType = 0x04, B frames), bMinQp = 0x18 and bMaxQp=0x2A for wLayerID zero.
  - i. GET\_CUR to bMinQp and bMaxQp for wLayerID zero.
  - j. Verify the value of bMinQp and bMaxQp as configured value.
  - k. If step (j) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- 11. If wLayerID is equal to the valid non zero value.
  - a. If step (11) is true.
  - b. SET\_CUR with bFrameType (bFrameType = 0x03, I and P frames), bMinQp = 0x18 and bMaxQp=0x2A for wLayerID zero.
  - c. GET\_CUR to bMinQp and bMaxQp for wLayerID zero.
  - d. Verify the value of bMinQp and bMaxQp as configured value.
  - e. If step (d) fails, fail the test and throw related assertion ([5.1.1.25](#)).
  - f. Verify the support of B frame by GET\_MAX bFrameType.
  - g. If B frame is not supported, then go to step (12).
  - h. SET\_CUR with bFrameType (bFrameType = 0x04, B frames), bMinQp = 0x18 and bMaxQp = 0x2A for wLayerID zero.
  - i. GET\_CUR to bMinQp and bMaxQp for wLayerID zero.
  - j. Verify the value of bMinQp and bMaxQp as configured value.
  - k. If step (j) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- 12. If wLayerID is equal to the valid non zero value.
  - a. The GET\_MAX values supported by the device for StreamID, QualityID, DependencyID and TemporalID.

- b. SET\_CUR with lower than the maximum valid value of StreamID, QualityID, DependencyID, TemporalID with bFrameType (bFrameType = 0x03, I and P frames), bMinQp = 0x18 and bMaxQp = 0x2A. If max value is zero for ID, then use zero for that ID.
- c. GET\_CUR for bMinQp and bMaxQp using StreamID only. The StreamID value used shall be same as used in SET\_CUR.
- d. Verify the value of bMinQp and bMaxQp as set in step (b).
- e. If step (d) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- f. GET\_CUR for bFrameType = 0x01(I Frame), bMinQp and bMaxQp using QualityID only. The QualityID value used shall be same as used in SET\_CUR.
- g. Verify the value of bMinQp and bMaxQp as set in step (b).
- h. If step (g) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- i. GET\_CUR for bFrameType = 0x02(P Frame), bMinQp and bMaxQp using QualityID only. The QualityID value used shall be same as used in SET\_CUR.
- j. Verify the value of bMinQp and bMaxQp as set in step (b).
- k. If step (j) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- l. GET\_CUR for bFrameType = 0x01(I Frame), bMinQp and bMaxQp using DependencyID only. The DependencyID value used shall be same as used in SET\_CUR.
- m. Verify the value of bMinQp and bMaxQp as set in step (b).
- n. If step (m) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- o. GET\_CUR for bFrameType = 0x02(P Frame), bMinQp and bMaxQp using DependencyID only. The DependencyID value used shall be same as used in SET\_CUR.
- p. Verify the value of bMinQp and bMaxQp as set in step (b).
- q. If step (p) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- r. GET\_CUR for bFrameType = 0x01(I Frame), bMinQp and bMaxQp using TemporalID only. The TemporalID value used shall be same as used in SET\_CUR.
- s. Verify the value of bMinQp and bMaxQp as set in step (b).
- t. If step (s) fails, fail the test and throw related assertion ([5.1.1.25](#)).
- u. GET\_CUR for bFrameType = 0x02(P Frame), bMinQp and bMaxQp using TemporalID only. The TemporalID value used shall be same as used in SET\_CUR.

- v. Verify the value of bMinQp and bMaxQp as set in step (b).
- w. If step (v) fails, fail the test and throw related assertion ([5.1.1.25](#)).

#### **6.2.2.7 Device XU Control device reported UVCX\_PICTURE\_TYPE\_CONTROL parameter Tests**

TD 3.6 This test verifies the UVCX\_PICTURE\_TYPE\_CONTROL control of wPicType.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU Control UVCX\_PICTURE\_TYPE\_CONTROL defined in specification section 3.3.9.
8. Verify the wPicType for the support of valid data (0x00 to 0x02).
9. If step (8) fails, fail the test and throw related assertion ([5.1.1.26](#))

#### **6.2.2.8 Device XU Control device reported UVCX\_FRAMERATE\_CONFIG parameter Tests**

TD 3.7 This test verifies the UVCX\_FRAMERATE\_CONFIG control of dwFrameInterval.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.

3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MIN for XU Control UVCX\_FRAMERATE\_CONFIG defined in specification section 3.3.12.
8. SET\_CUR with doubling the dwFrameInterval from GET\_MIN and set the StreamID (part of wLayerID) with the value of bStreamID used in step (6).
9. GET\_CUR and verify the dwFrameInterval as set in step (8).
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.27](#)).

#### **6.2.2.9 Device XU Control device reported UVCX\_VIDEO\_ADVANCE\_CONFIG parameter Tests**

TD 3.8 This test verifies the UVCX\_VIDEO\_ADVANCE\_CONFIG control of dwMb\_max and blevel\_idc.

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. GET\_MAX for XU control UVCX\_VIDEO\_ADVANCE\_CONFIG.
8. Verify the value of dwMb\_max.
9. If dwMb\_max == 0, fail the test and throw related assertion ([5.1.1.28](#)).
10. SET\_CUR for XU control UVCX\_VIDEO\_ADVANCE\_CONFIG with blevel\_idc = 0x28 and set the StreamID (part of wLayerID) with the value of bStreamID used in step (6).

11. GET\_CUR for XU control UVCX\_VIDEO\_ADVANCE\_CONFIG for blevel\_idc. The device may return the value set in step (10) or device supported value.
12. SET\_CUR for XU control UVCX\_VIDEO\_ADVANCE\_CONFIG with blevel\_idc = 0x1F
13. GET\_CUR for XU control UVCX\_VIDEO\_ADVANCE\_CONFIG for blevel\_idc. The device may return the value set in step (12) or the device supported value.

#### **6.2.2.10 Device XU Control device reported UVCX\_ENCODER\_RESET parameter Tests**

TD 3.9 This test verifies the UVCX\_ENCODER\_RESET control of bStreamID

##### **Overview of Test Steps**

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. Save the default configuration for the next step verification.
6. GET\_MIN for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
7. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE
8. SET\_CUR for XU Control UVCX\_ENCODER\_RESET using the StreamID (part of wLayerID) with the value of bStreamID used in step (6).
9. GET\_CUR for Control UVCX\_VIDEO\_CONFIG\_PROBE.
10. Verify the parameters GET\_CUR and step (5).
11. If step (10) fails, fail the test and throw related assertion ([5.1.1.29](#)).



### 6.2.2.11 Device XU Control device reported UVCX\_LTR\_BUFFER\_SIZE\_CONTROL Test

TD 4.0 This test verifies the UVCX\_LTR\_BUFFER\_CONTROL control bLTRBufferSize and bLTREncoderControl.

#### Overview of Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.1.
5. SET\_CUR for Control UVCX\_LTR\_BUFFER\_CONTROL with bLTRBufferSize=0 and bLTREncoderControl = 0 for wLayerID = 0.
6. GET\_MAX for control UVCX\_LTR\_BUFFER\_CONTROL.
7. Save and Verify the returned max value of bLTRBufferSize.
8. If bLTRBufferSize is equal to zero, Log and exit the test. (means device does not support the external control)
9. SET\_CUR for Control UVCX\_LTR\_BUFFER\_CONTROL with bLTRBufferSize = value less than the max returned value and bLTREncoderControl = bLTRBufferSize-1 for wLayerID = 0.
10. GET\_CUR for Control UVCX\_LTR\_BUFFER\_CONTROL.
11. Verify the values of bLTRBufferSize and bLTREncoderControl as set in the step 9.  
  
The bLTRBufferSize can be lower than the set values.
12. If step 11 fails, fail the test and throw related assertion ([5.1.1.30](#)).

### 6.2.2.12 Device XU Control device reported UVCX\_LTR\_PICTURE\_CONTROL Tests

TD 4.1 This test verifies the UVCX\_LTR\_PICTURE\_CONTROL control bPutAtPositionInLTRBuffer and bEncodeUsingLTR.

### Overview of Test Steps

The test software tool performs the following steps:

1. If test [TD40](#) bLTRBufferSize has more than 0, then proceed with this test.
2. Execute the Init Procedure.
3. Put the device in desired State.
4. Enumerate the device.
5. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
6. SET\_CUR for Control UVCX\_VIDEO\_CONFIG\_COMMIT.
7. SET\_CUR for control UVCX\_LTR\_PICTURE\_CONTROL with bPutAtPositionInLTRBuffer (set as per spec 3.3.8) and bEncodeUsingLTR =1 for wLayerID =0.
8. This should appear in Stream. The stream testing is out of scope of this specification.

## 6.2.3 Static XU Control MIN/MAX tests

### 6.2.3.1 Device XU Control device reported UVCX\_VIDEO\_CONFIG\_PROBE's MIN/MAX Tests

TD 4.2 This test verifies the UVCX\_VIDEO\_CONFIG\_PROBE control's GET\_MIN/GET\_MAX limits.

### Overview of GET\_MIN limit Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MIN for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
5. Save the MIN all parameters.
6. Use GET\_MIN value of dwFrameInterval.
7. Use SET\_CUR with dwFrameInterval-1 (decrement by one. The parameter is related frame rate behavior).

8. Use GET\_CUR and verify the value of dwFrameInterval.
9. If dwFrameInterval is not equal to GET\_MIN value of dwFrameInterval.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.32](#))
11. Use GET\_MIN value of dwBitRate.
12. Use SET\_CUR with dwBitRate -1 (decrement by one if it is not zero).
13. Use GET\_CUR and verify the value of dwBitRate.
14. If dwBitRate is not equal to GET\_MIN value of dwBitRate.
15. If step (14) fails, fail the test and throw related assertion ([5.1.1.32](#))
16. Use GET\_MIN value of wWidth.
17. Use SET\_CUR with wWidth -1 (decrement by one if it is not zero).
18. Use GET\_CUR and verify the value of wWidth.
19. If wWidth is not equal to GET\_MIN value of wWidth.
20. If step (19) fails, fail the test and throw related assertion ([5.1.1.32](#))
21. Use GET\_MIN value of wHeight.
22. Use SET\_CUR with wHeight -1 (decrement by one if it is not zero).
23. Use GET\_CUR and verify the value of wHeight.
24. If wHeight is not equal to GET\_MIN value of wHeight.
25. If step (24) fails, fail the test and throw related assertion ([5.1.1.32](#))
26. Use GET\_MIN value of wFramePeriod.
27. Use SET\_CUR with wFramePeriod -1 (decrement by one if it is not zero).
28. Use GET\_CUR and verify the value of wFramePeriod.
29. If wFramePeriod is not equal to GET\_MIN value of wFramePeriod.
30. If step (29) fails, fail the test and throw related assertion ([5.1.1.32](#))
31. Use GET\_MIN value of wLeakyBucketSize.
32. Use SET\_CUR with wLeakyBucketSize -1 (decrement by one if it is not zero).
33. Use GET\_CUR and verify the value of wLeakyBucketSize.
34. If wLeakyBucketSize is not equal to GET\_MIN value of wLeakyBucketSize.
35. If step (34) fails, fail the test and throw related assertion ([5.1.1.32](#)).

### Overview of GET\_MAX limit Test Steps

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
5. Save the MAX all parameters.
6. Use GET\_MAX value of dwFrameInterval.
7. Use SET\_CUR with dwFrameInterval+1 (increment by one).
8. Use GET\_CUR and verify the value of dwFrameInterval.
9. If dwFrameInterval is not equal to GET\_MAX value of dwFrameInterval.
10. If step (9) fails, fail the test and throw related assertion ([5.1.1.32](#)).
11. Use GET\_MAX value of dwBitRate.
12. Use SET\_CUR with dwBitRate+1 (increment by one).
13. Use GET\_CUR and verify the value of dwBitRate.
14. If dwBitRate is not equal to GET\_MAX value of dwBitRate.
15. If step (14) fails, fail the test and throw related assertion ([5.1.1.32](#)).
16. Use GET\_MAX value of wWidth.
17. Use SET\_CUR with wWidth +1 (increment by one).
18. Use GET\_CUR and verify the value of wWidth.
19. If wWidth is not equal to GET\_MAX value of wWidth.
20. If step (19) fails, fail the test and throw related assertion ([5.1.1.32](#)).
21. Use GET\_MAX value of wHeight.
22. Use SET\_CUR with wHeight +1 (increment by one).
23. Use GET\_CUR and verify the value of wHeight.

24. If wHeight is not equal to GET\_MAX value of wHeight.
25. If step (24) fails, fail the test and throw related assertion ([5.1.1.32](#)).
26. Use GET\_MAX value of wFramePeriod.
27. Use SET\_CUR with wFramePeriod +1 (increment by one).
28. Use GET\_CUR and verify the value of wFramePeriod.
29. If wFramePeriod is not equal to GET\_MAX value of wFramePeriod.
30. If step (29) fails, fail the test and throw related assertion ([5.1.1.32](#)).
31. Use GET\_MAX value of bTemporalScaleMode.
32. Use SET\_CUR with bTemporalScaleMode +1 (increment by one).
33. Use GET\_CUR and verify the value of bTemporalScaleMode.
34. If bTemporalScaleMode is not equal to GET\_MAX value of bTemporalScaleMode.
35. If step (34) fails, fail the test and throw related assertion ([5.1.1.32](#)).
36. Use GET\_MAX value of bSpatialScaleMode.
37. Use SET\_CUR with bSpatialScaleMode +1 (increment by one).
38. Use GET\_CUR and verify the value of bSpatialScaleMode.
39. If bSpatialScaleMode is not equal to GET\_MAX value of bSpatialScaleMode.
40. If step (39) fails, fail the test and throw related assertion ([5.1.1.32](#)).
41. Use GET\_MAX value of bSNRScaleMode.
42. Use SET\_CUR with bSNRScaleMode +1 (increment by one).
43. Use GET\_CUR and verify the value of bSNRScaleMode.
44. If bSNRScaleMode is not equal to GET\_MAX value of bSNRScaleMode.
45. If step (44) fails, fail the test and throw related assertion ([5.1.1.32](#)).
46. Use GET\_MAX value of bNumOfReorderFrames.
47. Use SET\_CUR with bNumOfReorderFrames +1 (increment by one).
48. Use GET\_CUR and verify the value of bNumOfReorderFrames.
49. If bNumOfReorderFrames is not equal to GET\_MAX value of bNumOfReorderFrames.
50. If step (49) fails, fail the test and throw related assertion ([5.1.1.32](#)).
51. Use GET\_MAX value of bView.

52. Use SET\_CUR with bView +1 (increment by one).
53. Use GET\_CUR and verify the value of bView.
54. If bView is not equal to GET\_MAX value of bView.
55. If step (54) fails, fail the test and throw related assertion ([5.1.1.32](#)).
56. Use GET\_MAX value of wLeakyBucketSize.
57. Use SET\_CUR with wLeakyBucketSize +1 (increment by one).
58. Use GET\_CUR and verify the value of wLeakyBucketSize.
59. If wLeakyBucketSize is not equal to GET\_MAX value of wLeakyBucketSize.
60. If step (59) fails, fail the test and throw related assertion ([5.1.1.32](#)).

#### 6.2.4 Compliance test with the combination of parameters settings

TD 4.3 This test verifies the UVCX\_VIDEO\_CONFIG\_PROBE control's for the combination of parameters. The combinations of parameters are resolution, frame interval and profile.

The test software tool performs the following steps:

1. Execute the Init Procedure.
2. Put the device in desired State.
3. Enumerate the device.
4. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
5. Verify the bTemporalScaleMode, bSpatialScaleMode and bSNRScaleMode.
6. If (bTemporalScaleMode==0, bSpatialScaleMode==0 and bSNRScaleMode==0)
7. If step (6) is true.
  - a. SET\_CUR with bSpatialScaleMode=2 (testing for the device error).
  - b. GET\_CUR and verify the wWidth and wHeight parameters returned by the device.
  - c. If wWidth and/or wHeight are not equal to zero, fail the test and throw related assertion ([5.1.1.33](#)).

- d. GET\_MAX for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
  - e. Save the GET\_MAX configuration for next step use.
  - f. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
  - g. Check the value of wWidth and wHeight.
  - h. If wWidth and wHeight parameters are not equal to GET\_MAX values, then change them to GET\_MAX values.
  - i. Change the wProfile parameter to Constrained Baseline.
  - j. Change the dwFrameInterval parameter to GET\_MAX values. (The highest frame rate).
  - k. SET\_CUR with changes of step (h), (i) and (j).
  - l. GET\_CUR and verify the values of wWidth, wHeight, wProfile and dwFrameInterval.
  - m. If parameters wWidth and wHeight are not equal to zero, then check the remaining parameters wProfile and dwFrameInterval for acceptable valid range.
  - n. If parameters are not valid, then fail the test and throw related assertion ([5.1.1.33](#)).
  - o. Log and exit the test.
8. If step 6 has any non zero modes
- a. Save the GET\_MAX parameters to use in next steps.
  - b. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
  - c. Change following parameters:
    - I. dwFrameInterval to GET\_MAX values.
    - II. wWidth to GET\_MAX values.
    - III. wHeight to GET\_MAX values.
    - IV. wProfile equal to Scalable High Profile.
    - V. If bTemporalScaleMode is not equal to zero in GET\_MAX, then change the bTemporalScaleMode parameter to 2.
    - VI. If bSpatialScaleMode is not equal to zero in GET\_MAX, then change the parameter bSpatialScaleMode to 2.

- VII. bSNRScaleMode change to zero.
    - VIII. bStreamMuxOption change to 0x03 (enable auxiliary stream and embed H.264 stream).
    - IX. bStreamID change to zero ( Single H.264 stream)
  - d. SET\_CUR with above step (c) changes.
  - e. GET\_CUR
  - f. If parameters wWidth and wHeight are not equal to zero, then check the remaining parameters wProfile and dwFrameInterval for acceptable valid range.
  - g. If parameters are not valid, then fail the test and throw related assertion ([5.1.1.33](#)).
- 9. If step (6) has non zero.
  - a. Save the GET\_MAX parameters to use in next steps
  - b. GET\_DEF for XU Control UVCX\_VIDEO\_CONFIG\_PROBE defined in specification section 3.3.2.
  - c. Change following parameters:
    - I. dwFrameInterval to GET\_MAX values.
    - II. wWidth to GET\_MAX values.
    - III. wHeight to GET\_MAX values.
    - IV. wProfile equal to Constrained baseline Profile.
    - V. If bTemporalScaleMode is not equal to zero in GET\_MAX, then change the bTemporalScaleMode parameter to 2.
    - VI. If bSpatialScaleMode is not equal to zero in GET\_MAX, then change the parameter bSpatialScaleMode to 2.
    - VII. bSNRScaleMode change to zero.
    - VIII. bStreamMuxOption change to 0x03 (enable auxiliary stream and embed H.264 stream).
    - IX. bStreamID change to zero ( Single H.264 stream)
  - d. SET\_CUR with above step (c) changes.
  - e. GET\_CUR
  - f. If parameters wWidth and/or wHeight are not equal to zero, then check the remaining parameters wProfile, bTemporalScaleMode and bSpatialScaleMode for acceptable valid range.



(if wProfile is equal to Constrained baseline Profile, then  
bTemporalScaleMode and bSpatialScaleMode shall be equal to zero)

- g. If parameters are not valid, then fail the test and throw related assertion ([5.1.1.33](#)).
- h. Log and exit the test.