Xger:2.58

API Reference

X Platform, X5 HEVC SDI 1.0.2

23/04/2026

appear

TABLE OF CONTENTS

1 Overview

Changelog

2.58

- **Added**
  - codedVideo version 1.0
- **Changed**
  - audioStatus from 2.16 to 2.17
  - serviceStatus from 2.47 to 2.48
  - cardStatus from 1.9 to 1.10
  - videoProfile from 2.13 to 2.14
  - coderService from 2.46 to 2.47
  - multiService from 2.18 to 2.19
  - coderConfig from 2.48 to 2.49
  - multiServiceProfile from 2.8 to 2.9
  - expectedInput from 1.7 to 1.8
  - cardAllocation from 1.10 to 1.11
  - videoStatus from 2.17 to 2.18

2.57

- **Changed**
  - coderService from 2.45 to 2.46
  - multiService from 2.17 to 2.18
  - coderConfig from 2.47 to 2.48
  - sdiStatus from 1.11 to 1.12
  - serviceStatus from 2.46 to 2.47
  - sdiVideoModes 1.0 moved from sdiStatus

2.56

- **Changed**
  - coderConfig from 2.46 to 2.47
  - coderService from 2.44 to 2.45
  - multiService from 2.16 to 2.17
  - audioStatus from 2.15 to 2.16
  - multiServiceProfile from 2.7 to 2.8
  - serviceStatus from 2.45 to 2.46
  - st2110Status from 1.5 to 1.6
  - audioProfile from 2.6 to 2.7
  - colorProfile from 2.2 to 2.3

  - `colorComponent` from 1.1 to 1.2
  - `videoStatus` from 2.16 to 2.17
  - `videoUsability` from 1.1 to 1.2
  - `videoProfile` from 2.12 to 2.13
  - `cardAllocation` from 1.9 to 1.10
  - `expectedInput` from 1.6 to 1.7

2.55

- **Changed**
  - `cardAllocation` from 1.8 to 1.9
  - `coderService` from 2.43 to 2.44
  - `coderConfig` from 2.45 to 2.46
  - `expectedInput` from 1.5 to 1.6
  - `multiServiceProfile` from 2.6 to 2.7
  - `multiService` from 2.15 to 2.16
  - `serviceStatus` from 2.44 to 2.45
  - `videoProfile` from 2.11 to 2.12
  - `videoStatus` from 2.15 to 2.16
  - `timecodeConfig` from 1.0 to 1.1
  - `scte35Config` from 1.0 to 1.1

2.54

- **Changed**
  - `coderConfig` from 2.44 to 2.45
  - `coderService` from 2.42 to 2.43
  - `multiService` from 2.14 to 2.15
  - `lockStatus` from 1.4 to 1.5
  - `serviceStatus` from 2.43 to 2.44
  - `poolCtrlAllocs` from 1.1 to 1.2

2.53

- **Changed**
  - `coderService` from 2.41 to 2.42
  - `coderConfig` from 2.43 to 2.44
  - `multiService` from 2.13 to 2.14
  - `lockStatus` from 1.3 to 1.4
  - `serviceStatus` from 2.42 to 2.43
  - `codedAudio` from 1.0 to 1.1
  - `rawAudio` from 1.1 to 1.2

  - audioProfile from 2.5 to 2.6
  - audioStatus from 2.14 to 2.15
  - multiServiceProfile from 2.5 to 2.6
  - st2110Status from 1.4 to 1.5
  - sdiStatus from 1.10 to 1.11

2.52

- **Changed**
  - rawVideo from 1.4 to 1.5
  - videoProfile from 2.10 to 2.11
  - multiServiceProfile from 2.4 to 2.5
  - multiService from 2.12 to 2.13
  - coderService from 2.40 to 2.41
  - videoStatus from 2.14 to 2.15
  - serviceStatus from 2.41 to 2.42
  - expectedInput from 1.4 to 1.5
  - cardAllocation from 1.7 to 1.8
  - nmosStatusTypes from 1.1 to 1.2
  - cardStatus from 1.8 to 1.9
  - coderConfig from 2.42 to 2.43
  - vancProfile from 1.6 to 1.7
  - vancStatus from 1.12 to 1.13
- **Added**
  - dpi version 1.0
  - dpiStatus version 1.0

2.51

- **Changed**
  - videoProfile from 2.9 to 2.10
  - coderService from 2.39 to 2.40
  - multiService from 2.11 to 2.12
  - multiServiceProfile from 2.3 to 2.4
  - cardAllocation from 1.6 to 1.7
  - videoStatus from 2.13 to 2.14
  - expectedInput from 1.3 to 1.4
  - serviceStatus from 2.40 to 2.41

2.50

- **Added**

  - `vpidInfo` version 1.0
  - `videoUsabilityInfo` version 1.1
- **Changed**
  - `coderService` from 2.38 to 2.39
  - `colorProfile` from 2.1 to 2.2
  - `colorComponent` from 1.0 to 1.1
  - `videoStatus` from 2.12 to 2.13
  - `serviceStatus` from 2.39 to 2.40
  - `sdiStatus` from 1.9 to 1.10
  - `multiService` from 2.10 to 2.11
  - `vancProfile` from 1.5 to 1.6
  - `vancStatus` from 1.11 to 1.12
  - `subtitling` from 1.1 to 1.2
  - `tsSource` from 1.4 to 1.5

2.49

No changes

2.48

- **Changed**
  - `audioProfile` from 2.4 to 2.5
  - `audioStatus` from 2.13 to 2.14
  - `coderConfig` from 2.39 to 2.40
  - `coderService` from 2.37 to 2.38
  - `multiService` from 2.9 to 2.10
  - `multiServiceProfile` from 2.2 to 2.3
  - `serviceStatus` from 2.38 to 2.39
  - `st2110Status` from 1.3 to 1.4
  - `vancStatus` from 1.10 to 1.11

2.47

- **Added**
  - 'scte35LogApi' version 1.0
  - sdiRouterConfig version 1.1
- **Changed**
  - `vancStatus` from 1.9 to 1.10
  - `serviceStatus` from 2.37 to 2.38

2.46

- **Added**

  - nmosStatusTypes version 1.1

- **Changed**
  - nmosConfig from 1.0 to 1.1
  - cardConfig from 1.2 to 1.3
  - videoProfile from 2.8 to 2.9
  - expectedInput from 1.2 to 1.3
  - multiServiceProfile from 2.1 to 2.2
  - cardAllocation from 1.5 to 1.6
  - serviceStatus from 2.36 to 2.37
  - videoStatus from 2.11 to 2.12
  - coderService from 2.36 to 2.37
  - multiService from 2.8 to 2.9
  - cardStatus from 1.7 to 1.8

2.45

- **Changed**
  - multiService from 2.7 to 2.8
  - coderService from 2.35 to 2.36
  - tsSource from 1.3 to 1.4
  - serviceStatus from 2.35 to 2.36
  - vancStatus from 1.8 to 1.9

2.44

- **Changed**
  - multiService from 2.6 to 2.7
  - coderService from 2.34 to 2.35
  - tsSource from 1.2 to 1.3
  - ipStream from 1.2 to 1.3
  - st2110Types from 1.0 to 1.2
  - serviceStatus from 2.34 to 2.35
  - st2110Status from 1.2 to 1.3
  - ipConnection from 1.4 to 1.5
  - ipService from 1.1 to 1.2
  - rawAudio from 1.0 to 1.1
  - audioProfile from 2.3 to 2.4
  - multiServiceProfile from 2.0 to 2.1
  - audioStatus from 2.12 to 2.13
  - vancStatus from 1.7 to 1.8

  - vancProfile from 1.4 to 1.5

2.43

- **Added**
  - logoInsertion 1.0
- **Changed**
  - coderService from 2.33 to 2.34
  - multiService from 2.5 to 2.6
  - audioStatus from 2.11 to 2.12
  - sdiStatus from 1.8 to 1.9
  - serviceStatus from 2.33 to 2.34

2.42

- **Changed**
  - serviceStatus from 2.32 to 2.33
  - vancStatus from 1.6 to 1.7
  - cardConfig from 1.1 to 1.2
- **Added**
  - nmosConfig 1.0
  - nmosLabelConfig 1.0

2.41

- **Changed**
  - vancProfile from 1.3 to 1.4
  - coderService from 2.32 to 2.33
  - vancStatus from 1.5 to 1.6
  - multiService from 2.4 to 2.5
  - serviceStatus from 2.31 to 2.32

2.40

- **Changed**
  - coderService from 2.31 to 2.32
  - multiService from 2.3 to 2.4
  - ipInterface from 1.4 to 1.5
  - slotIpInterface from 1.4 to 1.5
  - linkModes from 1.1 to 1.2

2.39

- **Added**
  - poisServerStatus version 1.0
- **Changed**

  - poolCtrlAllocs from 1.0 to 1.1
  - lockStatus from 1.2 to 1.3
  - coderService from 2.30 to 2.31
  - serviceStatus from 2.30 to 2.31
  - multiService from 2.2 to 2.3

2.38

- **Added**
  - esamConfig version 1.0
- **Changed**
  - multiService from 2.1 to 2.2
  - poolConfig from 1.1 to 1.2
  - coderService from 2.29 to 2.30

2.37

- **Changed**
  - coderService from 2.28 to 2.29
  - multiService from 2.0 to 2.1
  - dejitterTypes from 1.0 to 1.1
  - ipConnection from 1.3 to 1.4
  - ipService from 1.0 to 1.1
  - videoStatus from 2.10 to 2.11
  - serviceStatus from 2.29 to 2.30
  - psiTypes from 1.5 to 1.6
  - psiStatus from 1.5 to 1.6
  - inputRedundancyStatus from 1.2 to 1.3

2.36

- **Changed**
  - coderService from 2.27 to 2.28
  - multiServiceProfile from 1.10 to 2.0
  - multiService from 1.22 to 2.0
  - ipConnection from 1.2 to 1.3
  - serviceStatus from 2.28 to 2.29
  - sdpUpload from 1.0 to 1.1
- **Added**
  - ipStreamUtils version 0.0
  - ipService version 1.0

2.35

- **Changed**
  - `coderService` from 2.26 to 2.27
  - `multiService` from 1.21 to 1.22
  - `sdiStatus` from 1.7 to 1.8
  - `serviceStatus` from 2.27 to 2.28

2.34

- **Changed**
  - `redundancyGroupStatus` from 1.1 to 1.2;

2.33

- **Added**
  - `slateInsertion` 1.0
  - `imageUpload` 1.0
- **Changed**
  - `multiService` from 1.20 to 1.21
  - `coderService` from 2.25 to 2.26

2 afd (1.0)

2.1 Type Reference

2.1.1 AfdCode

enum

UNKNOWN
RESERVED_1
BOX_16_9_TOP
BOX_14_9_TOP
BOX_16_9_CENTRE
RESERVED_5
RESERVED_6
RESERVED_7
SAME
_4_3_CENTRE
_16_9_CENTRE
_14_9_CENTRE
RESERVED_12
_4_3_PROTECT_14_9_CENTRE
_16_9_PROTECT_14_9_CENTRE
_16_9_PROTECT_4_3_CENTRE

3 alternateInputStatus (1.4)

3.1 Type Reference

3.1.1 AlternateInputStatus

- When alternative input is used, this struct holds the lists of IpStreamStatus of both input paths.
- It also holds the status of the alternative or seamless input configuration.
-

struct

path1	**list** of **ipStreamStatus.IpStreamStatus** Status of the IpStreams on path 1. •
path2	**list** of **ipStreamStatus.IpStreamStatus** Status of the IpStreams on path 2. •
seamlessType	**list** of **SeamlessTypeStatus** list of statuses of the detected type of seamless input.

3.1.2 NearSeamlessStatus

Status of a near seamless input.

struct

active	**Path** The currently active path. Only used for near seamless.
synchronized	**bool** True if the inputs are synchronized.

3.1.3 Path

The path of a near seamless input which a status describes.

enum

path1
path2

3.1.4 SeamlessTypeStatus

Status of a SMPTE 2022-7 Seamless input or a near seamless input.

variant

smpte2022_7	**Smpte2022_7Status** Status of a SMPTE 2022-7 Seamless input.
	NearSeamlessStatus

`nearSeamless`	Status of a near seamless input.

3.1.5 Smpte2022_7Status

Status of a SMPTE 2022-7 Seamless input.

struct

`id`	**UUID** The id of the IpStream this status represents.
`synchronized`	**bool** True if the inputs are synchronized.
`relativeDelay`	**int** Delay between path A and path B in milliseconds.
`margin`	**int** Difference between the delay buffer size and the current relative delay in milliseconds. It is what remains of the buffer.
`maxRelativeDelay`	**int** The size of the delay buffer in milliseconds.

4 aspectRatio (1.0)

4.1 Type Reference

4.1.1 AspectRatioStandard

enum

```
SMPTE_2016
WSS
VIDEO_INDEX
```

5 audioProfile (2.7)

5.1 Overview

Changelog

2.7

- **Added**
  - `surroundmixlev` and `heightmixlev` to `DdppMultiChannelParameters` for overhead speakers support
  - `loudness` to `DdppEncManualMd` and `DdppTransMd`
  - `DolbyAudioLoudness`
  - `DolbyAudioLoudnessLevelerMode`
  - `DolbyAudioLoudnessManualMode`
- **Changed**
  - Removed `dsurmod`. Deprecated in latest Dolby Audio SDK; forced to 'NotIndicated'
  - Removed `DdppStereoParameters` as it's now empty without 'dsurmod'
  - Removed `DdppStereoParameters` from `DdppChannelModeParameters`
  - Removed `lfelpfon`, `suratton` and `dsurexmod` from `DdppMultiChannelParameters`. Deprecated in latest Dolby Audio SDK; forced to 'true', 'false' and 'NotIndicated' respectively
  - Removed `mixlevel`, `roomtyp`, `copyrightb`, `origbs` and `adconvtyp` from `DdppEncManualMd`. Deprecated in latest Dolby Audio SDK; forced to '80', 'NotIndicated', 'true', 'true' and 'Standard' respectively

2.6

- **Changed**
  - `codedAudio` from 1.0 to 1.1
  - `rawAudio` from 1.1 to 1.2

2.5

- **Added**
  - followInputChannelMode as optional(bool) in `AudioProfileStruct`

2.4

- **Changed**
  - `rawAudio` from 1.0 to 1.1

5.2 Command Reference

5.2.1 GetAudioProfiles

- **message GetAudioProfiles.Request**
- **message GetAudioProfiles.Response**
- **message GetAudioProfiles.Failure**

5.2.2 SetAudioProfiles

- message SetAudioProfiles.Request
- message SetAudioProfiles.Response
- message SetAudioProfiles.Failure

5.2.3 DeleteAudioProfiles

- message DeleteAudioProfiles.Request
- message DeleteAudioProfiles.Response
- message DeleteAudioProfiles.Failure

5.3 Type Reference

5.3.1 AacContainer

enum

ADTS
LATM

5.3.2 AacLcParameters

struct

container	AacContainer

5.3.3 AudioProfile

AudioProfileStruct

5.3.4 AudioProfileStruct

struct

label	string
codec	codedAudio.AudioCodec
channelMode	rawAudio.ChannelMode
followInputChannelMode	optional bool
sampleRate	SampleRate
bitRate	int
cparams	CodecParameters

5.3.5 CodecParameters

variant

mpeg1l2	Mpeg1l2Parameters
aac	AacLcParameters
dd	DolbyDigitalPlusParameters

5.3.6 CompProfileAndDialNorm

struct

dialnorm	**int**
compProfile	**DdppCompressionProfile**

5.3.7 DdppAdConverterType

enum

Undefined
Standard
HDCD

5.3.8 DdppBitstreamMode

enum

Undefined
CompleteMain
MusicAndEffects
VisuallyImpaired
HearingImpaired
Dialogue
Commentary
Emergency
VoiceOver
Karaoke

5.3.9 DdppChannelModeParameters

variant

mono	**DdppChannelModeParameters.mono**
stereo	**DdppChannelModeParameters.stereo**
multi	**DdppMultiChannelParameters**

5.3.10 DdppChannelModeParameters.mono

empty **struct**

5.3.11 DdppChannelModeParameters.stereo

empty **struct**

5.3.12 DdppCompressionProfile

struct

dynrng	DdppDrcProfile
compr	DdppDrcProfile

5.3.13 DdppDownmixLevel

enum

Undefined
Pos_3Db
Pos_1_5Db
Pos_0Db
Neg_1_5Db
Neg_3Db
Neg_4_5Db
Neg_6Db
NegInfDb
Neg_9Db
Neg_12Db

5.3.14 DdppDownmixMode

enum

Undefined
NotIndicated
LtRt
LoRo
ProLogicII

5.3.15 DdppDrcProfile

enum

Undefined
None
FilmStandard
FilmLight
MusicStandard
MusicLight
Speech

5.3.16 DdppEncExtMd

struct

extMetadataProgramNum	int
extMetadataRevMode	DdppExtMetadataReversionMode
extMetadataOverwriteMode	DdppExtMetadataOverwriteMode

5.3.17 DdppEncManualMd

struct

dialnorm	**int**
bsmod	**DdppBitstreamMode**
compProfile	**DdppCompressionProfile**
evolutionFrameworkEnabled	**bool**
cmParams	**DdppChannelModeParameters**
loudness	**optional DolbyAudioLoudness**

5.3.18 DdppExtMetadataOverwriteMode

variant

none	**DdppExtMetadataOverwriteMode.none**
dialnorm	**int**
compProfile	**DdppCompressionProfile**
compProfileAndDialNorm	**CompProfileAndDialNorm**

5.3.19 DdppExtMetadataOverwriteMode.none

empty **struct**

5.3.20 DdppExtMetadataOverwriteModeEnum

enum

NONE
DIALNORM
COMPPROFILE
COMPPROFILEANDDIALNORM

5.3.21 DdppExtMetadataReversionMode

enum

Undefined
LastUsedMetadata
LastHostSettings

5.3.22 DdppHeadphoneMode

enum

Undefined
NotIndicated
Disabled
Enabled

5.3.23 DdppMetadata

variant

encManualMd	DdppEncManualMd
encExtMd	DdppEncExtMd
transMd	DdppTransMd

5.3.24 DdppMetadataEnum

enum

DDPPENCMANUALMD
DDPPENCEXTMD
DDPPTRANSMD

5.3.25 DdppMultiChannelParameters

struct

sur90on	bool
lorocmixlev	DdppDownmixLevel
lorosurmixlev	DdppDownmixLevel
ltrtcmixlev	DdppDownmixLevel
ltrtsurmixlev	DdppDownmixLevel
surroundmixlev	DdppDownmixLevel
heightmixlev	DdppDownmixLevel
dmixmod	DdppDownmixMode

5.3.26 DdppRoomType

enum

Undefined
NotIndicated
LargeRoom
SmallRoom

5.3.27 DdppSurroundMode

enum

Undefined
NotIndicated
Disabled
Enabled

5.3.28 DdppTransMd

struct

dialnorm	int
evolutionFrameworkEnabled	bool
loudness	optional DolbyAudioLoudness

5.3.29 DeleteAudioProfiles.Failure

empty struct

5.3.30 DeleteAudioProfiles.Request

struct

ids	list of UUID

5.3.31 DeleteAudioProfiles.Response

empty struct

5.3.32 DolbyAudioLoudness

struct

levelerMode	DolbyAudioLoudnessLevelerMode
regulationType	DolbyAudioLoudnessRegulationType

5.3.33 DolbyAudioLoudnessAribTrB32Mode

empty struct

5.3.34 DolbyAudioLoudnessAtscA85Mode

empty struct

5.3.35 DolbyAudioLoudnessEbuR128Mode

empty struct

5.3.36 DolbyAudioLoudnessFreeTvOp59Mode

empty struct

5.3.37 DolbyAudioLoudnessLevelerMode

enum

ON
AUTO

5.3.38 DolbyAudioLoudnessManualMode

Manual mode for Dolby Audio Loudness.

struct

`peakLimit`	**int** Maximum output level to prevent signal clipping or distortion. Has unit dBTP/10.
`dialogueNormalization`	**int** Metadata value that sets playback gain so dialogue stays at a consistent level.
`dialogueIntelligence`	**bool** Measures only the dialogue portions of a program to ensure consistent loudness.

5.3.39 DolbyAudioLoudnessRegulationType

variant

`ebu_r128`	**DolbyAudioLoudnessEbuR128Mode**
`atsc_a85`	**DolbyAudioLoudnessAtscA85Mode**
`freetv_op59`	**DolbyAudioLoudnessFreeTvOp59Mode**
`arib_trb32`	**DolbyAudioLoudnessAribTrB32Mode**
`manual`	**DolbyAudioLoudnessManualMode**

5.3.40 DolbyAudioLoudnessRegulationTypeEnum

enum

`EbuR128`
`AtscA85`
`FreeTvOp59`
`AribTrB32`
`Manual`

5.3.41 DolbyDigitalPlusParameters

struct

`metadata`	**DdppMetadata**

5.3.42 GetAudioProfiles.Failure

empty **struct**

5.3.43 GetAudioProfiles.Request

empty **struct**

5.3.44 GetAudioProfiles.Response

struct

data	**map** from **UUID** to **AudioProfile**
info	**GetAudioProfiles.Response.info**

5.3.45 GetAudioProfiles.Response.info

struct

totalEntries	**int**

5.3.46 Mpeg1l2Parameters

empty **struct**

5.3.47 SampleBitDepth

enum

SBD_16
SBD_20
SBD_24

5.3.48 SampleRate

enum

SR_48kHz

5.3.49 SetAudioProfiles.Failure

empty **struct**

5.3.50 SetAudioProfiles.Request

struct

data	**map** from **UUID** to **AudioProfile**

5.3.51 SetAudioProfiles.Response

empty **struct**

6 audioStatus (2.17)

6.1 Overview

Changelog

2.17

- **Added**
  - `DolbyAudioLoudnessCorrectionTypeEnum` enum for signalling correction type in Dolby audio loudness status.
  - `correctionType` to `DolbyAudioLoudnessStatus`.
  - `programDescription` to `DepMetadata`.
  - `numCrcErrors` to `DdppMetadata`.

2.16

- **Changed**
  - `audioProfile` from 2.6 to 2.7
  - added optional `surroundmixlev` and `heightmixlev` to `DdppEnc`
  - added optional `surroundmixlev` and `heightmixlev` to `DdppMetadata`
  - added optional 'DolbyAudioLoudnessStatus' to `DdppEnc` and `DdppMetadata`

2.15

- **Changed**
  - `audioProfile` from 2.5 to 2.6
  - `codedAudio` from 1.0 to 1.1
  - `rawAudio` from 1.1 to 1.2

2.14

- **Changed**
  - `audioProfile` from 2.4 to 2.5

2.13

- **Changed**
  - `rawAudio` from 1.0 to 1.1
  - `audioProfile` from 2.3 to 2.4

2.12

- **Changed**
  - `embeddedChannel` to `embeddedChannels` in `BasicConfig`; now a list of all channels in use and not just the starting index.

2.11

- **Changed**
  - `component` from 1.3 to 1.5

6.2 Type Reference

6.2.1 Audio

struct

channelId	**int**
uid	**int**
source	**Source**
output	**Output**
incorrectDspSpeed	**optional bool**

6.2.2 AudioSet

struct

main	**Audio**
backup	**optional Audio**
active	**Input**

6.2.3 AudioStatus

struct

audios	**list** of **AudioSet**

6.2.4 BasicConfig

struct

codec	**optional codedAudio.AudioCodec**
componentType	**optional component.ComponentType**
sampleRate	**optional audioProfile.SampleRate**
bitRate	**optional int**
channelMode	**optional rawAudio.ChannelMode**
numChannels	**optional int**
aacContainer	**optional audioProfile.AacContainer**
embeddedChannels	**optional list** of **int**

6.2.5 CondensedAudio

struct

channelId	**int**
uid	**int**
source	**CondensedAudio.source**
output	**CondensedAudio.output**

6.2.6 CondensedAudio.output

struct

codec	optional codedAudio.AudioCodec
channelMode	optional rawAudio.ChannelMode
bitRate	optional int
ddpp	optional CondensedAudio.output.ddpp

6.2.7 CondensedAudio.source

struct

addBsiByte0	optional int

6.2.8 CondensedAudioStatus

struct

audios	list of CondensedAudio

6.2.9 DdppAcMod

enum

_1_0
_1_1
_2_0
_2_1
_2_2
_3_0
_3_1
_3_2
_3_3
_3_4

6.2.10 DdppDec

struct

outputchannelconfig	rawAudio.ChannelMode
drcmode	DdppDecDrcMode
drchighcut	DdppDecDrcScale
drclowboost	DdppDecDrcScale
stereodownmixmode	DdppDecStereoDownmixMode

6.2.11 DdppDecDrcMode

enum

Undefined
Custom_0
Line
Rf
portable_Neg_8_Db
portable_Neg_11_Db
portable_Neg_14_Db

6.2.12 DdppDecDrcScale

enum

Undefined
Min
Max

6.2.13 DdppDecStereoDownmixMode

enum

Undefined
Auto
LtRt
LoRo

6.2.14 DdppEnc

struct

dialnorm	**int**
bsmod	**audioProfile.DdppBitstreamMode**
dynrng	**audioProfile.DdppDrcProfile**
compr	**audioProfile.DdppDrcProfile**
dsurmod	**audioProfile.DdppSurroundMode**
mixlevel	**int**
roomtyp	**audioProfile.DdppRoomType**
copyrightb	**YesNo**
origbs	**YesNo**
lorocmixlev	**audioProfile.DdppDownmixLevel**
lorosurmixlev	**audioProfile.DdppDownmixLevel**
ltrtcmixlev	**audioProfile.DdppDownmixLevel**
ltrtsurmixlev	**audioProfile.DdppDownmixLevel**
surroundmixlev	**optional audioProfile.DdppDownmixLevel**
heightmixlev	**optional audioProfile.DdppDownmixLevel**
dmixmod	**audioProfile.DdppDownmixMode**
dsurexmod	**audioProfile.DdppSurroundMode**
dheadphonmod	**audioProfile.DdppHeadphoneMode**
adconvtyp	**audioProfile.DdppAdConverterType**
lfelpfon	**OnOff**

sur90on	OnOff
suratton	OnOff
evolutionFrameworkEnabled	OnOff
loudness	optional DolbyAudioLoudnessStatus

6.2.15 DdppMetadata

struct

numCrcErrors	optional int
acmod	DdppAcMod
lfeon	OnOff
datarate	optional int
samplerate	optional int
dialnorm	int
dialnorm2	optional int
bsmod	audioProfile.DdppBitstreamMode
dsurmod	audioProfile.DdppSurroundMode
mixlevel	optional int
roomtyp	optional audioProfile.DdppRoomType
copyrightb	YesNo
origbs	YesNo
cmixlev	optional audioProfile.DdppDownmixLevel
surmixlev	optional audioProfile.DdppDownmixLevel
lorocmixlev	audioProfile.DdppDownmixLevel
lorosurmixlev	audioProfile.DdppDownmixLevel
ltrtcmixlev	audioProfile.DdppDownmixLevel
ltrtsurmixlev	audioProfile.DdppDownmixLevel
surroundmixlev	optional audioProfile.DdppDownmixLevel
heightmixlev	optional audioProfile.DdppDownmixLevel
dmixmod	audioProfile.DdppDownmixMode
dsurexmod	audioProfile.DdppSurroundMode
dheadphonmod	optional audioProfile.DdppHeadphoneMode
adconvtyp	audioProfile.DdppAdConverterType
adconvtyp2	optional audioProfile.DdppAdConverterType
addbsie	optional YesNo
addbsibyte0	optional int
loudness	optional DolbyAudioLoudnessStatus

6.2.16 DepMetadata

struct

frameRate	float
numberOfPrograms	int
programConfig	DolbyEProgramConfig
bitDepth	int
programSelect	optional int
programDescription	optional string

6.2.17 DolbyAudioLoudnessCorrectionTypeEnum

enum

Undefined
RealTime
FileBased

6.2.18 DolbyAudioLoudnessLevelerModeStatus

enum

OFF
ON
AUTO

6.2.19 DolbyAudioLoudnessStatus

Dolby Audio Loudness Status

struct

levelerMode	**DolbyAudioLoudnessLevelerModeStatus** Leveler mode being used, OFF/ON/AUTO.
regulationType	**optional audioProfile.DolbyAudioLoudnessRegulationTypeEnum** Regulation type being used.
peakLimit	**optional int** The configured true peak limit for the encoded audio. Has unit dBTP/10.
dialogueNormalization	**optional int** Configured playback gain for consistent dialogue level.
dialogueIntelligence	**optional YesNo** If only the dialogue portions of a program is measured prior to leveling.
correctionType	**optional DolbyAudioLoudnessCorrectionTypeEnum** Loudness correction type: real-time or file-based (if available).

6.2.20 DolbyDigitalPlusMetEvoStatus

enum

Nofound
ErrorParse
ErrorPcmhash
ErrorSerialize
valid

6.2.21 DolbyEProgramConfig

enum

_51_2
_51_1_1
_4_4
_4_2_2
_4_2_1_1
_4_1_1_1_1
_2_2_2_2
_2_2_2_1_1
_2_2_1_1_1_1
_2_1_1_1_1_1_1
_1_1_1_1_1_1_1_1
_51
_4_2
_4_1_1
_2_2_2
_2_2_1_1
_2_1_1_1_1
_1_1_1_1_1_1
_4
_2_2
_2_1_1
_1_1_1_1
_71
_71_Screen

6.2.22 EvolutionFrameworkMetadata

struct

evoPresent	**YesNo**
evoStatus	**DolbyDigitalPlusMetEvoStatus**

6.2.23 Input

enum

MAIN
BACKUP

6.2.24 LongTermLoudness

struct

loudnessLevel	**optional float**
maxTruePeak	**optional float**
gain	**float**

6.2.25 Loudness

struct

ltll	LongTermLoudness
stll	optional YesNo
peakLimiter	optional YesNo

6.2.26 OnOff

enum

Undefined
On
Off

6.2.27 Output

struct

pid	optional int
basicConfig	BasicConfig
advancedConfig	optional DdppEnc
cannotEncode	bool
loudness	optional Loudness

6.2.28 Process

enum

ENCODE
TRANSCODE
DECODE
PASSTHROUGH

6.2.29 Source

struct

process	Process
pid	optional int
basicConfig	BasicConfig
advancedConfig	optional DdppDec
bitstreamMetadata	Source.bitstreamMetadata
cannotDecode	optional bool

6.2.30 Source.bitstreamMetadata

struct

dep	optional DepMetadata
ddpp	optional DdppMetadata
evo	optional EvolutionFrameworkMetadata

6.2.31 YesNo

enum

Undefined
Yes
No

7 cardAllocation (1.11)

7.1 Overview

Changelog:

1.11

- Changed
  - expectedInput from 1.7 to 1.8

1.10

- Changed
  - expectedInput from 1.6 to 1.7

1.9

- **Changed**
  - `expectedInput` from 1.5 to 1.6

1.8

- **Changed**
  - `expectedInput` from 1.4 to 1.5

1.7

- **Changed**
  - `expectedInput` from 1.3 to 1.4

1.6

- **Changed**
  - `expectedInput` from 1.2 to 1.3

1.5

- **Changed**
  - `expectedInput` from 1.1 to 1.2

7.2 Command Reference

7.2.1 GetCardAllocations

- **message GetCardAllocations.Request**
- **message GetCardAllocations.Response**

7.2.2 AssessAllocationPossibilities

- **message AssessAllocationPossibilities.Request**
- **message AssessAllocationPossibilities.Response**
- **message AssessAllocationPossibilities.Failure**

7.2.3 ReallocateAll

- message ReallocateAll.Request
- message ReallocateAll.Response
- message ReallocateAll.Failure

7.3 Type Reference

7.3.1 AssessAllocationPossibilities.Failure

empty struct

7.3.2 AssessAllocationPossibilities.Request

empty struct

7.3.3 AssessAllocationPossibilities.Response

struct

chipsRequired	int

7.3.4 CardAllocation

struct

slot	optional int
inputInstance	optional int
outputInstance	optional int
enableDeinterlacer	optional bool
useCase	optional list of ResourceType
expectedInput	expectedInput.ExpectedInput

7.3.5 CardAllocationObj

CardAllocation

7.3.6 GetCardAllocations.Request

empty struct

7.3.7 GetCardAllocations.Response

struct

data	map from UUID to CardAllocationObj
info	GetCardAllocations.Response.info

7.3.8 GetCardAllocations.Response.info

struct

totalEntries	**int**

7.3.9 ReallocateAll.Failure

empty **struct**

7.3.10 ReallocateAll.Request

empty **struct**

7.3.11 ReallocateAll.Response

struct

success	**bool**

7.3.12 ResourceType

enum

NOT_USE
RES_4K2K
FULL_HD
HD
SD

8 cardConfig (1.3)

8.1 Overview

Changelog

1.3

- **Changed**
  - `nmosConfig` from 1.0 to 1.1

1.2

- **Added**
  - `nmosConfig` 1.0
- **Removed**
  - `nmosRegistryConfig` 1.0

8.2 Command Reference

8.2.1 GetCardConfig

- **message GetCardConfig.Request**
- **message GetCardConfig.Response**
- **message GetCardConfig.Failure**

8.2.2 SetCardConfig

- **message SetCardConfig.Request**
- **message SetCardConfig.Response**
- **message SetCardConfig.Failure**

8.3 Type Reference

8.3.1 CardConfig

CardConfigStruct

8.3.2 CardConfigStruct

Card configuration parameters

struct

`timeReference`	**timeMode.TimeMode** Time reference (PTP/NTP)
`nmos`	**optional nmosConfig.NmosConfig** Parameters for accessing NMOS registry and Label Templates

8.3.3 GetCardConfig.Failure

empty **struct**

8.3.4 GetCardConfig.Request

empty **struct**

8.3.5 GetCardConfig.Response

struct

`config`	**CardConfig**

8.3.6 SetCardConfig.Failure

empty **struct**

8.3.7 SetCardConfig.Request

struct

`config`	**CardConfig**

8.3.8 SetCardConfig.Response

empty **struct**

9 cardStatus (1.10)

9.1 Overview

Changelog:

1.10

- **Changed**
  - Renamed `qsfpStatus` to `sfpStatus`

1.9

- **Changed**
  - `nmosStatusTypes` from 1.1 to 1.2

1.8

- **Changed**
  - Moved `NmosStatus` to `nmosStatusTypes` version 1.1

1.7

- **Added**
  - `nodeApi` map of interfaces to node API URIs
  - `NmosStatus structure defined, and added to cardStatus structure.`
  - `NmosRegistryStatus` put into NmosStatus.
- **Changed**
  - `LinkModes` from 1.2 to 1.3

1.6

- **Changed**
  - `linkModes` from 1.1 to 1.2

1.5

- **Changed**
  - `sfpStatus` from 1.4 to 1.5

9.2 Command Reference

9.2.1 GetCardStatus

- **message GetCardStatus.Request**
- **message GetCardStatus.Response**
- **message GetCardStatus.Failure**

9.3 Type Reference

9.3.1 CardStatus

struct

physicalPortStatus	optional PhysicalPortStatus
ptpLock	optional bool
nmosStatus	optional nmosStatusTypes.NmosStatus
sfpStatus	optional map from physicalIpPort.PortName to sfpStatus.SfpStatus

9.3.2 DataRate

bigint

9.3.3 GetCardStatus.Failure

empty struct

9.3.4 GetCardStatus.Request

GetCardStatusRequest

9.3.5 GetCardStatus.Response

CardStatus

9.3.6 GetCardStatusRequest

struct

slot	int

9.3.7 PhysicalPortStatus

struct

linkSpeed	map from physicalIpPort.PortName to linkModes.IpLinkSpeed
dataRate	map from physicalIpPort.PortName to DataRate
lldpNeighbor	map from physicalIpPort.PortName to lldp.LldpNeighborStruct

10 codedAudio (1.1)

10.1 Overview

```
# Changelog
## 1.1
- **Added**
  - `AC4` to `AudioCodec`
```

10.2 Type Reference

10.2.1 AudioCodec

enum

PCM	
MPEG1L2	
MPEG1L3	
AAC_LC	
AAC_HEv1	
AAC_HEv2	
AC3	
EAC3	
DOLBY_E	
AC4	

10.2.2 Hack

struct

profile	**int**

11 codedVideo (1.0)

11.1 Type Reference

11.1.1 OptimalizationTarget

Optimalization Target

Target of encoding optimalization

enum

VISUAL	Optimize for optimal visuals
PSNR	Optimize for maximized Peak Signal To Noise Ratio

12 coderConfig (2.49)

12.1 Overview

Changelog:

2.49

- **Changed**
  - `coderService` from 2.46 to 2.47

2.48

- **Changed**
  - `coderService` from 2.45 to 2.46

2.47

- **Changed**
  - `coderService` from 2.44 to 2.45
  - `cardAllocation` from 1.9 to 1.10

2.46

- **Changed**
  - `coderService` from 2.43 to 2.44
  - `cardAllocation` from 1.8 to 1.9

2.45

- **Changed**
  - `coderService` from 2.42 to 2.43

2.44

- **Changed**
  - `coderService` from 2.41 to 2.42

2.43

- **Changed**
  - `rawVideo` from 1.4 to 1.5
  - `coderService` from 2.40 to 2.41
  - `cardAllocation` from 1.7 to 1.8

2.42

- **Changed**
  - `coderService` from 2.39 to 2.40
  - `cardAllocation` from 1.6 to 1.7

2.41

- **Changed**
  - `coderService` from 2.38 to 2.39

2.40

- **Changed**
  - `coderService` from 2.37 to 2.38

2.39

- **Changed**
  - `coderService` from 2.36 to 2.37
  - `cardAllocation` from 1.5 to 1.6

2.38

- **Changed**
  - `coderService` from 2.35 to 2.36

2.37

- **Changed**
  - `coderService` from 2.34 to 2.35

2.36

- **Changed**
  - `coderService` from 2.33 to 2.34

2.35

- **Changed**
  - `coderService` from 2.32 to 2.33

2.34

- **Changed**
  - `coderService` from 2.31 to 2.32

2.33

- **Changed**
  - `coderService` from 2.30 to 2.31

2.32

- **Changed**
  - imported `esamConfig` version 1.0
  - added `esamConfig` to `MultiServiceInfo`
  - `coderService` from 2.29 to 2.30

2.31

- **Changed**
  - `coderService` from 2.28 to 2.29

2.30

- **Changed**
  - `coderService` from 2.27 to 2.28

2.29

- **Changed**
  - `coderService` from 2.26 to 2.27

2.28

- **Changed**
  - added `ready` bool to StartupResponse
  - `coderService` from 2.25 to 2.26

12.2 Type Reference

12.2.1 Foo

struct

`id`	**UUID**
`service`	**coderService.CoderService**
`multiServiceInfo`	**optional MultiServiceInfo**
`cardAllocation`	**cardAllocation.CardAllocation**

12.2.2 MessageType

enum

`NEW`
`UPDATE`
`DELETE`

12.2.3 MultiServiceInfo

struct

`id`	**UUID**
`serviceIndex`	**int**
`profileMode`	**multiServiceProfileMode.ProfileMode**
`lowestFrameRateInMultiService`	**optional rawVideo.FrameRate**
`esamConfig`	**optional esamConfig.EsamConfig**

13 coderService (2.47)

13.1 Overview

Changelog

2.47

- **Changed**
  - `videoProfile` from 2.13 to 2.14

2.46

- **Added**
  - `DepDecoder` struct for Dolby E decoding configuration with addition of stereoDownmixMode
- **Changed**
  - `DolbyAudio` to use optional `DepDecoder` (`depdec`) instead of `dolbyEProgNum`. `dolbyEProgNum` moved into depdec.

2.45

- **Added**
  - `CommonAudioOptions` struct containing autoSelectUniqueAudioSourcePids and s302mBitDepth
- **Changed**
  - `audioProfile` from 2.6 to 2.7
  - `colorProfile` from 2.2 to 2.3
  - `colorComponet` from 1.1 to 1.2
  - 'autoSelectUniqueAudioSourcePids' moved from `CoderService` top level into 'commonAudioOptions'
  - `videoProfile` from 2.12 to 2.13

2.44

- **Changed**
  - `videoProfile` from 2.11 to 2.12
  - `timecodeConfig` from 1.0 to 1.1
  - `scte35Config` from 1.0 to 1.1

2.43

- **Added**
  - `Channel` struct containing the channel id and corresponding audio essence id
  - `LR_CLfe_LsRs_TflTfr_TblTbr` to `ChannelMapping` to match 5.1.4 channel mode use-case
  - `LR_CLfe_LsRs_LbRb_TflTfr_TblTbr` to `ChannelMapping` to match 7.1.4 channel mode use-case
  - `LR_CLfe_LsRs_LbRb_TflTfr_TslTsr_TblTbr_LwRw` to `ChannelMapping` to match 9.1.6 channel mode use-case
- **Changed**
  - `EmbeddedAudioChannels` to use list(Channel) instead of list(int) to allow custom essence

allocation

  - `poolCtrlAllocs` from 1.1 to 1.2
  - Rename all instances of `master`/`slave` to `leader`/`follower`

- **Removed**
  - `audioEssenceId` from `EmbeddedAudioChannels`. Moved to `Channel` struct

2.42

- **Added**
  - new optional `latency0verride` member to the `CoderService` type.
- **Changed**
  - `audioProfile` from 2.5 to 2.6

2.41

- **Changed**
  - `videoProfile` from 2.10 to 2.11
  - `vancProfile` from 1.6 to 1.7

2.40

- **Changed**
  - `videoProfile` from 2.9 to 2.10

2.39

- **Changed**
  - Decoder IDCL migrations, in order to support downgrading to older versions
  - `colorComponent` from 1.0 to 1.1
  - `colorProfile` from 2.1 to 2.2
  - `vancProfile` from 1.5 to 1.6
  - `tsSource` from 1.4 to 1.5

2.38

- **Changed**
  - `audioProfile` from 2.4 to 2.5

2.37

- **Added**
  - New optional `pidSelection` member to `VancComponent`.
  - New list `pidList` member to `VancComponent`.
- **Changed**
  - `videoProfile` from 1.8 to 1.9

2.36

- **Changed**
  - `tsSource` from 1.3 to 1.4

2.35

- **Added**
  - `LR_CLfe` to `ChannelMapping` to match 3.1 channel mode use-case.
  - `identifier` optional field added to VancComponent struct, to be able to reference individual vanc components.
- **Changed**
  - `tsSource` from 1.2 to 1.3
  - `ipStream` from 1.2 to 1.3
  - `ipService` from 1.1 to 1.2
  - `audioProfile` from 2.3 to 2.4
  - `vancProfile` from 1.4 to 1.5

2.34

- **Added**
  - `logoInsertion` 1.0
- **Changed**
  - EmbeddedAudio to use EmbeddedAudioChannels to enable custom source channels on encoders.

2.33

- **Changed**
  - `vancProfile` from 1.3 to 1.4
  - `EmbeddedAudio` `codec` made optional. Combine `codec` as empty optional with `passthrough` as true as the new way to configure `ULL - S302M Transparent Passthrough`

2.32

- **Added**
  - `s302mPayload` to `TsAudioSource` in order to specify the content of a s302m PID

2.31

- **Changed**
  - `poolCtrlAllocs` from 1.0 to 1.1

2.30

- **Added**
  - `scte35Config` 1.0
- **Changed**
  - `idrOnScte35` optional bool changed to optional `Scte35Config`

2.29

- **Added**
  - Optional list of `AudioComponentPattern` added to `EmbeddedAudio` to be used by ABR JPEG XS TS encoder only.
- **Changed**

  - `ipService` from 1.0 to 1.1
  - Moved the `ColorComponent` type to its own file.
  - `ipConnection` removed as it is no longer in use here.

2.28

- **Added**
  - Added ipService import which has the IpService type which supports input profile redundancy.
- **Changed**
  - `VideoOut` edited to fit the new input redundancy
  - `ipConnection` from 1.2 to 1.3

2.27

- **Added**
  - `latencyAdjustment` field added to AudioComponent for forwarding audio earlier than video.

2.26

- **Added**
  - `slateInsertion` 1.0
  - slateInsertion field added to CoderService for configuration of the Slate Insertion feature.

2.25

- **Changed**
  - `component` from 1.3 to 1.5
  - `tsSource` from 1.1 to 1.2

2.24

- **Changed**
  - `testGenComponent` from 1.0 to 1.1
  - `testGeneratorProfile` from 1.0 to 1.1

2.23

- **Added**
  - `sdiConnection` 1.0
  - `tsDestination` 1.0
  - `testGenComponent` 1.0
- **Removed**
  - `SdiConnection` struct, moved to a separate file
  - `SdiConnectors` enum, moved to a separate file
  - `TestGenComponent` struct, moved to a separate file
  - `TsDestination` struct, moved to a separate file

13.2 Command Reference

13.2.1 GetCoderServices

- message GetCoderServices.Request
- message GetCoderServices.Response
- message GetCoderServices.Failure

13.2.2 SetCoderServices

- message SetCoderServices.Request
- message SetCoderServices.Response
- message SetCoderServices.Failure

13.2.3 DeleteCoderServices

- message DeleteCoderServices.Request
- message DeleteCoderServices.Response
- message DeleteCoderServices.Failure

13.3 Type Reference

13.3.1 AudioComponent

struct

uid	int
source	AudioSource
numAuPerPes	optional int
lipSyncAdjustment	optional int
latencyAdjustment	optional int
levelAdjustment	optional int
loudness	optional Loudness
passthrough	bool
profile	AudioComponent.profile
destination	AudioOut

13.3.2 AudioComponent.profile

struct

id	optional UUID
cpy	optional audioProfile.AudioProfileStruct

13.3.3 AudioComponentPattern

struct

language	string
component	optional component.ComponentType
pid	optional int

13.3.4 AudioComponentSet

struct

main	**AudioComponent**
backup	**optional AudioComponent**

13.3.5 AudioOut

variant

embedded	**EmbeddedAudio**
ts	**TsAudio**

13.3.6 AudioSource

variant

embedded	**EmbeddedAudio**
ts	**TsAudioSource**

13.3.7 AudioType

enum

UNDEFINED
CLEAN_EFFECTS
HEARING_IMPAIRED
VISUALLY_IMPAIRED

13.3.8 Channel

Contains information about where a channel should be placed within an essence.

struct

channel	**int** The source or desitation channel .
audioEssenceId	**optional UUID** In which essence is this channel placed, if an 2110 product is used.

13.3.9 ChannelMapping

enum

C
LR
LR_CLfe
LR_CLfe_LsRs
LR_LsRs_CLfe

LC_RLs_RsLfe	
LLs_CRs_RLfe	
LC_RsR_LsLfe	
CL_RLs_RsLfe	
LR_CLfe_LsRs_LbRb	
LR_LsRs_CLfe_LbRb	
LC_RLs_RsLfe_LbRb	
LLs_CRs_RLfe_LbRb	
LC_RsR_LsLfe_LbRb	
CL_RLs_RsLfe_LbRb	
LR_CLfe_LsRs_TflTfr_TblTbr	
LR_CLfe_LsRs_LbRb_TflTfr_TblTb	
LR_CLfe_LsRs_LbRb_TflTfr_TslTs	

13.3.10 CoderService

Configuration for an encoder or decoder service.

struct

label	**string** A label for the service.
enabled	**bool** True if the service should be enabled.
slot	**int** Which slot the service should run on (not applicable for Transcoders).
coderType	**CoderType** The type of coder the service is for.
video	**VideoComponent** Configuration of the input and output video for the coder chip.
audios	**list** of **AudioComponentSet** Configuration for the audio components.
output	**OutputType** The type of output that the service will produce.
color	**optional colorComponent.ColorComponent** Configuration for the color of the video.
vancs	**list** of **VancComponent** Configuration for the vanc components.
signalLoss	**testGenerator.SignalLoss** Configuration for the output when input signal is lost.
lock	**LockComponent** Configuration for how the service should be locked.
passthroughComponents	**optional bool** Set to true if all components should be passthrough. Only valid for transcoder.
	optional int

additionalLatency	Additional latency to be added to the entire service. Unit is 100ms. The optional value should be unset for products which do not support this feature.
latencyOverride	**LatencyOverride** Latency override for encoders. Use to override the video latency through the encoder. Auto finds a suitable value automatically, manual can be used to set a custom latency.
testGenerator	**optional testGenComponent.TestGenComponent** Contains optional structure with link to a test generator profile.
scte35	**optional scte35Config.Scte35Config**
slateInsertion	**optional slateInsertion.SlateInsertion** Enable the slate insertion feature. Only supported on encoder.
logoInsertion	**optional logoInsertion.LogoInsertion** Contains optional structure with configuration for logo insertion.
commonAudioOptions	**CommonAudioOptions**

13.3.11 CoderServiceObj

CoderService

13.3.12 CoderType

The types of coders that a config can be made for.

enum

ENCODER
TRANSCODER
DECODER

13.3.13 CommonAudioOptions

Audio parameters that will apply to all audios in the CoderService

struct

autoSelectUniqueAudioSourcePid<	**optional bool** Automatically select unique PIDs for audio components. PIDs are selected in order of increasing numerical PID value from the PMT such that the first audio component uses the lowest PID, second audio component the next to lowest PID and so forth. PMT updates with changes to PID lineup will trigger a reconfiguration of all audio components. If the number of audio components exceed the number of audio PIDs in the PMT the extra audio components will not run. Only valid for decoders.
s302mBitDepth	**optional audioProfile.SampleBitDepth** Contains optional structure with configuration for S302M passthrough configuration

13.3.14 DdppDecDrcMode

enum

NONE	
LINE	
RF	

13.3.15 DdppDecoder

DdppDecoder.

struct

ddppDecDrcMode	**DdppDecDrcMode** Selects the mode for Dynamic Range Compression.
stereoDownmixMode	**StereoDownmixMode** Selects the mode for downmixing.

13.3.16 DeleteCoderServices.Failure

empty **struct**

13.3.17 DeleteCoderServices.Request

struct

ids	**list** of **UUID**

13.3.18 DeleteCoderServices.Response

empty **struct**

13.3.19 DepDecoder

DepDecoder.

struct

dolbyEProgNum	**int** Selects a Dolby E program as source for the decoder and transcoder.
stereoDownmixMode	**StereoDownmixMode** Selects the mode for downmixing.

13.3.20 DolbyAudio

Dolby Audio.

struct

	optional DepDecoder

depdec	Specifies configuration for Dolby E decoding.
ddppdec	**optional** DdppDecoder Specifies configuration for Dolby Digital and Dolby Digital Plus decoding.

13.3.21 EmbeddedAudio

Configuration for embedded audio.

struct

codec	**optional** codedAudio.AudioCodec Audio codec
channels	EmbeddedAudioChannels Embedded audio content
dolbyEProgNum	**optional int** Dolby E program number
channelMode	**optional** rawAudio.ChannelMode Channel mode - mono, stereo, 5.1, etc.
channelMapping	**optional** ChannelMapping Dictates how embedded audio channels are interpreted with respect to order.
componentPriorityList	**optional list** of AudioComponentPattern Optional prioritized list of input components from a transport stream.

13.3.22 EmbeddedAudioChannels

States where embedded audio content should be read from or written to in the SDI signal.

struct

channels	**list** of Channel The source or destination channel(s) for the audio content
isCustom	**bool** States whether a user specified list deviates from the standard set, i.e. non-adjacent channels.

13.3.23 Genlock

struct

offset	**int**

13.3.24 GetCoderServices.Failure

empty **struct**

13.3.25 GetCoderServices.Request

empty **struct**

13.3.26 GetCoderServices.Response

struct

data	**map** from **UUID** to **CoderServiceObj**
info	**GetCoderServices.Response.info**

13.3.27 GetCoderServices.Response.info

struct

totalEntries	**int**

13.3.28 IpOut

string

13.3.29 LatencyOverride

variant

auto	**LatencyOverride.auto**
manual	**int**

13.3.30 LatencyOverride.auto

empty **struct**

13.3.31 LockComponent

struct

source	**optional bool**
genlock	**optional Genlock**
videoAlignment	**optional VideoAlignment**

13.3.32 LockComponentEnum

enum

SOURCE
GENLOCK
VIDEOALIGNMENT
VIDEOALIGNMENT_AND_GENLOCK
NONE

13.3.33 Loudness

struct

longTermLoudnessTargetLevel	**int**
shortTermDeviation	**optional int**
peakLimit	**optional int**

13.3.34 OutputType

The types of output that a service can produce.

variant

ts	**tsDestination.TsDestination**
sdi	**SdiOut**
ip	**IpOut**

13.3.35 PidSelection

enum

ALL
ANY
LIST

13.3.36 SdiOut

empty **struct**

13.3.37 SetCoderServices.Failure

empty **struct**

13.3.38 SetCoderServices.Request

struct

data	**map** from **UUID** to **CoderServiceObj**

13.3.39 SetCoderServices.Response

empty **struct**

13.3.40 StereoDownmixMode

enum

AUTO
LTRT
LORO

13.3.41 TsAudio

struct

language	**string**
pid	**int**
audioType	**AudioType**

13.3.42 TsAudioSource

Transport Stream Audio Source.

struct

componentPriorityList	**list** of **AudioComponentPattern** A list of AudioComponentPattern objects describing audio components in a transport stream.
dolbyAudio	**DolbyAudio** Specifies decode configuration for Dolby Digital, Dolby Digital Plus, and Dolby E.
s302mPayload	**optional codedAudio.AudioCodec** Codec present in the s302m payload. Leave empty for transparent passthrough of all types of S302M payload.

13.3.43 TsService

struct

pid	**int**

13.3.44 VancComponent

Configuration of VANC component

struct

uid	**int** Unique identifier for a component.
profile	**VancComponent.profile** Reference and copy to a VANC profile.
pidSelection	**optional PidSelection** Optional options for which PIDs to be selected
pidList	**list** of **int** List of PIDs to accompany `PidSelection.LIST`.
pid	**int** PID.
vancParams	**vancProfile.VancParams** Configuration of the type of ancillary data.
identifier	**optional int** Optional int to reference individual components.

13.3.45 VancComponent.profile

struct

id	**UUID**
cpy	**optional vancProfile.VancProfileStruct**

13.3.46 VideoAlignment

struct

index	**int**
alignmentLeader	**optional poolCtrlAllocs.SlotFlow**

13.3.47 VideoComponent

Video component of the coder service.

struct

source	**VideoSource** The input video
profile	**VideoComponent.profile** Video profile used by the coder service
destination	**VideoOut** The output video
timecodeSource	**optional timecodeConfig.TimecodeSource** Source of the timecode to be used for SEI messages (only set for encoders, but not for JPEG 2000 encoder)

13.3.48 VideoComponent.profile

struct

id	**UUID**
cpy	**optional videoProfile.VideoProfileStruct**

13.3.49 VideoOut

variant

sdi	**sdiConnection.SdiConnection**
ts	**TsService**
ip	**ipService.IpService**

13.3.50 VideoSource

Configuration type for the video source.

variant

`sdi`	**sdiConnection.SdiConnection** configuration for SDI video source.
`ts`	**tsSource.TsSource** configuration for TS video source.
`ip`	**ipService.IpService** configuration for IP video source.

14 coderTypes (1.0)

14.1 Type Reference

14.1.1 Range

struct

min	**int**
max	**int**

15 colorComponent (1.2)

15.1 Overview

Changelog

1.2

- **Changed**
  - `colorProfile` from 2.2 to 2.3

1.1

- **Changed**
  - `colorProfile` from 2.1 to 2.2

1.0

- **Added**
  - Moved the ColorComponent type from coderService

15.2 Type Reference

15.2.1 ColorComponent

Configuration for a single color profile.

struct

`profile`	**ColorComponent.profile** The color profile with a UUID and a copy of the color profile struct

15.2.2 ColorComponent.profile

struct

`id`	**UUID**
`cpy`	**optional colorProfile.ColorProfileStruct**

16 colorProfile (2.3)

16.1 Command Reference

16.1.1 GetColorProfiles

- message GetColorProfiles.Request
- message GetColorProfiles.Response
- message GetColorProfiles.Failure

16.1.2 SetColorProfiles

- message SetColorProfiles.Request
- message SetColorProfiles.Response
- message SetColorProfiles.Failure

16.1.3 DeleteColorProfiles

- message DeleteColorProfiles.Request
- message DeleteColorProfiles.Response
- message DeleteColorProfiles.Failure

16.2 Type Reference

16.2.1 ColorProfile

ColorProfileStruct

16.2.2 ColorProfileStruct

struct

label	string
colorType	ColorType
vui	videoUsabilityInfo.VuiParameters
followInput	bool

16.2.3 ColorType

Changelog

2.3

- Changed
  - Use new videoUsability to make use of new common definition of Transfer, matrix and color space

enum

OFF
HLG_10
HDR_10

PQ_10	
SDR	
SDR_WCG	
HLG_10_ATF	
Manual	

16.2.4 DeleteColorProfiles.Failure

empty **struct**

16.2.5 DeleteColorProfiles.Request

struct

ids	**list** of **UUID**

16.2.6 DeleteColorProfiles.Response

empty **struct**

16.2.7 GetColorProfiles.Failure

empty **struct**

16.2.8 GetColorProfiles.Request

empty **struct**

16.2.9 GetColorProfiles.Response

struct

data	**map** from **UUID** to **ColorProfile**
info	**GetColorProfiles.Response.info**

16.2.10 GetColorProfiles.Response.info

struct

totalEntries	**int**

16.2.11 SetColorProfiles.Failure

empty **struct**

16.2.12 SetColorProfiles.Request

struct

`data`	**map** from **UUID** to **ColorProfile**

16.2.13 SetColorProfiles.Response

empty **struct**

17 commonTypes (1.0)

17.1 Type Reference

17.1.1 NmosVersionNumber

Generic version number type for NMOS APIs

enum

V1_2
V1_3

18 component (1.5)

18.1 Overview

Changelog:

1.5

- **Added**
  - SCTE27_SUB to ComponentType enum and respective mappings.

18.2 Type Reference

18.2.1 AnyComponent

empty **struct**

18.2.2 ComponentGroup

```
ComponentGroup: Grouping of ComponentType, used for filtering
```

enum

UNDEFINED	
AUDIO	
VIDEO	
SUBTITLE	
DATA	

18.2.3 ComponentLanguage

string

18.2.4 ComponentSelector

```
Component selector.

@param pid          Selects a component using a specific (source) PID.
@param language     Selects a component by a three character language string.
@param type         Selects a component by its type.
@param group        Selects a component by group membership.
@param any          Selects any component.
```

variant

pid	**Pid**
language	**ComponentLanguage**
componentType	**ComponentType**
componentGroup	**ComponentGroup**
any	**AnyComponent**

18.2.5 ComponentType

A ComponentType is typically identified by a stream type (see ISO 13818-1 Table 2-29), often distinguished with another header field such as a descriptor tag.

enum

MPEG_VIDEO
MPEG_AUDIO
TTX
DVBSUB
ECM
PCR
PRIVATE
AC_3
H264
VBI
AAC
VC_1
SCS_EMM
DPI_CUE
EAC3
AIT
HBBTV_CAROUSEL
DATA_CAROUSEL
H265
S302M
JPEG2000
S2038
DVBTTMLSUB
JPEG_XS_VIDEO
SCTE27_SUB

18.2.6 Pid

int

18.2.7 SubtitleFormat

enum

DVB
DVB_HARD_OF_HEARING
EBU
EBU_HARD_OF_HEARING

19 dejitterTypes (1.1)

19.1 Overview

Changelog:

1.1

- **Added**
  - PCR value added to DejitterType enum
  - PCR added to DejitterType2str

19.2 Type Reference

19.2.1 DejitterType

enum

```
OFF
RTP_2022_6
PCR
```

20 dpi (1.0)

20.1 Type Reference

20.1.1 Dpi

Configuration of the DPI data

struct

`source`	**DpiSource** Source of the DPI data
`heartbeat`	**bool** Whether heartbeat is on or off
`heartbeatInterval`	**int** Interval between each heartbeat
`ptsOffset`	**int** SCTE-35 PTS adjustment
`index`	**int** SCTE 104 DPI PID index

20.1.2 DpiSettings

Mapping an identifier to the DPI data

struct

`identifier`	**optional int** The identifier for mapping SCTE104 messages to the service.
`dpiData`	**Dpi** Configuration of the DPI data

20.1.3 DpiSource

Source of the DPI data

enum

`ANC`
`TCP`

21 dpiStatus (1.0)

21.1 Type Reference

21.1.1 DpiCommand

DPI Command info

struct

`commandType`	**int** Last SCTE command type.
`descriptorIds`	**list** of **int** List of segmentation descriptor IDs.

21.1.2 DpiInput

DPI input status.

struct

`sourceIdentifier`	**optional int** identifies the source of the input. Encoder: ANC is represented by -1, TCP input is represented by its identifier.
`received`	**bool** Received DPI input.
`invalidPackets`	**int** Input contained invalid packets.
`pidIndexes`	**list** of **int**
`timeSinceLastReceived`	**int**
`lastGeneratedCommand`	**optional DpiCommand**

21.1.3 DpiStatus

DPI status.

struct

`heartbeat`	**bool** Specifies whether heartbeat signal is on or off.
`enabled`	**bool** Specifies whether DPI is enabled.
`running`	**bool** Specifies whether DPI encoding is running without errors.
`pidIndex`	**int** SCTE PID index.
`timeSinceLastReceived`	**int** Seconds since last received DPI packet.
`timeSinceLastGenerated`	**int** Seconds since last generated DPI packet.
`lastCommandType`	**int** Type of last received SCTE command.

`discardedPackets`	**int** Number of discarded packets.

22 esamConfig (1.0)

22.1 Type Reference

22.1.1 EsamConfig

struct

`poisUri`	**string**

22.1.2 EsamServiceConfig

Multiservice ESAM configuration.

struct

`acquisitionPointId`	**string** The acquisition point ID that will be sent to the POIS server.

23 esamStatus (1.0)

23.1 Type Reference

23.1.1 EsamStatus

ESAM status.

struct

`poisConnectionStatus`	**PoisConnectionStatus** An enum to represent the connection status with the POIS server.

23.1.2 PoisConnectionStatus

UNKNOWN means no communication with the server yet

enum

`CONNECTED`
`NO_CONNECTION`
`UNKNOWN`

24 expectedInput (1.8)

24.1 Overview

Changelog

1.8

- **Changed**
  - `videoProfile` from 2.13 to 2.14

1.7

- **Changed**
  - `videoProfile` from 2.12 to 2.13

1.6

- **Changed**
  - `videoProfile` from 2.11 to 2.12

1.5

- **Changed**
  - `videoProfile` from 2.10 to 2.11
  - `rawVideo` from 1.4 to 1.5

1.4

- **Changed**
  - `videoProfile` from 2.9 to 2.10

1.3

- **Changed**
  - `videoProfile` from 2.8 to 2.9

1.2

- **Added**
  - Added ExpectedTsInput struct
  - Added ExpectedRawInput variant
  - New fields in ExpectedInput:
  - ts: ExpectedTsInput,
  - raw: ExpectedRawInput
- **Changed**
  - ExpectedInput changed from a struct to a variant of either ExpectedTsInput or ExpectedRawInput

24.2 Type Reference

24.2.1 ExpectedInput

Expected input format.

variant

`ts`	**ExpectedTsInput** the expected input format for a TS video source.
`raw`	**ExpectedRawInput** the expected input format for a RAW video source.

24.2.2 ExpectedRawInput

Expected raw input.

struct

`hResolution`	**rawVideo.HorizontalResolution** the expected horizontal resolution of the input.
`vResolution`	**rawVideo.VerticalResolution** the expected vertical resolution of the input.
`scan`	**videoProfile.ScanningMode** the expected scanning mode of the input.
`fps`	**rawVideo.FrameRate** the expected frame rate per second of the input.

24.2.3 ExpectedTsInput

Expected ts input.

struct

`interlaced`	**bool** Enables handling of interlaced inputs. If set it may consume more resources.
`codec`	**optional videoProfile.VideoCodec** Empty optional represents Any codec (in set MPEG2|AVC|HEVC)
`allow4k`	**bool** Enables handling of 4k inputs. If set it will consume more resources.
`allow10bit`	**bool** Enables handling of 10bit inputs. If set it will consume more resources.

25 firewallTypes (1.0)

25.1 Type Reference

25.1.1 IpAddress

string

25.1.2 PaginatedRequest

```
Instructs the server to return only a subset of the data found on the server

@param pageSize number of entries returned per request
@param pageNumber what page of the set to returned, determined by the slice [pageSize*pageNumber
```

struct

pageSize	**int**
pageNumber	**int**

25.1.3 PaginationInfo

struct

totalEntries	**int** the total number of entries on the server, before matching

25.1.4 SocketAddress

Tuple with IP address and port

struct

address	**IpAddress** IP address, either IPv4 or IPv6
port	**int**

25.1.5 UUID

string

26 imageUpload (1.0)

26.1 Overview

Changelog

1.0

- **Added**
  - First version of module

26.2 Command Reference

26.2.1 GetImages

- **message GetImages.Request**
- **message GetImages.Response**

26.2.2 SetImage

Must be sent after an image has been uploaded. It will make sure that image is stored permanently and is available for selection.

- **message SetImage.Request**
- **message SetImage.Response**

26.2.3 DeleteImage

- **message DeleteImage.Request**
- **message DeleteImage.Response**

26.3 Type Reference

26.3.1 DeleteImage.Request

struct

`id`	**string**

26.3.2 DeleteImage.Response

empty **struct**

26.3.3 GetImages.Request

empty **struct**

26.3.4 GetImages.Response

struct

`images`	**list** of **ImageInfo**

26.3.5 ImageInfo

Information about an image.

struct

`id`	**string** a unique identifier for the image.
`fileName`	**string** the image file name.

26.3.6 SetImage.Request

struct

`fileName`	**string**

26.3.7 SetImage.Response

struct

`fileId`	**string** A unique ID for the uploaded image. It is important to be aware that the ID is based on the content of the file. So if identical files (with different file names) are uploaded they will overwrite eachother.

27 inputRedundancyStatus (1.3)

27.1 Overview

Changelog

1.3

- **Changed**
  - `psiStatus` from 1.5 to 1.6

1.2

- **Changed**
  - `psiStatus` from 1.4 to 1.5

27.2 Type Reference

27.2.1 HotStandbyStatus

Hot Standby status for input redundancy.

struct

`identicalSourcesStatus`	**optional** **IdenticalSourcesStatus**

27.2.2 IdenticalSourcesStatus

Identical sources status for input redundancy.

struct

`measuredOffset`	**int** Offset between two identical sources in ms.

27.2.3 SourceStatus

Source status for input redundancy.

struct

`components`	**psiStatus.PsiStatus** Components present in the given source.
`present`	**bool** True if the source is present, false otherwise.
`active`	**bool** True if the source is the active one in a input redundancy set-up.

27.2.4 TsInputRedundancyStatus

TS input redundancy status.

struct

`sourceStatus`	**map** from **streamSource.StreamSource** to **SourceStatus** Status for the given stream source.
`hotStandbyStatus`	**HotStandbyStatus**

28 ipConnection (1.5)

28.1 Overview

Changelog:

1.5

- **Changed**
  - `ipStream` from 1.2 to 1.3

1.4

- **Changed**
  - `dejitterTypes` from 1.0 to 1.1

1.3

- **Added**
  - `MPEG_TS` value added to IpStandard enum

1.2

- **Added**
  - `nmosEnable` field added to IpConnection, with a default value of false

28.2 Command Reference

28.2.1 GetIpConnections

- **message GetIpConnections.Request**
- **message GetIpConnections.Response**
- **message GetIpConnections.Failure**

28.2.2 SetIpConnections

- **message SetIpConnections.Request**
- **message SetIpConnections.Response**
- **message SetIpConnections.Failure**

28.2.3 DeleteIpConnections

- **message DeleteIpConnections.Request**
- **message DeleteIpConnections.Response**
- **message DeleteIpConnections.Failure**

28.3 Type Reference

28.3.1 Dejitter

struct

`mode`	**dejitterTypes.DejitterType**
`bufferSize`	**int**

28.3.2 DeleteIpConnections.Failure

empty **struct**

28.3.3 DeleteIpConnections.Request

struct

ids	**list** of **UUID**

28.3.4 DeleteIpConnections.Response

empty **struct**

28.3.5 GetIpConnections.Failure

empty **struct**

28.3.6 GetIpConnections.Request

empty **struct**

28.3.7 GetIpConnections.Response

struct

data	**map** from **UUID** to **IpConnectionObj**
info	**GetIpConnections.Response.info**

28.3.8 GetIpConnections.Response.info

struct

totalEntries	**int**

28.3.9 IpConnection

Group of settings for the IP inputs/outputs of a single service.

struct

label	**string** Label for the entire Group.
connection	**ipStream.Connection** Connections settings like addresses, ports, etc.
dejitter	**optional Dejitter** Dejitter for the input of IP 2110 and IP 2022 encoders.
standard	**IpStandard** SMPTE standard that is used for this group.
	bool

nmosEnable	for allowing NMOS system to update the input/output profiles

28.3.10 IpConnectionObj

IpConnection

28.3.11 IpStandard

enum

SMPTE_2022
SMPTE_2110
MPEG_TS

28.3.12 SetIpConnections.Failure

empty **struct**

28.3.13 SetIpConnections.Request

struct

data	**map** from **UUID** to **IpConnectionObj**

28.3.14 SetIpConnections.Response

empty **struct**

29 ipInterface (1.6)

29.1 Overview

Changelog

1.6

- **Changed**
  - `linkModes` import from 1.2 to 1.3

1.5

- **Added**
  - `ipFecMode` added to `IpInterfaceStruct`

1.4

- **Changed**
  - `lldp` import from 1.0 to 1.1

29.2 Command Reference

29.2.1 GetIpInterfaces

- **message GetIpInterfaces.Request**
- **message GetIpInterfaces.Response**
- **message GetIpInterfaces.Failure**

29.2.2 SetIpInterfaces

Interface uuids are deterministically calculated The uuids used in the RPC Request are ignored

- **message SetIpInterfaces.Request**
- **message SetIpInterfaces.Response**
- **message SetIpInterfaces.Failure**

29.2.3 DeleteIpInterfaces

- **message DeleteIpInterfaces.Request**
- **message DeleteIpInterfaces.Response**
- **message DeleteIpInterfaces.Failure**

29.3 Type Reference

29.3.1 DeleteIpInterfaces.Failure

empty **struct**

29.3.2 DeleteIpInterfaces.Request

struct

`ids`	**list** of **UUID**

29.3.3 DeleteIpInterfaces.Response

empty **struct**

29.3.4 GetIpInterfaces.Failure

empty **struct**

29.3.5 GetIpInterfaces.Request

empty **struct**

29.3.6 GetIpInterfaces.Response

struct

data	**map** from **UUID** to **IpInterface**

29.3.7 IpInterface

IpInterfaceStruct

29.3.8 IpInterfaceConfig

struct

ipAddress	**firewallTypes.IpAddress**
gateway	**firewallTypes.IpAddress**
netmask	**int**

29.3.9 IpInterfaceStruct

struct

ipLink	**linkModes.IpLinkMode**
ipFecMode	**optional linkModes.IpFecMode**
interfaces	**IpInterfaceVariant**

29.3.10 IpInterfaceVariant

variant

single	**LogicalInterface**
quad	**IpInterfaceVariant.quad**

29.3.11 IpInterfaceVariant.quad

struct

first	**LogicalInterface**

second	LogicalInterface
third	LogicalInterface
fourth	LogicalInterface

29.3.12 LogicalInterface

struct

label	string
enabled	bool
physicalPort	physicalIpPort.PortName
ipv4	optional IpInterfaceConfig
ipv6	optional IpInterfaceConfig
lldpMode	optional lldp.LldpMode

29.3.13 SetIpInterfaces.Failure

empty struct

29.3.14 SetIpInterfaces.Request

struct

data	map from UUID to IpInterface

29.3.15 SetIpInterfaces.Response

empty struct

30 ipService (1.2)

30.1 Overview

Changelog

1.2

- **Changed**
  - `ipConnection` from 1.4 to 1.5
  - `ipStream` from 1.2 to 1.3

1.1

- **Changed**
  - `ipConnection` from 1.3 to 1.4

1.0

- **Added**
  - New type `IpServices` added, containing multiple IP profiles and switching mode in order to allow for 2022-7 Input redundancy
  - Moved the IpProfile type from coderService

30.2 Type Reference

30.2.1 IpProfile

Configuration for a single IP video source.

struct

`id`	**UUID** UUID of the IpConnection profile containing the configuration.
`cpy`	**optional ipConnection.IpConnection** A copy of the content of the IpConnection profile.

30.2.2 IpService

Configuration for one or more IP video sources.

struct

`profiles`	**list** of **IpProfile**
`profileRedundancy`	**optional ipStream.NearSeamless**
`sdiType`	**sdiType.SdiType**

31 ipStatus (1.7)

31.1 Overview

Changelog:

1.7

- **Changed**
  - `sfpStatus` from 1.4 to 1.5

31.2 Type Reference

31.2.1 InstanceId

int

31.2.2 IpStatus

struct

`id`	**InstanceId**
`tico`	**optional bool**
`sfp`	**optional sfpStatus.SfpStatus**
`ipStreamStatuses`	**optional list** of **ipStreamStatus.IpStreamStatus**
`alternateInputStatus`	**optional alternateInputStatus.AlternateInputStatus**

32 ipStream (1.3)

32.1 Overview

```
# Changelog
## 1.3
- **Changed**
 - `st2110Types` from 1.0 to 1.2.
```

32.2 Type Reference

32.2.1 AlternateInput

IP config for a service using alternative input.

struct

path1	**IpStreamGroup** Config for path one.
path2	**IpStreamGroup** Config for path two.
bufferSize	**int** Input buffer size in milliseconds.
seamlessType	**SeamlessType** Config for the different types of alternative input.

32.2.2 Connection

A redundant Ip Connection.

variant

direct	**IpStreamGroup** No redundancy.
alternateInput	**AlternateInput** Seamless or alternative input.

32.2.3 EssenceProperties

Essence-specific properties of a 2110 essence.

variant

audio	**St2110AudioProperties** Properties of an audio essence.

32.2.4 IpStream

This struct stores the address and other configurable options of an input or output stream on an IP network. On 2110 products it represents an essence.

struct

`label`	**string** A descriptive label or name for this stream.
`address`	**firewallTypes.SocketAddress** The IP address of this stream.
`igmpv3Source`	**optional firewallTypes.IpAddress** IP address of the IGMP source used for source filtering.
`payloadId`	**int** Id of the essence.
`st2110Info`	**optional St2110Info** Information which is only available for 2110 applications.
`id`	**UUID** Unique id within an IpStreamGroup. Redundant streams (in different groups) have the same id.

32.2.5 IpStreamGroup

A group of IpStreams that share the same physical port.

struct

`bufferSize`	**optional int** Size of buffer used when synchronizing essences.
`source`	**list** of **IpStream** The list of IpStreams in the group. Shall contain a single entry for 2022, but may contain multiple for 2110.
`physicalPortId`	**physicalIpPort.PortName** The id of the physical port on which the IpStreams are sent/received.

32.2.6 NearSeamless

Config for near seamless mode.

struct

`switchingMode`	**SwitchingMode** The mode of switching.
`timeoutMainToSpare`	**int** Timeout before switching to spare in milliseconds.
`timeoutSpareToMain`	**int** Timeout before switching to main in milliseconds.

32.2.7 SeamlessType

Config for seamless.

variant

`smpte2022_7`	**SeamlessType.smpte2022_7** Config for SMPTE 2022-7 seamless.
`nearSeamless`	**NearSeamless** Config for near seamless mode.

32.2.8 SeamlessType.smpte2022_7

empty **struct**

32.2.9 St2110AudioProperties

Properties specific to a 2110 audio essence. Either -30 or -31.

struct

`totalChannels`	**int** The number of audio channels in the essence.
`packetTime`	**st2110Types.St2110AudioPacketTime** Audio packet time.

32.2.10 St2110Info

Information regarding any kind of 2110 essence.

struct

`standard`	**st2110Types.St2110Standard** The standard which the essence content follows.
`properties`	**EssenceProperties** Properties specific to a type of essence.

32.2.11 SwitchingMode

The switching modes that can be used with alternative input.

enum

`PREFER_PATH1`
`PREFER_PATH2`
`FLOATING`
`FORCE_PATH1`
`FORCE_PATH2`

33 ipStreamStatus (1.1)

33.1 Type Reference

33.1.1 IpStreamStatus

Represents the status of an IpStream.

struct

id	**UUID** The id of the IpStream this status represents.
bitrate	**float** The bitrate of the IpStream.
networkProtocol	**NetworkProtocol** Protocol used for seamless input.
sequenceErrors	**int** Number of gaps in RTP sequence numbers.
rtpSsrc	**int** Synchronization Source that uniquely identifies the source of a stream. Ssrc's within an RTP session will be unique.
minDelay	**int** Minimum network latency. [ms]
maxDelay	**int** Maximum network latency. [ms]

33.1.2 NetworkProtocol

Possible protocols of a seamless input.

enum

RTP
SequencedUDP
UDP
Unknown

34 ipStreamUtils (0.0)

35 linkModes (1.3)

35.1 Type Reference

35.1.1 IpFecMode

enum

NO_FEC
RS_FEC
FC_FEC

35.1.2 IpLinkMode

enum

IP_10G
IP_25G_COPPER
IP_25G_OPTICAL
IP_40G
IP_4x10G

35.1.3 IpLinkSpeed

enum

IP_10G
IP_25G
IP_40G
IP_4x10G
NO_LINK

36 lldp (1.1)

36.1 Overview

```
# Changelog

## 1.1
- **Changed**
    - Extracted the structure inside of the `LldpNeighbor` object definition into a new type cal
```

36.2 Type Reference

36.2.1 LldpMedInventory

LLDP-MED Inventory. Appear specific implementation in parentheses.

struct

hwRev	**string** Hardware Revision
swRev	**string** Software Revision (Software Version)
fwRev	**string** Firmware Revision (FPGA Version)
serial	**string** Serial Number
manufacturer	**string** Manufacturer (“Appear”)
model	**string** Model (“X10” or “X20”)
assetId	**string** Asset ID (Hardware ID)

36.2.2 LldpMode

Choose either private or public configuration of LLDP frames.

enum

PUBLIC	Only mandatory TLVs and port detail TLVs.
PRIVATE	LLDP-MED inventory in addition to the PUBLIC TLVs.

36.2.3 LldpNeighbor

LldpNeighborStruct

36.2.4 LldpNeighborStruct

The optional variables are only present if GetLldpNeighborRequest.details is set to true in the rpc request. The following descriptions are the Appear specific implementations of each LLDP TLV. If neighbors from other manufacturers are detected, TLV values might differ. The LldpMode is specified in

parentheses for variables that have different implementations based on the LldpMode set in the neighbor.

struct

`chassisId`	**string** MAC address of CTRL on MMI 1, fallback to CTRL on MMI 2 (PRIVATE) or "N/A" (PUBLIC).
`mgmtIp`	**list** of **string** IPv4 address of CTRL on MMI 1 and/or CTRL on MMI 2. Also IPv6 if enabled on the neighbor.
`sysDescr`	**optional string** Desktop Heading (PRIVATE) or "Unknown" (PUBLIC)
`sysName`	**optional string** Hostname (PRIVATE) or Interface.Port (PUBLIC)
`portId`	**string** MAC address of local port
`portDescr`	**optional string** Slot, Interface: Port-label
`ttl`	**optional string** Time-To-Live, how long an LLDP packet received from that neighbor should be considered valid.
`lldpMedInventory`	**optional LldpMedInventory** LLDP-MED Inventory

37 lockStatus (1.5)

37.1 Overview

Changelog:

1.5

- **Changed**
  - `poolCtrlAllocs` from 1.1 to 1.2
  - Rename all instances of `master/slave` to `leader/follower`

1.4

- **Changed**
  - `LockStatus` is now a struct instead of a variant
  - `LockStatus` struct takes both `genLockStatus` and `videoAlignmentStatus` as optional

1.3

- **Changed**
  - `poolCtrlAllocs` from 1.0 to 1.1
- **Added**
  - `flow` from 1.0 to 1.1

37.2 Type Reference

37.2.1 GenlockStatus

struct

`locked`	**bool**
`offset`	**int**
`validMode`	**bool**
`localReference`	**ReferenceLockStatus**
`remoteReference`	**ReferenceLockStatus**

37.2.2 LockStatus

struct

`genlockStatus`	**optional GenlockStatus**
`videoAlignmentStatus`	**optional VideoAlignmentStatus**

37.2.3 ReferenceLockStatus

enum

`UNLOCKED`
`LOCKED`
`HOLDOVER`

37.2.4 VideoAlignmentStatus

struct

locked	**bool**
offset	**int**
group	**int**
alignmentLeader	**poolCtrlAllocs.SlotFlow**
outOfAlignment	**bool**
flow	**int**

38 logoInsertion (1.0)

38.1 Overview

Changelog

1.0

- **Added**
  - First version of module

38.2 Type Reference

38.2.1 LogoInsertion

Configure logo insertion.

struct

`id`	**string** A unique identifier for the logo. This is the value returned by the imageUpload.SetImage RPC.
`xOffset`	**int** X offset from the top left corner in pixels.
`yOffset`	**int** Y offset from the top left corner in pixels.

39 movedService (1.0)

39.1 Overview

Changelog

1.0

- **Added**
  - added MovedServiceStruct and Obj
  - added slot type

39.2 Type Reference

39.2.1 MovedService

MovedServiceStruct

39.2.2 MovedServiceStruct

Struct that stores where services are being moved during service redundancy switch. For example in N+M redundancy mode

struct

`serviceId`	**UUID** Id of the service which will be moved
`fromSlot`	**Slot** From which slot the service will be moved
`toSlot`	**optional Slot** To which slot the service will be moved

39.2.3 Slot

Slot used by the service in the MovedServiceStruct

int

40 multiService (2.19)

40.1 Overview

Changelog

2.19

- **Changed**
  - `coderService` from 2.46 to 2.47

2.18

- **Changed**
  - `coderService` from 2.45 to 2.46

2.17

- **Changed**
  - `coderService` from 2.44 to 2.45
  - `colorComponent` from 1.1 to 1.2

2.16

- **Changed**
  - `scte35Config` from 1.0 to 1.1

2.15

- **Changed**
  - `coderService` from 2.42 to 2.43

2.14

- **Changed**
  - `coderService` from 2.41 to 2.42

2.13

- **Changed**
  - `coderService` from 2.40 to 2.41

2.12

- **Changed**
  - `coderService` from 2.39 to 2.40

2.11

- **Changed**
  - `coderService` from 2.38 to 2.39
  - `colorComponent` from 1.0 to 1.1
  - `tsSource` from 1.4 to 1.5

2.10

- **Changed**

  - added list of vanc components to `MultiService`
  - `coderService` from 2.37 to 2.38

2.9

- **Changed**
  - `coderService` from 2.36 to 2.37

2.8

- **Changed**
  - `tsSource` from 1.3 to 1.4

2.7

- **Changed**
  - `coderService` from 2.34 to 2.35
  - `tsSource` from 1.2 to 1.3
  - `ipService` from 1.1 to 1.2

2.6

- **Added** `-logoInsertion` 1.0
- **Changed**
  - `coderService` from 2.33 to 2.34

2.5

- **Changed**
  - `coderService` from 2.32 to 2.33

2.4

- **Changed**
  - `coderService` from 2.31 to 2.32

2.3

- **Changed**
  - `coderService` from 2.30 to 2.31

2.2

- **Added**
  - `Scte35Config` 1.0
- **Changed**
  - `idrOnScte35` bool is now moved from `MultiServiceOptions` to `Scte35Options` in scte35Config 1.0
  - `scte35` contains Scte35Options instead of idrOnmScte35 bool
  - `coderService` from 2.29 to 2.30

2.1

- **Added**
  - `IpSource` struct containing an IpService and a list of slots in use

- **Changed**
  - `coderService` from 2.28 to 2.29
  - `ipService` from 1.0 to 1.1
  - added `colorComponent` import and colorProfile to MultiService
  - replaced `IpService` as ip variant in MsSource with newly added IpSource

2.0

- **Changed**
  - `coderService` from 2.27 to 2.28
  - added `ipService` as a possible multiservice source
  - moved `audioOptions` to MultiServiceOptions
  - replaced `audioOptions` on ServiceOptions with `audioSubscriptions` to refer to shared audioOptions
  - these changes are not backwards compatible

1.22

- **Added**
  - `latencyAdjustment` field added to AudioOptions struct for forwarding audio earlier than video.
- **Changed**
  - `coderService` from 2.26 to 2.27

1.21

- **Changed**
  - `coderService` from 2.25 to 2.26

40.2 Command Reference

40.2.1 SetMultiServices

- **message SetMultiServices.Request**
- **message SetMultiServices.Response**

40.2.2 GetMultiServices

- **message GetMultiServices.Request**
- **message GetMultiServices.Response**

40.2.3 DeleteMultiServices

- **message DeleteMultiServices.Request**
- **message DeleteMultiServices.Response**

40.3 Type Reference

40.3.1 AudioOptions

Audio options for a given Transcoder or ABR encoder audio track.

struct

source	**coderService.AudioSource** Source configuration for a track.
numAuPerPes	**optional int** Number of access units per PES, commonly referred to as PES alignment.
lipSyncAdjustment	**optional int** Lip sync adjustment
latencyAdjustment	**optional int** Adjust latency of audio relative to video.
levelAdjustment	**optional int** Level adjustment.
loudness	**optional coderService.Loudness** Loudness configuration.
destination	**optional coderService.TsAudio** Destination configuration for an ABR encoder.
passthrough	**optional bool** Flag based on the presence of an audio profile in the multi service profile for a given audio track; missing profiles indicates passthrough (true) here. ABR encoder only.

40.3.2 DeleteMultiServices.Request

struct

ids	**list** of **UUID**

40.3.3 DeleteMultiServices.Response

empty **struct**

40.3.4 GetMultiServices.Request

empty **struct**

40.3.5 GetMultiServices.Response

struct

data	**map** from **UUID** to **MultiServiceObj**
info	**GetMultiServices.Response.info**

40.3.6 GetMultiServices.Response.info

struct

totalEntries	**int**

40.3.7 IpSource

Configuration for IP sources.

struct

ipService	**ipService.IpService** Configuration for one or more IP video sources.
slot	**list** of **int** A list of slots utilizing the source.

40.3.8 MsSource

Multi Service Source. Configuration of the source of the Multi Service. Can be either ts, sdi or ip

variant

ts	**tsSource.TsSource** Configuration of the TS source. Transcoder only
sdi	**SdiSource** Configuration of the SDI source. ABR Encoder only
ip	**IpSource** Configuration of the IP source.

40.3.9 MultiService

Configuration for a Transcoder or ABR Encoder multi service.

struct

label	**string** A label for the service.
enabled	**bool** True if the service should be enabled.
source	**MsSource** Configuration of the source.
profile	**MultiService.profile** The profile that describes the Multi Service
options	**MultiServiceOptions** Configuration of components and IDR on SCTE35.
signalLoss	**optional** **testGenerator.SignalLoss** Configuration for the output when input signal is lost. ABR Encoder only
testPatternGenerator	**optional** **testGenComponent.TestGenComponent** Contains optional structure with link to a test generator profile. ABR Encoder only.
colorProfile	**optional** **colorComponent.ColorComponent** Contains optional structure with link to a color profile.
	list of **coderService.VancComponent**

vancs	A list of vanc components. SDI ABR Encoder only.

40.3.10 MultiService.profile

struct

id	**UUID**

40.3.11 MultiServiceObj

MultiService

40.3.12 MultiServiceOptions

Options for a given Transcoder or ABR encoder service.

struct

scte35	**scte35Config.Scte35Config** Configuration of SCTE 35.
audioOptions	**map** from **int** to **AudioOptions** Configuration of audio coding instances.
serviceOptions	**map** from **int** to **ServiceOptions** Configuration of component services.

40.3.13 SdiSource

Configuration for an ABR encoder.

struct

sdiConnection	**sdiConnection.SdiConnection** Configuration of the SDI connection of the source
slot	**int** Which slot the source comes from.

40.3.14 ServiceOptions

- Service options for a specific Transcoder or ABR encoder service.
-

struct

videoOptions	**optional VideoOptions** Video options (unused). •
audioSubscriptions	**list** of **int** References to the audio tracks selected for this service. These indentifiers must exist as keys in the map of AudioOptions. •

passthroughComponents	**optional bool** On transcoder only, setting this flag to true passes through all whitelisted transport stream components that are not transcoded by the service.

40.3.15 SetMultiServices.Request

struct

data	**map** from **UUID** to **MultiServiceObj**

40.3.16 SetMultiServices.Response

empty **struct**

40.3.17 VideoOptions

struct

logoInsertion	**optional logoInsertion.LogoInsertion** Contains optional structure with configuration for logo insertion.

41 multiServiceProfile (2.9)

41.1 Overview

Changelog

2.9

- **Changed**
  - `videoProfile` from 2.13 to 2.14
  - `expectedInput` from 1.7 to 1.8

2.8

- **Changed**
  - `audioProfile` from 2.6 to 2.7
  - `videoProfile` from 2.12 to 2.13
  - `expectedInput` from 1.6 to 1.7

2.7

- **Changed**
  - `videoProfile` from 2.11 to 2.12
  - `expectedInput` from 1.5 to 1.6

2.6

- **Changed**
  - `audioProfile` from 2.5 to 2.6

2.5

- **Changed**
  - `videoProfile` from 2.10 to 2.11

2.5

- **Changed**
  - `videoProfile` from 2.10 to 2.11

2.4

- **Changed**
  - `videoProfile` from 2.9 to 2.10
  - `expectedInput` from 1.3 to 1.4

2.3

- **Changed**
  - `audioProfile` from 2.4 to 2.5

2.2

- **Changed**
  - `videoProfile` from 2.8 to 2.9

  - `expectedInput` from 1.2 to 1.3

2.1

- **Changed**
  - `audioProfile` from 2.3 to 2.4

2.0

- **Changed**
  - moved `audios` to MultiServiceProfile
  - added `audioSubscriptions` to SubService to refer to shared `audios`
  - these changes are not backwards compatible

1.10

- **Changed**
- AudioTrack.profile field is changed from AudioProfileStruct to optional.
- `expectedInput` from 1.1 to 1.2

41.2 Command Reference

41.2.1 SetMultiServiceProfiles

- **message SetMultiServiceProfiles.Request**
- **message SetMultiServiceProfiles.Response**

41.2.2 GetMultiServiceProfiles

- **message GetMultiServiceProfiles.Request**
- **message GetMultiServiceProfiles.Response**

41.2.3 DeleteMultiServiceProfiles

- **message DeleteMultiServiceProfiles.Request**
- **message DeleteMultiServiceProfiles.Response**

41.3 Type Reference

41.3.1 AudioTrack

Audio Track configuration parameters.

struct

`uid`	**int** An audio track's unique id. Values are not required to be contigous.
`profile`	**optional audioProfile.AudioProfileStruct** A profile describing an audio encoding process. Empty optional signals passthrough.

41.3.2 DeleteMultiServiceProfiles.Request

struct

ids	**list** of **UUID**

41.3.3 DeleteMultiServiceProfiles.Response

empty **struct**

41.3.4 GetMultiServiceProfiles.Request

empty **struct**

41.3.5 GetMultiServiceProfiles.Response

struct

data	**map** from **UUID** to **MultiServiceProfileObj**
info	**GetMultiServiceProfiles.Response.info**

41.3.6 GetMultiServiceProfiles.Response.info

struct

totalEntries	**int**

41.3.7 MultiServiceProfile

Multiservice configuration parameters.

struct

profileName	**string** An text indentifier for the profile.
profileMode	**multiServiceProfileMode.ProfileMode** Flag for broadcast or ABR profile modes.
input	**expectedInput.ExpectedInput** Describes the expected input format of the video signal.
gopSize	**int** Describes the distances in frames between two keyframes.
aspectRatio	**videoProfile.AspectRatio** Describes the ratio of width to height of the video images.
frameRateDomain	**rawVideo.FrameRate** Describes the restricted set of frame rates these services should support.
audios	**list** of **AudioTrack** List of audio configurations for this profile. Each AudioTrack can be used in multiple SubServices and must be used by at least one SubService.
	list of **SubService**

subServices	List of subservice definitions.

41.3.8 MultiServiceProfileObj

MultiServiceProfile

41.3.9 SetMultiServiceProfiles.Request

struct

data	**map** from **UUID** to **MultiServiceProfileObj**

41.3.10 SetMultiServiceProfiles.Response

empty **struct**

41.3.11 SubService

Subservice configuration parameters.

struct

index	**int** Subservice unique id. Values are not required to be contigous.
video	**optional VideoTrack** Video parameters.
audioSubscriptions	**list** of **int** List of AudioTrack uid's to include in this subservice. These must be present in MultiServiceProfile.audios.

41.3.12 VideoTrack

Video Track configuration parameters.

struct

profile	**videoProfile.VideoProfileStruct** Describes a video coding process.

42 multiServiceProfileMode (1.0)

42.1 Type Reference

42.1.1 ProfileMode

enum

ABR
BROADCAST

43 nmosConfig (1.1)

43.1 Overview

```
# Changelog

## 1.1
- **Added**
    - Added new type `ConnectionType`
    - Added member 'connectionType' to NmosConfig
```

43.2 Type Reference

43.2.1 ConnectionType

Enum to control whether the NMOS traffic is in-band (through line card dataports) or out-of-band (through the MMI).

enum

IN_BAND	
OUT_OF_BAND	

43.2.2 NmosConfig

Configuration parameters for setting up NMOS.

struct

connectionType	**ConnectionType** Configuration to control whether traffic should be in-band or out-of-band
registry	**optional nmosRegistryConfig.NmosRegistryConfigStruct** Configuration parameters for connecting to the NMOS registry
labelTemplates	**optional nmosLabelConfig.NmosLabelConfig** Custom NMOS labels configuration. This feature is enabled when the optional is set

44 nmosLabelConfig (1.0)

44.1 Overview

Changelog

1.0

NMOS labelling

NMOS labelling API allows operator to define templates based on auto text options. This templates can be defined for for node, device, source, flow, sender and receiver.

Version 1.0 allows six auto text options.

What are the auto text options?

- ${SLOT} - The slot number of the card in use.
- ${DEV_ID} - The Device number. **Not supported for Node templates**.
- ${ESS_GR} - The label of the group where the essences belong to. **Not supported for Node templates**.
- ${ESS} - The label of the Essence in question. **Not supported for Node and Device templates**.
- ${ESS_T} - The Essence type: Video, Audio, Ancillary. This auto text option is substituted by what is defined in `EssenceTypeStrings` for each of the essences type. **Not supported for Node and Device templates**.
- ${ESS_T_INDEX} - The number of the essence belonging to one of the essence type - video, audio, ancillary. **Not supported for Node and Device templates**.

How to use the NMOS Labelling.

- NMOS templates are set in the fields part of `ResourceLabelTemplates`. Unused templates should be set with an empty string.
- Each field in `ResourceLabelTemplates` can have as many auto text options as the operator wants, as long as they are supported for the respective template.
- `EssenceTypeStrings` strings will substitute the auto text option ${ESS_T} with the defined string for the respective Essence type.

Example of strings to use for the fields in `ResourceLabelTemplates` **and** `EssenceTypeStrings`

- `nodeTemplate` - ECx210 Decoder ST2110 Sender Slot ${SLOT}
- `deviceTemplate` - ECx210 Decoder Sender Slot ${SLOT} ID ${DEV_ID}
- `sourceTemplate` - ECx210 Slot ${SLOT} ID ${DEV_ID}, ${ESS_T} Source: ${ESS}, Group: ${ESS_GR}, Essence Number: ${ESS_T_INDEX}
- `flowTemplate` - ECx210 Slot ${SLOT} ID ${DEV_ID}, ${ESS_T} Flow: ${ESS}, Group: ${ESS_GR}, Essence Number: ${ESS_T_INDEX}
- `senderTemplate` - ECx210 Slot ${SLOT} ID ${DEV_ID}, ${ESS_T} Sender: ${ESS}, Group: ${ESS_GR}, Essence Number: ${ESS_T_INDEX}
- `receiverTemplate` - ECx210 Slot ${SLOT} ID ${DEV_ID}, ${ESS_T} Receiver: ${ESS}, Group: ${ESS_GR}, Essence Number: ${ESS_T_INDEX}
- `video` - Video
- `audio` - Audio
- `ancillary` - Anc

44.2 Type Reference

44.2.1 EssenceTypeStrings

Labels for each essence type to use when generating custom NMOS labels.

struct

`video`	**string** Video essence label
`audio`	**string** Audio essence label
`ancillary`	**string** Ancillary essence label

44.2.2 NmosLabelConfig

Custom NMOS labels configuration.

struct

`resourceTemplates`	**ResourceLabelTemplates** Templates for each NMOS resource
`essenceStrings`	**EssenceTypeStrings** Labels to use for each essence type

44.2.3 ResourceLabelTemplates

NMOS label templates for each NMOS resource.

struct

`nodeTemplate`	**string** Template string for the node label
`deviceTemplate`	**string** Template string for the device label
`sourceTemplate`	**string** Template string for the source label
`flowTemplate`	**string** Template string for the flow label
`senderTemplate`	**string** Template string for the sender label
`receiverTemplate`	**string** Template string for the receiver label

45 nmosRegistryConfig (1.0)

45.1 Type Reference

45.1.1 DnsConfig

Method to use for discovery of registry server

variant

mDns	**DnsConfig.mDns** Use mDNS to reach the registry server
dnsSd	**DnsSdConfig** Configuration parameters to use DNS-SD for discovery of registry service

45.1.2 DnsConfig.mDns

empty **struct**

45.1.3 DnsSdConfig

Configuration parameters for DNS-SD

struct

dnsServer	**firewallTypes.IpAddress** IPv4 address of DNS-SD server
search	**string** Search domain name

45.1.4 NmosRegistryConfigStruct

Configuration parameters for reaching the NMOS registry

struct

registryConfig	**RegistryManualConfig** Configuration parameters to use in case of failure in mDNS or DNS-SD (secondary/fallback method)
dnsConfig	**DnsConfig** Configuration parameters for mDNS or DNS-SD (primary method)

45.1.5 RegistryManualConfig

Registry information to be used in case registry server can not be reached by mDNS or DNS-SD.

struct

fallbackEnable	**bool** Enable use of given information as a fallback
	firewallTypes.IpAddress

`registryIpAddr`	IPv4 address of NMOS registry
`registryPort`	**int** Port number of NMOS registry
`registryVersion`	**commonTypes.NmosVersionNumber** Version of registry API

46 nmosRegistryStatus (1.0)

46.1 Type Reference

46.1.1 NmosRegistryStatus

NMOS registry status

struct

`ipAddr`	**optional firewallTypes.SocketAddress** Currently used IP address and port number of the registry server. This value may have been obtained from DNS-SD, mDNS, or a manual IP entry.
`registered`	**bool** True if there is an active connection to the registry server.

47 nmosStatusTypes (1.2)

47.1 Type Reference

47.1.1 NmosStatus

NmosStatus Status of the NMOS service

struct

`nmosRegistryStatus`	**optional nmosRegistryStatus.NmosRegistryStatus** Status of the NMOS registry connection. Empty if no registry is configured.
`nodeApi`	**map** from **physicalIpPort.PortName** to **string** Map from port to the endpoint for NMOS Node API. Ex: PortName.D1 -> “10.20.10.34:8001”, PortName.D2_2 -> “10.20.11.34:8001”
`activeOutOfBandPath`	**optional int** The active out-of-band path. Optional: empty for in-band configuration, 0 if MMI slot 1, 1 if MMI slot 2.

48 physicalIpPort (1.0)

48.1 Type Reference

48.1.1 PortName

enum

CTRL
D1
D2
D3
D4
D1_1
D1_2
D1_3
D1_4
D2_1
D2_2
D2_3
D2_4
D3_1
D3_2
D3_3
D3_4
D4_1
D4_2
D4_3
D4_4

49 poisServerStatus (1.0)

49.1 Overview

Changelog

Version 1.0

- **Added**
  - `PoisServerStatus`
  - `GetPoisServerStatus`

49.2 Command Reference

49.2.1 GetPoisServerStatus

- **message GetPoisServerStatus.Request**
- **message GetPoisServerStatus.Response**

49.3 Type Reference

49.3.1 GetPoisServerStatus.Request

empty **struct**

49.3.2 GetPoisServerStatus.Response

struct

`data`	**PoisServerStatus**

49.3.3 PoisServerStatus

Status of POIS server

struct

`connectionStatus`	**esamStatus.PoisConnectionStatus** Status of the connection to the server

50 poolConfig (1.1)

50.1 Overview

Changelog

1.1

- **Changed**
  - included `esamConfig` in `PoolConfig` struct

50.2 Command Reference

50.2.1 GetPoolConfig

- **message GetPoolConfig.Request**
- **message GetPoolConfig.Response**

50.2.2 SetPoolConfig

- **message SetPoolConfig.Request**
- **message SetPoolConfig.Response**

50.3 Type Reference

50.3.1 GetPoolConfig.Request

empty **struct**

50.3.2 GetPoolConfig.Response

struct

`data`	**PoolConfigObj**

50.3.3 PoolConfig

struct

`spareModules`	**optional int**
`esamConfig`	**optional esamConfig.EsamConfig**

50.3.4 PoolConfigObj

PoolConfig

50.3.5 SetPoolConfig.Request

struct

data	**PoolConfigObj**

50.3.6 SetPoolConfig.Response

empty **struct**

51 poolCtrlAllocs (1.2)

51.1 Overview

Changelog:

1.2

- **Added**
  - changed NotifyServiceFlowMap to map id to FlowAndValidity
  - new struct FlowAndValidity
  - new internal message: SetFlowState
  - Rename all instances of sensitive words to `leader/follower`

1.1

- **Added**
  - Constant base UUID used to generate deterministic UUIDs for video alignement groups.
- **Changed**
  - generalization of the video alignment in order to work for both SDI and IP: `VideoAlignGroupMaster`, `VideoAlignGroupMasterObj` and`FlowStatus` should replace `VideoAlignGroupMasters`, `Allocations` and `Distribution`
  - `SlotPort` with member `port` changed to `SlotFlow` with member `flow`

51.2 Type Reference

51.2.1 FlowAndValidity

struct

`flow`	**int**
`valid`	**bool**

51.2.2 SlotFlow

struct

`slot`	**int**
`flow`	**int**

51.2.3 VideoAlignGroupLeader

struct

`groupIndex`	**int**
`leader`	**UUID**
`flow`	**SlotFlow**

51.2.4 VideoAlignGroupLeaderObj

VideoAlignGroupLeader

52 psiStatus (1.6)

52.1 Overview

Changelog

1.6

- **Changed**
  - `psiTypes` from 1.5 to 1.6

1.5

- **Changed**
  - `psiTypes` from 1.4 to 1.5

52.2 Type Reference

52.2.1 PsiStatus

struct

`components`	**list** of **psiTypes.Component**
`name`	**optional string**

53 psiTypes (1.6)

53.1 Overview

Changelog

1.6

- **Changed**
  - Component.bitrate changed from int to bigint. Necessary for D

1.5

- **Changed**
- `component` from 1.3 to 1.5

53.2 Type Reference

53.2.1 Component

struct

`flowId`	**int**
`pid`	**Pid**
`lang`	**Lang**
`componentType`	**component.ComponentType**
`streamType`	**int**
`isPcr`	**bool**
`cas_ids`	**list** of **int**
`ccError`	**int**
`bitrate`	**bigint**

53.2.2 Components

list of **Component**

53.2.3 Lang

string

53.2.4 Pid

int

54 rawAudio (1.2)

54.1 Overview

```
# Changelog
## 1.2
- **Added**
  - `SURROUND_514`, `SURROUND_714` and `SURROUND_916` to `ChannelMode`.
```

54.2 Type Reference

54.2.1 ChannelMode

enum

MONO
DUALMONO
STEREO
SURROUND_31
SURROUND_51
SURROUND_71
SURROUND_514
SURROUND_714
SURROUND_916

54.2.2 Hack

struct

profile	**int**

55 rawVideo (1.5)

55.1 Type Reference

55.1.1 ChromaSampling

enum

CS_422
CS_420

55.1.2 FrameRate

enum

FPS_25
FPS_29_97
FPS_30
FPS_50
FPS_59_94
FPS_60

55.1.3 Hack

struct

profile	**int**

55.1.4 HorizontalResolution

enum

H_3840
H_2560
H_1920
H_1440
H_1280
H_1024
H_960
H_854
H_848
H_768
H_720
H_704
H_640
H_544
H_528
H_512
H_480
H_400

H_384
H_320
H_256
H_192
H_416

55.1.5 VerticalResolution

enum

V_2160
V_1440
V_1080
V_720
V_640
V_576
V_540
V_480
V_432
V_396
V_384
V_360
V_288
V_270
V_224
V_216
V_192
V_180
V_144
V_240

56 redundancyGroup (1.0)

56.1 Overview

Changelog

1.0

- **Added**
  - added RedundancyMode
  - added RedundancyType
  - added RedundancyCard
  - added RedundancyGroupStruct

56.2 Command Reference

56.2.1 GetRedundancyGroups

- **message GetRedundancyGroups.Request**
- **message GetRedundancyGroups.Response**
- **message GetRedundancyGroups.Failure**

56.2.2 SetRedundancyGroups

- **message SetRedundancyGroups.Request**
- **message SetRedundancyGroups.Response**
- **message SetRedundancyGroups.Failure**

56.2.3 DeleteRedundancyGroups

- **message DeleteRedundancyGroups.Request**
- **message DeleteRedundancyGroups.Response**
- **message DeleteRedundancyGroups.Failure**

56.3 Type Reference

56.3.1 DeleteRedundancyGroups.Failure

empty **struct**

56.3.2 DeleteRedundancyGroups.Request

struct

`ids`	**list** of **UUID**

56.3.3 DeleteRedundancyGroups.Response

empty **struct**

56.3.4 GetRedundancyGroups.Failure

empty **struct**

56.3.5 GetRedundancyGroups.Request

empty **struct**

56.3.6 GetRedundancyGroups.Response

struct

`data`	**map** from **UUID** to **RedundancyGroup**

56.3.7 RedundancyCard

N+M redundancy configuration for a single card

struct

`slot`	**int** The slot the card resides in
`redundancyType`	**RedundancyType** Specifies whether the card is a main or backup card
`disabled`	**bool** True if the card has been disabled

56.3.8 RedundancyGroup

RedundancyGroupStruct

56.3.9 RedundancyGroupStruct

N+M redundancy group configuration

struct

`label`	**string** A label for this group
`router`	**optional UUID** The SDI router connected to the cards in this group
`mode`	**RedundancyMode** The redundancy mode used for this group
`cards`	**list** of **RedundancyCard** List of cards that are part of this group

56.3.10 RedundancyMode

The supported N+M redundancy modes

enum

FLOATING
REVERTING

56.3.11 RedundancyType

The roles a card can have in a redundancy group

enum

MAIN
BACKUP

56.3.12 SetRedundancyGroups.Failure

empty **struct**

56.3.13 SetRedundancyGroups.Request

struct

data	**map** from **UUID** to **RedundancyGroup**

56.3.14 SetRedundancyGroups.Response

empty **struct**

57 redundancyGroupStatus (1.2)

57.1 Overview

Changelog

1.2

- **Added**
  - added redundancyStateToString

1.1

- **Added**
  - added ProtoRedundancyGroupStatus

57.2 Command Reference

57.2.1 GetRedundancyGroupStatus

- **message GetRedundancyGroupStatus.Request**
- **message GetRedundancyGroupStatus.Response**

57.3 Type Reference

57.3.1 GetRedundancyGroupStatus.Request

empty **struct**

57.3.2 GetRedundancyGroupStatus.Response

struct

`data`	**map** from **UUID** to **RedundancyGroupStatus**

57.3.3 RedundancyCardStatus

N+M redundancy status for a single card

struct

`state`	**RedundancyState** The current state of the card
`activeConfig`	**optional Slot** The active configuration of the card. A slot means that the services on that slot was moved to this card.

57.3.4 RedundancyGroupStatus

N+M redundancy status for a redundancy group

struct

`status`	**map** from **Slot** to **RedundancyCardStatus** Redundancy status for each card in this group

57.3.5 RedundancyState

The states a single card can be in.

enum

`UP`
`DOWN`
`DISABLED`

57.3.6 Slot

int

58 scte35Config (1.1)

58.1 Overview

Changelog

1.1

- **Added**
  - Added heartbeat: optional(int)

58.2 Type Reference

58.2.1 Scte35Config

Config for a Scte35 configuration.

struct

`idrOnScte35`	**bool** Enables IDR insertion on SCTE 35 splice points.
`esamConfig`	**optional esamConfig.EsamServiceConfig** ESAM configuration. A set optional indicates that ESAM is enabled.
`heartbeat`	**optional int** Heartbeat interval in seconds if set.

59 scte35LogApi (1.0)

59.1 Command Reference

59.1.1 GetScte35History

- message GetScte35History.Request
- message GetScte35History.Response

59.2 Type Reference

59.2.1 GetScte35History.Request

struct

serviceId	UUID
slot	int

59.2.2 GetScte35History.Response

struct

scte35History	string

60 sdiConnection (1.0)

60.1 Overview

New in v1.0:

- These types were factored out from coderService

60.2 Type Reference

60.2.1 SdiConnection

struct

sdiType	**sdiType.SdiType**
connectors	**SdiConnectors**

60.2.2 SdiConnectors

enum

SDI_1A
SDI_1B
SDI_1C
SDI_1D
SDI_2A
SDI_2B
SDI_2C
SDI_2D
SDI_1ABCD
SDI_2ABCD
SDI_1AC
SDI_1BD
SDI_2AC
SDI_2BD
UNDEFINED

61 sdiRouterConfig (1.1)

61.1 Overview

Changelog

1.1

- **Added**
  - added brand SWP08
  - added port to SdiRouterConfig

1.0

- **Added**
  - added SdiRouterStruct
  - added SdiRouterBrand
  - added CardConnector

61.2 Command Reference

61.2.1 GetSdiRouters

- **message GetSdiRouters.Request**
- **message GetSdiRouters.Response**
- **message GetSdiRouters.Failure**

61.2.2 SetSdiRouters

- **message SetSdiRouters.Request**
- **message SetSdiRouters.Response**
- **message SetSdiRouters.Failure**

61.2.3 DeleteSdiRouters

- **message DeleteSdiRouters.Request**
- **message DeleteSdiRouters.Response**
- **message DeleteSdiRouters.Failure**

61.3 Type Reference

61.3.1 CardConnector

Holds slot and connector which is enough to specify a distinct input port in a chassis.

struct

slot	**int**
connector	**sdiConnection.SdiConnectors**

61.3.2 DeleteSdiRouters.Failure

empty **struct**

61.3.3 DeleteSdiRouters.Request

struct

`ids`	**list** of **UUID**

61.3.4 DeleteSdiRouters.Response

empty **struct**

61.3.5 GetSdiRouters.Failure

empty **struct**

61.3.6 GetSdiRouters.Request

empty **struct**

61.3.7 GetSdiRouters.Response

struct

`data`	**map** from **UUID** to **SdiRouter**

61.3.8 Label

string

61.3.9 Port

int

61.3.10 SdiRouter

SdiRouterStruct

61.3.11 SdiRouterBrand

The different SDI router brands we support

enum

```
BLACKMAGIC
NEVION
SWP08
```

61.3.12 SdiRouterStruct

SDI router configuration

struct

`label`	**string** A label for the SDI router
`ip`	**firewallTypes.IpAddress** The IP address of the router
`port`	**optional int** The port of the router
`brand`	**SdiRouterBrand** The router brand
`numInputs`	**int** The number of SDI inputs the router has
`numOutputs`	**int** The number of SDI outputs the router has
`inputs`	**map** from **Port** to **Label** Label for each input port
`outputs`	**map** from **Port** to **Label** Label for each output port
`outputMapping`	**map** from **Port** to **CardConnector** Maps router output port to coder slot and connector

61.3.13 SetSdiRouters.Failure

empty **struct**

61.3.14 SetSdiRouters.Request

struct

`data`	**map** from **UUID** to **SdiRouter**

61.3.15 SetSdiRouters.Response

empty **struct**

62 sdiRouterConnection (1.0)

62.1 Overview

Changelog

1.0

- **Added**
  - added SdiRouterConnectionStruct

62.2 Command Reference

62.2.1 SetSdiRouterConnections

- message **SetSdiRouterConnections.Request**
- message **SetSdiRouterConnections.Response**
- message **SetSdiRouterConnections.Failure**

62.3 Type Reference

62.3.1 InputPort

int

62.3.2 OutputPort

int

62.3.3 SdiRouterConnectionStruct

Keeps track of the internal routing in a SDI router

struct

`routerId`	**UUID** The ID of the SDI router
`routing`	**map** from **OutputPort** to **InputPort** Map of which input is available on each output

62.3.4 SetSdiRouterConnections.Failure

empty **struct**

62.3.5 SetSdiRouterConnections.Request

struct

`data`	**map** from **UUID** to **SdiRouterConnectionStruct**

62.3.6 SetSdiRouterConnections.Response

empty **struct**

63 sdiRouterStatus (1.0)

63.1 Overview

Changelog

1.0

- **Added**
  - added SdiRouterConfigStatus
  - added GetSdiRouterConfigStatuses

63.2 Command Reference

63.2.1 GetSdiRouterStatuses

- message **GetSdiRouterStatuses.Request**
- message **GetSdiRouterStatuses.Response**

63.3 Type Reference

63.3.1 GetSdiRouterStatuses.Request

empty **struct**

63.3.2 GetSdiRouterStatuses.Response

struct

`data`	**map** from **UUID** to **SdiRouterStatus**

63.3.3 InputPort

int

63.3.4 Label

string

63.3.5 OutputPort

int

63.3.6 SdiRouterStatus

Status for a connected SDI router

struct

	map from **OutputPort** to **InputPort**

`routing`	The currently configured routing for each output on the router
`inputLabels`	**map** from **InputPort** to **Label** The labels for the inputs on the switch
`outputLabels`	**map** from **OutputPort** to **Label** The labels for the outputs on the switch
`link`	**bool**

64 sdiStatus (1.12)

64.1 Overview

Changelog

1.12

- **Changed**
  - `SdiVideoModes` moved to atv_definitions sdiVideoModes

1.11

- **Changed**
  - `codedAudio` from 1.0 to 1.1

1.10

- **Added**
  - VPID status to SDI status

1.9

- **Added**
  - SdiAudioStatus analysedMaxLevel and analysedCodec.

1.8

- **Added**
  - SDI_2160P25, SDI_2160P30, SDI_2160P30_1001 to the SdiVideoModes enum

64.2 Type Reference

64.2.1 AesChannelMode

enum

MONO
STEREO
MULTI
N_A
UNKNOWN

64.2.2 AesData

struct

mode	**AesMode**
channelMode	**AesChannelMode**
crcOk	**bool**

64.2.3 AesMode

enum

PCM	
NON_PCM	

64.2.4 InstanceId

int

64.2.5 QuadMode

enum

SDI_4X3G_SQUARE	
SDI_4X3G_INTERL	

64.2.6 QuadStatus

struct

lock	**bool**
lockLoss	**int**
skew	**float**

64.2.7 SdiAudioSampleRate

enum

SYNC_48_kHz	
ASYNC_48_kHz	
SYNC_44_1_kHz	
ASYNC_44_1_kHz	
SYNC_32_kHz	
ASYNC_32_kHz	
SYNC_FREE_RUN	
ASYNC_FREE_RUN	

64.2.8 SdiAudioStatus

SDI input audio status on an encoder.

struct

aesChannel	**int** Embedded audio channels.
active	**bool** Indicates if a channel is active meaning that there exists a valid signal on the input.
lossCnt	**int** Number of lost packets during the last second.
sampleRate	**SdiAudioSampleRate** Audio sample rate.

sampleCnt	**int** Actual number of samples received during the last second.
errors	**bool** True if any errors during the last second, false otherwise.
aesData	**optional AesData** Embedded audio channel data.
analysedMaxLevel	**optional float** Analysed input max level.
analysedCodec	**optional codedAudio.AudioCodec** Analysed input codec.

64.2.9 SdiStatus

SDI input status on an encoder, and SDI output status on a decoder.

struct

id	**InstanceId** Used with quad SDI input. IDs 0-3 correspond to signals on 1A-D and 4-7 correspond to 2A-D.
videoLockInput	**bool** True if the encoder has locked onto a signal, false otherwise.
videoMode	**sdiVideoModes.SdiVideoModes** Input video mode on encoder, and output video mode on the decoder.
edhErrorCount	**int** Number of EDH errors on the input signal.
audio	**list** of **SdiAudioStatus** List of audio statuses for each audio channel in the SDI signal.
quad	**optional QuadStatus** Quad status for this SDI instance. Only available with quad input.
quadMode	**optional QuadMode** Which mode was used to generate the quad signal. Only available with quad input.
transceiverStatus	**optional transceiverStatus.TransceiverStatus** Status from the transceiver. Only available on encoder.
vpidStatus	**optional vpidInfo.ColorInfo** VPID color info going out of the decoder, and coming in on the encoder.

65 sdiType (1.0)

65.1 Type Reference

65.1.1 SdiType

enum

SDI_3G
SDI_12G
SDI_4X3G_QUANT
SDI_4X3G_SQUARE
SDI_4X3G_INTERL

66 sdiVideoModes (1.0)

66.1 Overview

```
# Changelog
## 1.0
- **Added**
  - `SdiVideoModes`
  - `SdiVideoModes2str` ToString conversion for `SdiVideoModes`
```

66.2 Type Reference

66.2.1 SdiVideoModes

enum

SDI_OFF
SDI_480I
SDI_480P
SDI_576I
SDI_576P
SDI_720P24
SDI_720P24_1001
SDI_720P25
SDI_720P30
SDI_720P30_1001
SDI_720P50
SDI_720P60
SDI_720P60_1001
SDI_1080P24
SDI_1080P24_1001
SDI_1080P25
SDI_1080P30
SDI_1080P30_1001
SDI_1080I50
SDI_1080I60
SDI_1080I60_1001
SDI_1080P50
SDI_1080P60
SDI_1080P60_1001
SDI_2160P50
SDI_2160P60
SDI_2160P60_1001
SDI_2160P25
SDI_2160P30
SDI_2160P30_1001
ANY

67 sdpUpload (1.1)

67.1 Overview

Changelog

1.1

- **Changed**
  - `profileId` added to `SdpData` struct for use with Profile Redundancy

67.2 Command Reference

67.2.1 UploadSdpData

- **message UploadSdpData.Request**
- **message UploadSdpData.Response**

67.3 Type Reference

67.3.1 SdpData

- SDP data for storing the SDP file uploaded through GUI.
- This file defines the external RPC interface for uploading SDP files in order to change an IP Profile.
-

struct

`serviceId`	**UUID** The uuid of the service for which the SDP file is uploaded. •
`profileId`	**optional UUID** The uuid of the profile for which the SDP file is uploaded. If profileId is empty, the uploaded SDP should belong to the first profile (index 0). •
`slot`	**int** The slot on which this SDP data will be used. •
`sdpInfo`	**map** from **UUID** to **string** The SDP file stored with a unique uuid.

67.3.2 UploadSdpData.Request

SdpData

67.3.3 UploadSdpData.Response

empty **struct**

68 serviceStatus (2.48)

68.1 Overview

Changelog:

2.48

- **Changed**
  - audioStatus from 2.16 to 2.17
  - videoStatus from 2.17 to 2.18

2.47

- **Changed**
  - sdiStatus from 1.11 to 1.12

2.46

- **Changed**
  - audioStatus from 2.15 to 2.16
  - st2110Status from 1.5 to 1.6
  - videoStatus from 2.16 to 2.17

2.45

- **Changed**
  - videoStatus from 2.15 to 2.16
  - vancStatus from 1.13 to 1.14

2.44

- **Changed**
  - lockStatus from 1.4 to 1.5

2.43

- **Changed**
  - lockStatus from 1.3 to 1.4
  - audioStatus from 2.14 to 2.15
  - st2110Status from 1.4 to 1.5
  - sdiStatus from 1.10 to 1.11

2.42

- **Changed**
  - videoStatus from 2.14 to 2.15
  - vancStatus from 1.12 to 1.13

2.41

- **Changed**
  - videoStatus from 2.13 to 2.14

2.40

- **Changed**
  - videoStatus from 2.12 to 2.13
  - sdiStatus from 1.9 to 1.10
  - vancStatus from 1.11 to 1.12

2.39

- **Changed**
  - vancStatus from 1.10 to 1.11
  - audioStatus from 2.13 to 2.14
  - st2110Status from 1.3 to 1.4

2.38

- **Changed**
  - vancStatus from 1.9 to 1.10

2.37

- **Changed**
  - videoStatus from 2.11 to 2.12

2.36

- **Changed**
  - vancStatus from 1.8 to 1.9

2.35

- **Changed**
  - st2110Status from 1.2 to 1.3
  - audioStatus from 2.12 to 2.13
  - vancStatus from 1.7 to 1.8

2.34

- **Changed**
  - audioStatus from 2.11 to 2.12
  - sdiStatus from 1.8 to 1.9

2.33

- **Changed**
  - vancStatus from 1.6 to 1.7

2.32

- **Changed**
  - vancStatus from 1.5 to 1.6

2.31

- **Changed**

- `lockStatus` from 1.2 to 1.3

2.30

- **Changed**
  - `videoStatus` from 2.10 to 2.11
  - `psiStatus` from 1.5 to 1.6
  - `inputRedundancyStatus` from 1.2 to 1.3

2.29

- **Added**
  - New struct `St2110ProfileStatus` added, replacing `st2110` list in `ServiceStatus`

2.28

- **Changed**
  - `sdiStatus` from 1.7 to 1.8

2.27

- **Changed**
  - `audioStatus` from 2.10 to 2.11
  - `psiStatus` from 1.4 to 1.5
  - `inputRedundancyStatus` from 1.1 to 1.2

2.26

- **Changed**
  - `ipStatus` from 1.6 to 1.7

68.2 Command Reference

68.2.1 GetServiceStatus

- **message GetServiceStatus.Request**
- **message GetServiceStatus.Response**

68.3 Type Reference

68.3.1 GetServiceStatus.Request

ServiceStatusRequest

68.3.2 GetServiceStatus.Response

ServiceStatusResponse

68.3.3 Jpeg2000Status

struct

`tr01Mode`	**videoProfile.Jpeg2000Profile**

68.3.4 ServiceStatus

Status for a specific service

struct

video	**optional** **videoStatus.VideoStatus** Status for resulting encoded or decoded video.
audio	**optional** **audioStatus.AudioStatus** Audio status
sdi	**map** from **int** to **sdiStatus.SdiStatus** SDI input status per channel (also used for IP-2022-Encoder)
st2110	**optional** **St2110ProfileStatus** ST-2110 specific status; list of essences and uuid of the active ip profile
jpeg2000	**optional** **Jpeg2000Status** specific status; TR01 mode recieved from input.
lock	**optional** **lockStatus.LockStatus** Lock status (genlock etc.)
vanc	**optional** **vancStatus.VancStatus** Status for the supported VANC protocols (teletext etc.)
ip	**map** from **int** to **ipStatus.IpStatus** IP status
components	**optional** **psiStatus.PsiStatus** Status for component passthrough (transcoder)
latency	**optional** **int** Total latency of the service in ms (encoder only)
tsInputRedundancy	**optional** **inputRedundancyStatus.TsInputRedundancyStatus**

68.3.5 ServiceStatusQuery

struct

id	**UUID**
slot	**int**

68.3.6 ServiceStatusRequest

struct

query	**list** of **ServiceStatusQuery**

68.3.7 ServiceStatusResponse

struct

data	**map** from **UUID** to **ServiceStatus**

68.3.8 St2110ProfileStatus

struct

st2110Essences	**list** of **st2110Status.St2110Status**
profileId	**UUID**

69 sfpConstants (1.4)

69.1 Overview

Changelog

1.4

- **Changed**
- Changed “SFP or SFP+” to “SFP/SFP+/SFP28” The corresponding identifier refers to the given 3 SFP types as per SNIA SFF-8024 specification.

69.2 Type Reference

69.2.1 Identifier

Physical Device Identifier Values. (SFF 8472 Rev12.3 spec Table 5-1)

enum

UNKNOWN	Unknown or unspecified
GBIC	GBIC
SOLDERED	Module soldered to motherboard (ex. SFF)
SFP_SFPP	SFP, SFP+ or SFP28
QSFPP	QSFP+
QSFP28	QSFP28
NOT_USED	Not used by SFF 8472 Rev12.2.1. These values are maintained in the Transceiver Management section of SFF-8024.
VENDOR	Vendor specific

69.2.2 Transceiver

Transceiver Compliance Codes. (SFF 8472 Rev12.3 spec Table 5-3)

enum

ER_10G	10G Base-ER
LRM_10G	10G Base-LRM
LR_10G	10G Base-LR
SR_10G	10G Base-SR
ACTIVE_CABLE_40G	40G Active cable
LR4_40G	40G Base-LR4
SR4_40G	40G Base-SR4
CR4_40G	40G Base-CR4
PX	BASE-PX
BX10	BASE-BX10
FX_100	100BASE-FX
LX_100	100BASE-LX/LX10
T_1000	1000BASE-T
CX_1000	1000BASE-CX
LX_1000	1000BASE-LX
SX_1000	1000BASE-SX

ELECTRICAL_INTER_ENCLOSURE	Electrical inter enclosure
LONGWAVE_LASER_LL	Longwave laser LL
MEDIUM	Medium distance
LONG_DISTANCE	Long distance
INTERMEDIATE_DISATANCE	Intermediate distance
SHORT_DISTANCE	Short distance
VERY_LONG_DISTANCE	Very long distance
ACTIVE_CABLE	Active Cable
PASSIVE_CABLE	Passive Cable
LONGWAVE_LASER_LC	Longwave laser LC
SHORTWAVE_LASER_w_OFC	Shortwave laser w OFC
SHORTWAVE_LASER_wo_OFC	Shortwave laser w/o OFC
ELECTRICAL_INTRA_ENCLOSURE	Electrical intra enclosure
SINGLE_MODE	Singel mode
MULTIMODE_50um	Multimode 50um
MULTIMODE_50m	Multimode 50m
MULTIMODE_625m	Multimode 62.6m
VIDEO_COAX	Video coax
MINIATURE_COAX	Miniature coax
SHIELDED_TWISTED_PAIR	Shielded twisted pair
TWIN_AXIAL_PAIR	Twin axial pair
M_BYTES_SEC_100	100M bytes per sec
M_BYTES_SEC_200	200M bytes per sec
M_BYTES_SEC_400	400M bytes per sec
M_BYTES_SEC_1600	1600M bytes per sec
M_BYTES_SEC_800	800M bytes per sec
M_BYTES_SEC_1200	1200M bytes per sec Extended Specification Compliance. (SFF-8024 Rev 4.8 spec Table 4-4)
AOC_100G_AOC_25GAUI_C2M_BER_5_	100G AOC (Active Optical Cable) or 25GAUI C2M AOC. Providing a worst BER of $5 * 10^{-5}$
SR4_100G_BASE_SR_25G_BASE	100GBASE-SR4 or 25GBASE-SR
LR4_100G_BASE_LR_25G_BASE	100GBASE-LR4 or 25GBASE-LR
ER4_100G_BASE_ER_25G_BASE	100GBASE-ER4 or 25GBASE-ER
SR10_100G_BASE	100GBASE-SR10
CWDM4_100G	100G CWDM4
PSM4_PARALLEL_SMF_100G	100G PSM4 Parallel SMF
ACC_100G_ACC_25GAUI_C2M_BER_5_	100G ACC (Active Copper Cable) or 25GAUI C2M ACC. Providing a worst BER of $5 * 10^{-5}$
CR4_100G_BASE_25G_BASE_CR_CA_2	100GBASE-CR4 or 25GBASE-CR CA-L
CR_25G_BASE_CA_25G_S_50G_BASE_	25GBASE-CR CA-S
CR_25G_BASE_CA_25G_N_50G_BASE_	25GBASE-CR CA-N
ER4_40G_BASE	40GBASE-ER4
SR_4_X_10G_BASE	4 x 10GBASE-SR
PSM4_PARALLEL_SMF_40G	40G PSM4 Parallel SMF
G959_1_PROFILE_P1I1_2D1	G959.1 profile P1I1-2D1 (10709 MBd, 2km, 1310nm SM)
G959_1_PROFILE_P1S1_2D2	G959.1 profile P1S1-2D2 (10709 MBd, 40km, 1550nm SM)
G959_1_PROFILE_P1L1_2D2	G959.1 profile P1L1-2D2 (10709 MBd, 80km, 1550nm SM)
T_10G_BASE	10GBASE-T with SFI electrical interface

CLR4_100G	100G CLR4
AOC_100G_AOC_25GAUI_C2M_BER_10	100G AOC or 25GAUI C2M AOC. Providing a worst BER of 10^{-12} or below
ACC_100G_ACC_25GAUI_C2M_BER_10	100G ACC or 25GAUI C2M ACC. Providing a worst BER of 10^{-12} or below
DWDM2_100GE	100GE-DWDM2 (DWDM transceiver using 2 wavelengths on a 1550nm DWDM grid with a reach up to 80km)

70 sfpStatus (1.5)

70.1 Overview

Changelog

1.5

- **Changed**
  - `sfpConstants` from 1.3 to 1.4

70.2 Type Reference

70.2.1 SfpDiagnostics

Real Time Diagnostic for SFP/SFP+/QSFP28/QSFP+, with A2h/A0h memory space address in square brackets. (SFF 8472 Rev12.3 spec Table 9-11) (SFF 8636 Rev2.10 spec Table 6-1)

struct

`temp`	**float** Internally measured module temperature (°C). [96-97], [22-23]
`vcc`	**float** Internally measured supply voltage in transceiver (V). [98-99], [26-27]
`txPwr`	**list** of **float** Measured TX output power (mW). [102-103], [50-57]
`rxPwr`	**list** of **float** Measured RX input power (mW). [104-105], [34-41]

70.2.2 SfpStatus

SFP Status Data Fields, with A0h memory space address in square brackets. (SFF 8472 Rev12.3 spec Table 4-1) QSFP A0h memory space address is specified next to the SFP A0h. (SFF 8636 Rev 2.10 spec Table 6-14)

struct

`identifier`	**sfpConstants.Identifier** Type of transceiver (SFF 8024 Rev4.8a spec Table 4-1) [0],[128]
`transceiver`	**list** of **sfpConstants.Transceiver** Code for electronic or optical compatibility [3-10], [131-138]
`brNominal`	**float** Nominal signalling rate, units of 100 MBd. [12], [140/222]
`vendorName`	**string** SFP vendor name. [20-35], [148-163]
`vendorOUI`	**string** SFP vendor IEEE company ID. [37-39], [165-167]
`vendorPN`	**string** Part number provided by SFP vendor. [40-55], [168-183]
	string

`vendorRev`	Revision level for part number provided by vendor. [56-59], [184-185]
`wavelength`	**string** Laser wavelength. [60-61], [186-187]
`vendorSN`	**string** Serial number provided by SFP vendor. [68-83], [196-211]
`dateCode`	**string** Vendor's manufacturing date code. [84-91], [212-219]
`diagnostics`	**optional SfpDiagnostics** Optional Diagnostics Monitor Data.

71 slateInsertion (1.0)

71.1 Overview

Changelog

1.0

- **Added**
  - First version of module

71.2 Type Reference

71.2.1 SlateInsertion

Configure slate insertion for a CoderService.

struct

`trigger`	**SlateInsertionTrigger** What events will trigger slate insertion
`slateId`	**string** A unique identifier for the slate

71.2.2 SlateInsertionTrigger

enum

`DPI`
`FORCED`

72 slotIpInterface (1.6)

72.1 Overview

- This file defines the external RPC interface to forward
- GetSlotIpInterface request to the correct card.

Changelog:

1.6

- **Changed**
  - `ipInterface` from 1.5 to 1.6

1.5

- **Changed**
  - `ipInterface` from 1.4 to 1.5

72.2 Command Reference

72.2.1 GetSlotIpInterfaces

- **message GetSlotIpInterfaces.Request**
- **message GetSlotIpInterfaces.Response**
- **message GetSlotIpInterfaces.Failure**

72.3 Type Reference

72.3.1 GetSlotIpInterfaces.Failure

empty **struct**

72.3.2 GetSlotIpInterfaces.Request

IpInterfaceRequest

72.3.3 GetSlotIpInterfaces.Response

struct

`data`	**map** from **UUID** to **ipInterface.IpInterface**

72.3.4 IpInterfaceRequest

struct

`slot`	**int**

73 st2110Status (1.6)

73.1 Overview

Changelog

1.6

- **Changed**
  - `audioProfile` from 2.6 to 2.7

1.4

- **Changed**
  - `audioProfile` from 2.5 to 2.6

1.4

- **Changed**
  - `audioProfile` from 2.4 to 2.5

1.3

- **Changed**
  - `st2110Types` from 1.0 to 1.2.
  - `audioProfile` from 2.3 to 2.4
- **Removed**
  - `St2110AudioPacketTime` moved to `st2110Types` 1.2.

73.2 Type Reference

73.2.1 St2110AncStatus

struct

`ipStreamUuid`	**UUID**
`standard`	**st2110Types.St2110Standard**
`sdpStatus`	**list** of **string**

73.2.2 St2110AudioStatus

struct

`ipStreamUuid`	**UUID**
`standard`	**st2110Types.St2110Standard**
`packetTime`	**st2110Types.St2110AudioPacketTime**
`totalChannels`	**int**
`sampleRate`	**optional audioProfile.SampleRate**
`sdpStatus`	**list** of **string**

73.2.3 St2110Status

variant

videoStatus	St2110VideoStatus
audioStatus	St2110AudioStatus
ancStatus	St2110AncStatus

73.2.4 St2110VideoStatus

struct

ipStreamUuid	UUID
standard	st2110Types.St2110Standard
sdpStatus	list of string

74 st2110Types (1.2)

74.1 Overview

```
# Changelog

## 1.2
- **Added**
 - `St2110AudioPacketTime`
```

74.2 Type Reference

74.2.1 St2110AudioPacketTime

Valid audio packet time according to SMPTE ST2110.

enum

N_A	Not applicable
US83_3	83.3 us
US125	125 us
MS1	1ms

74.2.2 St2110Standard

The SMPTE ST2110 standards that describe video, audio or ancillary content.

enum

ST2110_20	ST2110 uncompressed video standard
ST2110_22	ST2110 compressed video standard
ST2110_30	ST2110 audio standard
ST2110_31	ST2110 AES3 formatted audio
ST2110_40	ST2110 ancillary standard

75 streamSource (1.0)

75.1 Type Reference

75.1.1 StreamSource

StreamSource types of stream sources.

enum

MAIN	the main stream source.
BACKUP	the backup stream source.

76 subtitling (1.2)

76.1 Overview

Changelog

1.2

- **Added**
  - New enum `HorizontalAlignment`.

1.1

- **Changed**
  - `component` from 1.3 to 1.5

76.2 Type Reference

76.2.1 HorizontalAlignment

enum

`FOLLOW_INPUT`
`LEFT`
`CENTER`
`RIGHT`

76.2.2 SubtitleComponentPattern

Transport stream subtitle source.

struct

`language`	**string** Three letter language code.
`component`	**optional component.SubtitleFormat** Subtitle format.
`pid`	**optional int** Subtitle PID.

77 testGenComponent (1.1)

77.1 Overview

New in v1.1:

- Use version 1.1 of testGeneratorProfile that includes Freeze Frame New in v1.0:
- These types were factored out from coderService

77.2 Type Reference

77.2.1 TestGenComponent

Configuration for a test pattern generator profile.

struct

`profile`	**TestGenComponent.profile** struct with id UUID of the test pattern generator profile containing the configuration; and optionally cpy a copy of the content of the profile.
`enable`	**bool** Parameter to force test generation to override any input

77.2.2 TestGenComponent.profile

struct

`id`	**UUID**
`cpy`	**optional testGeneratorProfile.TestGeneratorProfile**

78 testGenerator (1.1)

78.1 Overview

```
# Changelog

## 1.1
- **Changed**
    - Added Freeze Frame as a signal loss option
```

78.2 Type Reference

78.2.1 BitmapBackgroundProperty

Specifies the color property of the background in a bitmap.

enum

BLACK	Make the bitmap background solid black.
TRANSPARENT	Make the bitmap background transparent, such that the bitmap in essence becomes an overlay.

78.2.2 Color

A color defined in terms of RGB.

struct

red	**int** Amount of red. Value 0 to 255.
green	**int** Amount of green. Value 0 to 255.
blue	**int** Amount of blue. Value 0 to 255.

78.2.3 SignalLoss

The types of output that can be produced in the event of signal loss.

enum

COLOR_BAR	Displays color bar image.
BLACK_FRAME	Displays black frame.
FREEZE_FRAME	Displays last frame available.
SYNC_LOSS	Displays nothing.
TEST_GENERATOR	

78.2.4 TextPosition

Specify the vertical placement for the text.

enum

`TOP`	Place it at the top of the image.
`MIDDLE`	Place it in the middle of the image.
`BOTTOM`	Place it at the bottom of the image.

79 testGeneratorProfile (1.1)

79.1 Overview

New in v1.1:

- update to 1.1 version of testGenerator (includes Freeze Frame)

79.2 Command Reference

79.2.1 GetTestGeneratorProfiles

- **message** GetTestGeneratorProfiles.Request
- **message** GetTestGeneratorProfiles.Response
- **message** GetTestGeneratorProfiles.Failure

79.2.2 SetTestGeneratorProfiles

- **message** SetTestGeneratorProfiles.Request
- **message** SetTestGeneratorProfiles.Response
- **message** SetTestGeneratorProfiles.Failure

79.2.3 DeleteTestGeneratorProfiles

- **message** DeleteTestGeneratorProfiles.Request
- **message** DeleteTestGeneratorProfiles.Response
- **message** DeleteTestGeneratorProfiles.Failure

79.3 Type Reference

79.3.1 DeleteTestGeneratorProfiles.Failure

empty **struct**

79.3.2 DeleteTestGeneratorProfiles.Request

struct

`ids`	**list** of **UUID**

79.3.3 DeleteTestGeneratorProfiles.Response

empty **struct**

79.3.4 GetTestGeneratorProfiles.Failure

empty **struct**

79.3.5 GetTestGeneratorProfiles.Request

empty **struct**

79.3.6 GetTestGeneratorProfiles.Response

struct

data	**map** from **UUID** to **TestGeneratorProfile**
info	**GetTestGeneratorProfiles.Response.info**

79.3.7 GetTestGeneratorProfiles.Response.info

struct

totalEntries	**int**

79.3.8 MovingBox

Parameters for specifying properties of test generation moving box

struct

enable	**bool** Enable moving box.
color	**testGenerator.Color** Enum describing color of moving box.

79.3.9 SetTestGeneratorProfiles.Failure

empty **struct**

79.3.10 SetTestGeneratorProfiles.Request

struct

data	**map** from **UUID** to **TestGeneratorProfile**

79.3.11 SetTestGeneratorProfiles.Response

empty **struct**

79.3.12 TestGenVariant

Variant for the different types of test generation.

variant

lipSync	**TestGenVariant.lipSync** variant specifies that the profile activates lipSync.
testGenerator	**TestGenerator** variant contains test generation parameters.

79.3.13 TestGenVariant.lipSync

empty **struct**

79.3.14 TestGenerator

Properties of test generation

struct

`testPattern`	**testGenerator.SignalLoss** Enum describing type of pattern to be generated.
`testTone`	**bool** Audio tone generation for channels.
`text`	**Text** Parameters regarding adding text to the test generation.
`box`	**MovingBox** Parameters regarding unsg a moving box for the test generation.

79.3.15 TestGeneratorProfile

TestGeneratorProfileStruct

79.3.16 TestGeneratorProfileStruct

top level profile structure

struct

`label`	**string** Name of the profile
`enable`	**bool** Parameter to force test generation to override any input.
`testGenVariant`	**TestGenVariant** either lipSync or test generator.

79.3.17 Text

Parameters for specifying properties of test generation text

struct

`enable`	**bool** Enable text.
`position`	**testGenerator.TextPosition** Specifies screen location for the text.
`backgroundType`	**testGenerator.BitmapBackgroundProperty** Specifies the color property of the background.

80 timeMode (1.0)

80.1 Type Reference

80.1.1 TimeMode

Specifies the source of timing information.

enum

NTP
PTP_IF_LOCKED
PTP_FORCED

81 timecodeConfig (1.1)

81.1 Type Reference

81.1.1 TimecodeSource

The clock source to be used by the encoder.

enum

OFF
SMPTE_12M
PTP
NTP
PTP_OR_NTP
PTP_UTC_ADJUSTED

82 timecodeStatus (1.0)

82.1 Type Reference

82.1.1 Timecode

- Timecode information from SMPTE 12M, PTP or NTP source.

struct

`timecode`	**UnixTime** Current time in seconds in unix format
`isPresent`	**bool** Availability of timecode information

82.1.2 UnixTime

bigint

83 transceiverStatus (1.1)

83.1 Type Reference

83.1.1 CableLengthIndicator

Information about the cable length indicator measurement. The measurement is given in terms of the Belden 1694A cable.

struct

`length`	**int** An indication of the cable length carrying the SDI input signal [m].
`lengthPrecision`	**int** The precision of the cable length measurement [m]. The actual length of the cable is likely to be in the interval [length - precision, length + precision].
`maxLength`	**int** The maximum supported cable length [m].
`threshold`	**int** The threshold below which measurements of cable length are very imprecise [m].

83.1.2 DataRate

The various input data rates the transceiver can lock onto.

enum

`DR_11_88_GBPS`
`DR_5_94_GBPS`
`DR_2_97_GBPS`
`DR_1_485_GBPS`
`DR_270_MBPS`
`NO_LOCK`

83.1.3 SignalProperties

struct

`dataRate`	**DataRate** An indication of the data rate of the SDI signal that the transceiver has locked.
`cableLength`	**optional CableLengthIndicator** An indication of the cable length carrying the SDI input signal [m]. N/A if dataRate is NO_LOCK.
`horizontalEyeOpening`	**optional float** Horizontal eye opening of the equalized SDI signal [UI]. Not available if dataRate is NO_LOCK. Not available for SD.

83.1.4 TransceiverStatus

Status about the quality of an SDI signal. These measurements are taken by the transceiver which is connected to the physical connector carrying the SDI signal.

struct

`carrierDetect`	**bool** True if the transceiver has detected that there is an input signal from an SDI cable.
`signalProperties`	**optional SignalProperties** Properties of the input signal that are available if a carrier is detected.

84 tsDestination (1.0)

84.1 Overview

New in v1.0:

- These types were factored out from coderService
- Was TsOut before. Got renamed to TsDestination

84.2 Type Reference

84.2.1 TsDestination

struct

serviceName	**string**
serviceId	**int**
serviceProvider	**string**
pcrInterval	**float**
pcrPid	**int**
pmtPid	**int**

85 tsSource (1.5)

85.1 Overview

Changelog

1.5

- **Changed**
  - `subtitling` from 1.1 to 1.2

1.4

- **Added**
  - Amend `switchbackDelay` documentation in `InputRedundancy`; milliseconds, not seconds.
  - `switchDelay` to `InputRedundancy`

1.3

- **Added**
  - `FLOATING` to `SwitchingMode`.

1.2

- **Changed**
  - `subtitling` from 1.0 to 1.1

85.2 Type Reference

85.2.1 HotStandby

HotStandby configuration parameters.

struct

`sourceProperty`	**SourceProperty** specifies whether the stream sources are identical or not.

85.2.2 IdenticalSources

IdenticalSources configuration parameters.

struct

`maxOffset`	**int** The maximum tolerated delay between identical sources in ms.

85.2.3 InputRedundancy

InputRedundancy configuration parameters.

struct

	SwitchingMode

mode	the input redundancy switching mode.
switchbackDelay	**int** a delay in milliseconds to wait before reverting to main. N/A for switching mode floating.
switchDelay	**int** A delay in milliseconds to wait before switching to non-active source. N/A for other switching modes than floating.
manualOverride	**optional streamSource.StreamSource** a chosen preferred source will be forced even if it is not present.
hotStandby	**HotStandby** the hot standby configuration parameters.

85.2.4 SourceProperty

SourceProperty switching mode.

variant

nonIdenticalSources	**SourceProperty.nonIdenticalSources** the stream sources are not identical.
identicalSources	**IdenticalSources** identical stream sources configuration parameters.

85.2.5 SourceProperty.nonIdenticalSources

empty **struct**

85.2.6 SwitchingMode

InputRedundancy switching mode.

enum

REVERTING	allows stream source switching whenever the main is back.
FLOATING	allows switching to another input and use it as long as it's error free.
OFF	Input Redundancy is turned off, no switching will takeplace.

85.2.7 TsSource

TS video source parameters.

struct

sources	**map** from **streamSource.StreamSource** to **UUID** a map of the current stream source and their uuids. a MAIN stream source must always be present
inputRedundancy	**optional InputRedundancy** the input redundancy configuration parameters.
	list of **subtitling.SubtitleComponentPattern**

`subtitlePriorityList`	List of SubtitleComponentPattern describing which subtitle components in the transport stream may be used to process subtitling.

86 vancProfile (1.7)

86.1 Overview

Changelog:

1.7

- **Changed**
  - Moved `Dpi` and `DpiSource` to its own file.

1.6

- **Added**
  - New `horizontalAlignment` member to `DvbSubtitle`.
- **Changed**
  - `subtitling` from 1.1 to 1.2.

1.5

- **Added**
  - enum `DpiSource`
  - member 'source' to Dpi

1.4

- **Added**
  - type `DvbSubtitle`
  - member `dvbSubtitle` to VancParams
  - value `DVBSUBTITLE` to VancParamsEnum

86.2 Command Reference

86.2.1 GetVancProfiles

- **message GetVancProfiles.Request**
- **message GetVancProfiles.Response**
- **message GetVancProfiles.Failure**

86.2.2 SetVancProfiles

- **message SetVancProfiles.Request**
- **message SetVancProfiles.Response**
- **message SetVancProfiles.Failure**

86.2.3 DeleteVancProfiles

- **message DeleteVancProfiles.Request**
- **message DeleteVancProfiles.Response**
- **message DeleteVancProfiles.Failure**

86.3 Type Reference

86.3.1 Anc

Anc.

struct

`did`	**int** vanc did.
`sdid`	**int** vanc sdid.
`line`	**optional int** line selection.

86.3.2 ClosedCaption

empty **struct**

86.3.3 DataSelection

Specifies which ancillary data packets are included in the SMPTE 2038 PID. An empty variant is treated as if auto was specified.

variant

`manual`	**DataSelection.manual** Only ancillary data that matches the provided list of DID/SDID pairs is used (up to 4 pairs supported).
`auto`	**DataSelection.auto** Use any incoming data.

86.3.4 DataSelection.auto

empty **struct**

86.3.5 DataSelection.manual

struct

`input`	**list** of **Anc**

86.3.6 DeleteVancProfiles.Failure

empty **struct**

86.3.7 DeleteVancProfiles.Request

struct

`ids`	**list** of **UUID**

86.3.8 DeleteVancProfiles.Response

empty **struct**

86.3.9 DvbSubtitle

DVB subtitle

struct

`teletext`	**Teletext** EBU teletext to be converted to DVB subtitles
`horizontalAlignment`	**subtitling.HorizontalAlignment**

86.3.10 EN301775

struct

`teletext`	**optional Teletext**
`vps`	**optional Vps**
`wss`	**optional Wss**
`cc`	**optional ClosedCaption**
`dataId`	**optional int**

86.3.11 GetVancProfiles.Failure

empty **struct**

86.3.12 GetVancProfiles.Request

empty **struct**

86.3.13 GetVancProfiles.Response

struct

`data`	**map** from **UUID** to **VancProfile**
`info`	**GetVancProfiles.Response.info**

86.3.14 GetVancProfiles.Response.info

struct

`totalEntries`	**int**

86.3.15 SetVancProfiles.Failure

empty **struct**

86.3.16 SetVancProfiles.Request

struct

data	**map** from **UUID** to **VancProfile**

86.3.17 SetVancProfiles.Response

empty **struct**

86.3.18 Smpte2038

Smpte 2038.

struct

dataSelection	**DataSelection** modes of ANC data selection.
delay	**int** adds offset of x ms to PTS.
maxBitrate	**optional int** sets a max bitRate.

86.3.19 Teletext

struct

input	**VancStandard**
teletext	**list** of **TeletextPsi**
lineFilter	**list** of **int**

86.3.20 TeletextPsi

struct

page	**int**
language	**string**
teletextType	**TeletextType**

86.3.21 TeletextType

enum

START
SUBTITLE
HEARING_IMPAIRED

86.3.22 VancParams

Configuration of the type of ancillary data in transport stream

variant

dpi	**dpi.Dpi** Digital program insertion messages according to scte-35
en301775	**EN301775** VBI in DVB according to ETSI EN 301 775
smpte2038	**Smpte2038** Ancillary data packets in a transport stream according to SMPTE ST 2038
dvbSubtitle	**DvbSubtitle** DVB subtitles according to ETSI EN 300 743

86.3.23 VancParamsEnum

enum

DPI
EN301775
SMPTE2038
DVBSUBTITLE

86.3.24 VancProfile

VancProfileStruct

86.3.25 VancProfileStruct

struct

label	**string**
vancParams	**VancParams**

86.3.26 VancStandard

enum

OP_47
SMPTE_2031
VBI

86.3.27 Vps

empty **struct**

86.3.28 Wss

empty **struct**

87 vancStatus (1.14)

87.1 Overview

Changelog:

1.14

- **Changed**
  - `dataIdentifiers` in `Smpte2038Status` from `list(DataIdentifier)` to `optional(list(DataIdentifier))`

1.13

- **Changed**
  - `vancProfile` from 1.6 to 1.7
  - Moved `DpiCommand`, `DpiInput`, and `DpiStatus` to its own file.

1.12

- **Changed**
  - `vancProfile` from 1.5 to 1.6

1.11

- **Changed**
  - `sourceIdentifier` member in `dpiInput`.

1.10

- **Changed**
  - `dpiIn` member in `VancStatus` from `DpiInput` to `list(DpiInput)`.
  - Added `timeSinceLastReceived` to `DpiInput`.
  - Added `lastGeneratedCommand` to `DpiInput`.
  - Added `DpiCommand` type

1.9

- **Changed**
  - added value `timeSinceLastReceived` to `DpiStatus`
  - added value `timeSinceLastGenerated` to `DpiStatus`
  - added value `lastCommandType` to `DpiStatus`

1.8

- **Changed**
  - `vancProfile` from 1.4 to 1.5

1.7

- **Changed**
  - added value `dvbSubtitle` to `VancComponentStatus`
  - added value `DVBSUBTITLESTATUS` to `VancComponentStatusEnum`
  - changed function `VancComponentStatus2Enum` to accomodate `DVBSUBTITLESTATUS`

- **Added**
  - struct `DvbSubtitleStatus`

87.2 Type Reference

87.2.1 AspectRatioInput

struct

`standard`	**aspectRatio.AspectRatioStandard**
`format`	**afd.AfdCode**
`is16x9`	**bool**

87.2.2 DataIdentifier

The Data Identifier (DID) and Secondary Data Identifier (SDID) words which together indicate a type of ancillary data.

struct

`did`	**int** The Data Identifier word.
`sdid`	**int** The Secondary Data Identifier word.
`lineNum`	**int** The video line number of the DID/SDID.

87.2.3 DvbSubtitleStatus

DVB subtitle status

struct

`enabled`	**bool** Set to false if the feature is prevented from running (by licences, invalid config etc)
`timeSinceLastReceived`	**int** Seconds since last valid EBU subtitle input
`timeSinceLastGenerated`	**int** Seconds since last generated DVB subtitle

87.2.4 En301775Status

struct

`ttxtStatus`	**optional TeletextStatus**

87.2.5 Smpte2038Status

Smpte 2038 Status.

struct

enabled	**bool**
running	**bool**
dataIdentifiers	**optional list** of **DataIdentifier**

87.2.6 TeletextInput

struct

input	**vancProfile.VancStandard**
linesIn	**list** of **int**
invalidPackets	**int**
received	**bool**

87.2.7 TeletextStatus

struct

standard	**vancProfile.VancStandard**
enabled	**bool**
running	**bool**
linesOut	**list** of **int**

87.2.8 VancComponentStatus

variant

dpi	**dpiStatus.DpiStatus**
en301775	**En301775Status**
smpte2038	**Smpte2038Status**
dvbSubtitle	**DvbSubtitleStatus**

87.2.9 VancComponentStatusEnum

enum

DPISTATUS
En301775STATUS
SMPTE2038STATUS
DVBSUBTITLESTATUS

87.2.10 VancStatus

struct

ttxtIn	**list** of **TeletextInput**
dpiIn	**list** of **dpiStatus.DpiInput**
arIn	**list** of **AspectRatioInput**
outputs	**map** from **int** to **VancComponentStatus**

`sources`	**map** from **int** to **set** of **int**
`dataIdentifiers`	**list** of **DataIdentifier**

88 videoProfile (2.14)

88.1 Overview

Changelog:

2.14

- **Added**
  - JPEG_XS to video codecs
  - `codedVideo` version 1.0

2.13

- **Changed**
  - GopParameters.hierarchical: false is no longer allowed, must be true

2.12

- **Added**
  - jpeg2000 to CodecParameters
  - Jpeg2000Parameters struct (with 2013 and 2018 mode)

2.11

- **Changed**
  - `rawVideo` from 1.4 to 1.5

2.10

- **Added**
  - extendedSar bool to videoProfileStruct1

2.9

- **Changed**
  - Bitrate changed from int to variant of int (cbr) and struct containing two ints (cvbr)

88.2 Command Reference

88.2.1 GetVideoProfiles

- **message GetVideoProfiles.Request**
- **message GetVideoProfiles.Response**
- **message GetVideoProfiles.Failure**

88.2.2 SetVideoProfiles

- **message SetVideoProfiles.Request**
- **message SetVideoProfiles.Response**
- **message SetVideoProfiles.Failure**

88.2.3 DeleteVideoProfiles

- **message DeleteVideoProfiles.Request**

- message DeleteVideoProfiles.Response
- message DeleteVideoProfiles.Failure

88.3 Type Reference

88.3.1 Afd

enum

Off
Passthrough

88.3.2 AspectRatio

enum

AR_TRANSPARENT
AR_16x9
AR_14x9
AR_4x3
AR_AFD
AR_WSS
AR_WSS18
AR_VideoIndex_11_324
AR_VideoIndex_8_321

88.3.3 AvcLevel

enum

AUTO
LEVEL_3
LEVEL_3p1
LEVEL_3p2
LEVEL_4
LEVEL_4p1
LEVEL_4p2
LEVEL_5
LEVEL_5p1

88.3.4 AvcParameters

struct

profile	AvcProfile
level	AvcLevel
codingMode	CodingMode
idrFrequency	int
cabac	bool

88.3.5 AvcProfile

enum

MAIN	
HIGH	
HIGH10	
HIGH42210	
CONSTRAINED_BASELINE	

88.3.6 Bitrate

variant

cbr	**int**
cvbr	**Cvbr**

88.3.7 CodecParameters

variant

hevc	**HevcParameters**
avc	**AvcParameters**
mpeg2	**Mpeg2Parameters**
jpeg2000	**Jpeg2000Parameters**
jpegXs	**JpegXsParameters**

88.3.8 CodingMode

enum

AUTO	
FRAME	
FIELD	
PAFF	
MBAFF	

88.3.9 Cvbr

struct

maxBitrate	**int**
averageBitrate	**int**

88.3.10 DeleteVideoProfiles.Failure

empty **struct**

88.3.11 DeleteVideoProfiles.Request

struct

ids	**list** of **UUID**

88.3.12 DeleteVideoProfiles.Response

empty **struct**

88.3.13 GetVideoProfiles.Failure

empty **struct**

88.3.14 GetVideoProfiles.Request

empty **struct**

88.3.15 GetVideoProfiles.Response

struct

data	**map** from **UUID** to **VideoProfile**
info	**GetVideoProfiles.Response.info**

88.3.16 GetVideoProfiles.Response.info

struct

totalEntries	**int**

88.3.17 GopMode

enum

Static
Dynamic

88.3.18 GopParameters

struct

size	**int**
gopMode	**GopMode**
structure	**GopStructure**
maxBframes	**int**
ldb	**bool**
hierarchical	**bool**

88.3.19 GopStructure

enum

I_ONLY
IP
IPB
GDR

88.3.20 HevcLevel

enum

AUTO
LEVEL_3
LEVEL_3p1
LEVEL_4
LEVEL_4p1
LEVEL_5
LEVEL_5p1

88.3.21 HevcParameters

struct

profile	HevcProfile
level	HevcLevel
tier	HevcTier
idrFrequency	int

88.3.22 HevcProfile

enum

MAIN
MAIN10
MAIN42210

88.3.23 HevcTier

enum

Main
High

88.3.24 Jpeg2000Parameters

struct

profile	Jpeg2000Profile

88.3.25 Jpeg2000Profile

enum

MODE_2013	
MODE_2018	

88.3.26 JpegXsParameters

struct

optimizationTarget	codedVideo.OptimalizationTarget

88.3.27 Latency

enum

NORMAL	
LOW	
ULL	

88.3.28 Mpeg2Level

enum

MAIN	
HIGH	

88.3.29 Mpeg2Parameters

struct

profile	Mpeg2Profile
level	Mpeg2Level
codingMode	CodingMode
openGop	bool

88.3.30 Mpeg2Profile

enum

MAIN	
PROFILE_422P	

88.3.31 ProfileType

enum

ENCODER	
DECODER	

TRANSCODER

88.3.32 ScanningMode

enum

INTERLACED
PROGRESSIVE

88.3.33 SetVideoProfiles.Failure

empty **struct**

88.3.34 SetVideoProfiles.Request

struct

data	**map** from **UUID** to **VideoProfile**

88.3.35 SetVideoProfiles.Response

empty **struct**

88.3.36 VideoCodec

enum

HEVC
AVC
MPEG2
JPEG2000
JPEG_XS

88.3.37 VideoProfile

VideoProfileStruct

88.3.38 VideoProfileStruct

struct

profileType	**ProfileType**
label	**string**
bitDepth	**int**
latency	**Latency**
resolution	**VideoResolution**
codec	**VideoCodec**
bitrate	**Bitrate**

cparams	CodecParameters
ar	AspectRatio
fallbackAr	AspectRatio
gop	GopParameters
afd	Afd
wssBlanking	bool
extendedSar	bool

88.3.39 VideoResolution

struct

horizontal	rawVideo.HorizontalResolution
vertical	rawVideo.VerticalResolution
chromaSampling	rawVideo.ChromaSampling
scan	ScanningMode
fps	rawVideo.FrameRate
followInput	bool

89 videoStatus (2.18)

89.1 Overview

Video status types and objects. This IDCL is part of the IDCL sent to the GUI

Changelog:

2.18

- **Changed**
  - 'videoProfile from 2.13 to 2.14

2.17

- **Changed**
  - `videoUsabilityInfo` from 1.1 to 1.2
  - 'videoProfile from 2.12 to 2.13

2.16

- **Changed**
  - 'videoProfile from 2.11 to 2.12

2.15

- **Changed**
  - 'videoProfile from 2.10 to 2.11

2.14

- **Changed**
  - 'videoProfile from 2.9 to 2.10
  - `VuiColorimetry` added to `Es`

2.13

- **Changed**
  - `colorProfile` to `videoUsabilityInfo`

2.12

- **Changed**
  - `videoProfile` from 2.8 to 2.9
  - `ccPresent` added to `Es`

2.11

- **Changed**
  - Ts.bitRate changed from int to bigint to allow for high bitrates, which will be necessary for UHD

89.2 Type Reference

89.2.1 Es

Status for coded video.

struct

valid	**bool**
hSize	**int**
vSize	**int**
bitRate	**int**
cpb	**optional int**
bitDepth	**int**
chroma	**int**
codec	**videoProfile.VideoCodec**
aspectRatio	**optional videoProfile.AspectRatio**
framerate	**optional rawVideo.FrameRate**
scan	**optional videoProfile.ScanningMode**
colorDescPresent	**bool**
colorPrimaries	**videoUsabilityInfo.ColorSpace**
transferChar	**videoUsabilityInfo.TransferFunc**
matrixCoeffs	**videoUsabilityInfo.ColorSpace**
idrOnScte35	**optional int**
ccPresent	**bool**
vuiColorimetry	**optional VuiColorimetry**

89.2.2 InputStatus

struct

id	**InstanceId**
input	**VideoFormat**

89.2.3 InstanceId

int

89.2.4 OutputStatus

struct

id	**InstanceId**
output	**VideoFormat**

89.2.5 Raw

struct

valid	**bool**
numFrames	**int**
videoMode	**optional VideoModes**
framerate	**optional rawVideo.FrameRate**

chroma	optional int
bitDepth	optional int
vitcPresent	bool
vitcLineNr	optional int
ccPresent	bool
timecode	optional timecodeStatus.Timecode

89.2.6 Ts

struct

bitRate	bigint
ccError	int
pcrJitterError	optional bool
pcrDisc	optional bool
pcrIntervalErr	optional bool
es	Es
pid	optional int

89.2.7 VideoFormat

variant

raw	Raw
es	Es
ts	Ts

89.2.8 VideoModes

enum

V_UNKNOWN
V_480i
V_576i
V_720p
V_1080i
V_1080p
V_2160p

89.2.9 VideoStatus

struct

input	InputStatus
output	OutputStatus

89.2.10 VuiColorimetry

struct

colorPrimaries	int
transferChar	int
matrixCoeffs	int

90 videoUsabilityInfo (1.2)

90.1 Overview

```
# Changelog
## 1.2
- **Added**
  - BT_2100 to `ColorSpace` enum
  - BT_2100 support in `getColourPrimaries()` function
  - BT_2100 support in `getMatrixCoefficients()` function

- **Changed**
  - Corrected typos: BT_2010_SDR to BT_2020_SDR, BT_2011_PQ to BT_2100_PQ, BT_2011_HLG to BT_210
  - Fixed same typos in `getTransferCharacteristics()` function

## 1.1
- **Changed**
  Added BT_601 to `ColorSpace`
  Added BT_601 to `TransferFunc`
```

90.2 Type Reference

90.2.1 ColorSpace

ColorSpace

ColorSpace is a set of colour coordinates for what each of red, green, blue and white “mean”

enum

UNDEFINED
BT_709
BT_601
BT_2020
BT_2100

90.2.2 TransferFunc

TransferFunc

Transfer function to convert video signals to light

enum

UNDEFINED
BT_709
BT_601
BT_2020_SDR
BT_2100_PQ
BT_2100_HLG

90.2.3 VuiParameters

VuiParameters

Video Usability Info

struct

`colorSpace`	**ColorSpace** A set of colour coordinates for what each of red, green, blue and white “mean”.
`transfer`	**TransferFunc** Transfer function to convert video signals to light
`matrix`	**ColorSpace** Specifies the matrix coefficients used to derive the luma and chroma from the RGB primaries.

91 vpidInfo (1.0)

91.1 Type Reference

91.1.1 ColorInfo

struct

colorimetry	**Colorimetry**
transfer	**TransferCharacteristics**

91.1.2 Colorimetry

enum

REC_709
COLOR_VANC
UHD_TV
UNKNOWN

91.1.3 TransferCharacteristics

enum

SDR
HLG
PQ
UNSPECIFIED