VideoQ Productivity Tools

A suite of software modules
for advanced video processing workflow

Training Presentation

December 2024

http://www.videoq.com/vqpt.html

www.videoq.com

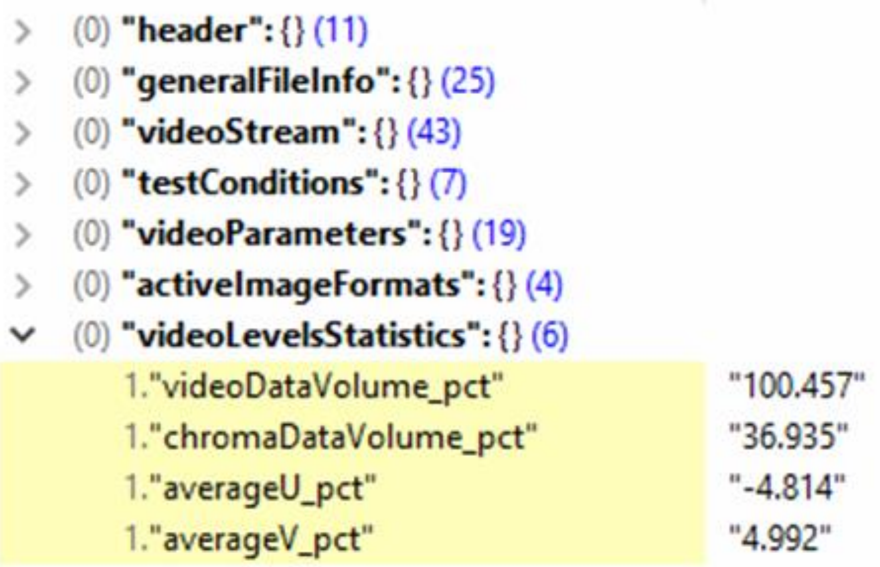
Table of Contents

Background Info

1. VideoQ Productivity Tools are designed “**by engineers, for engineers**”
2. An ever higher number of channels/programs/titles
3. And a permanently growing number of formats, frames sizes, bitrates, etc.
4. Human resources required for input QC and output QC has escalated
5. A new approach and **new tools** are needed *as demanded by our customers*
6. Hence VideoQ has changed the focus from our traditional T&M tools to

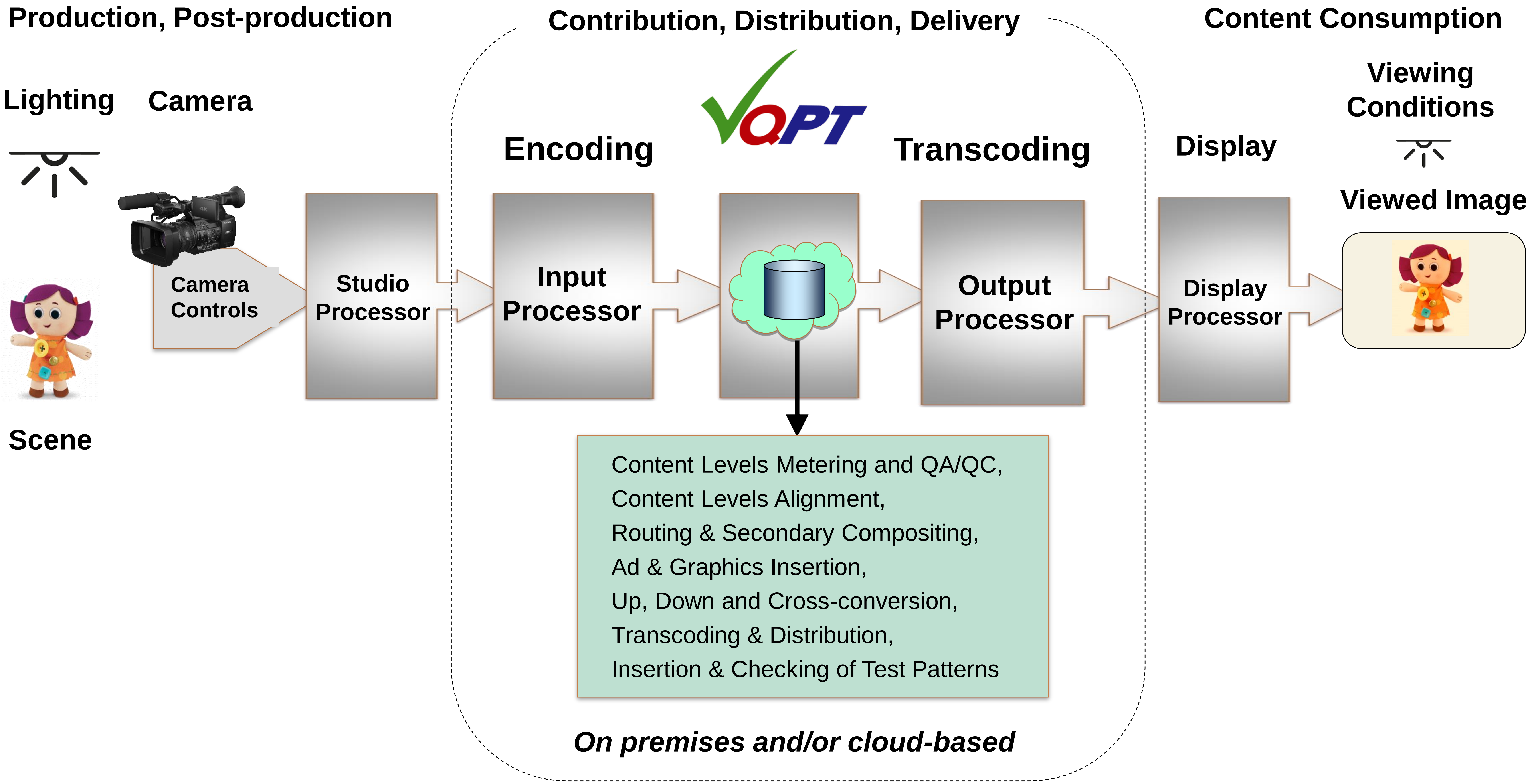
   Automated Productivity Tools

- Automation is essential, but …
- Human intervention cannot be excluded
- Thus, our slogan is: “**Robot-assisted human decisions**”



```
> (0) "header": {} (11)
> (0) "generalFileInfo": {} (25)
> (0) "videoStream": {} (43)
> (0) "testConditions": {} (7)
> (0) "videoParameters": {} (19)
> (0) "activeImageFormats": {} (4)
v (0) "videoLevelsStatistics": {} (6)
      1."videoDataVolume_pct"      "100.457"
      1."chromaDataVolume_pct"     "36.935"
      1."averageU_pct"             "-4.814"
      1."averageV_pct"             "4.992"
```

The Big Picture – Overall System View



Production, Post-production

Lighting

Camera

Scene

Camera Controls

Studio Processor

Contribution, Distribution, Delivery

Encoding

VQPT

Transcoding

Input Processor

Output Processor

Content Levels Metering and QA/QC,
Content Levels Alignment,
Routing & Secondary Compositing,
Ad & Graphics Insertion,
Up, Down and Cross-conversion,
Transcoding & Distribution,
Insertion & Checking of Test Patterns

On premises and/or cloud-based

Content Consumption

Display

Viewing Conditions

Viewed Image

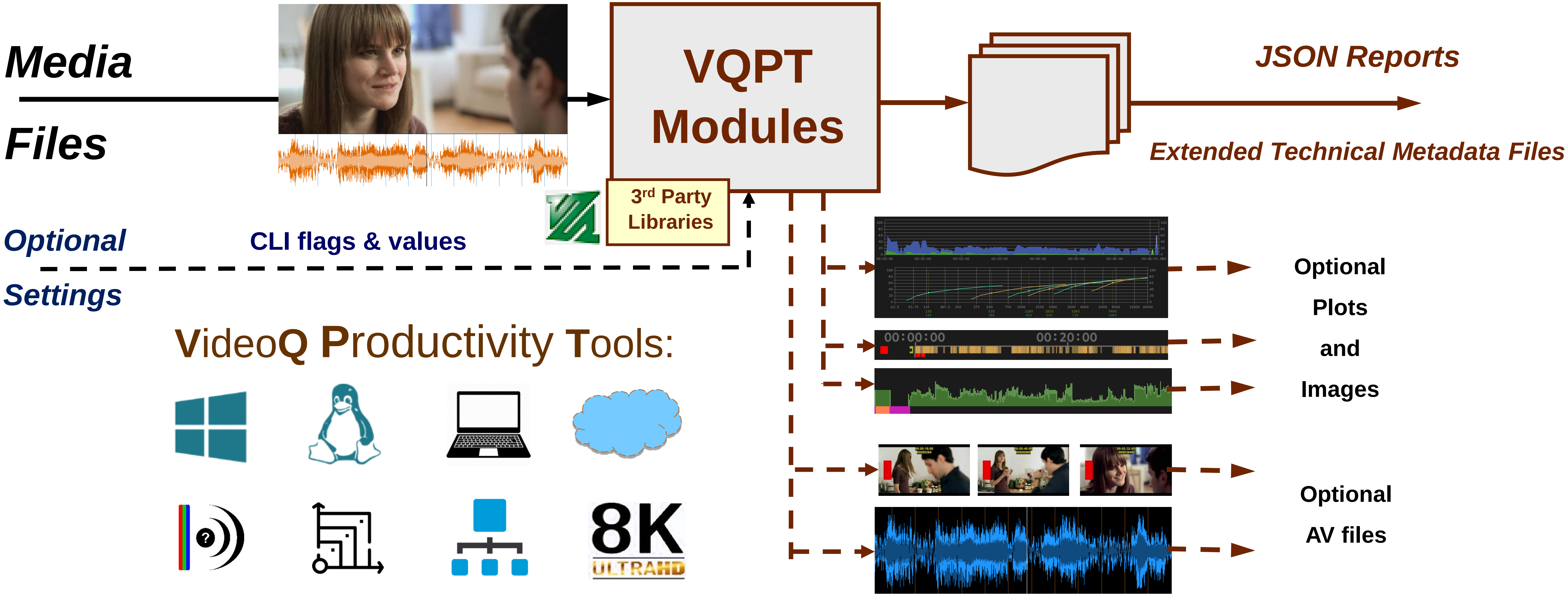
Display Processor

Top Level VQPT Automated Workflow Diagram

VQPT is a suite of portable Windows/Linux CLI programs for on premises and cloud computing.

It can be used for production, post-production and distribution applications.

The program modules can be purchased and used separately or grouped for typical applications.



Principles of Application

With the massive increase of volumes of hardware items and video related software, the strict rules established for the broadcast TV are not always recognized.

The solution is in establishing easy-to-use and straightforward **rules** and matching **tools**.

1. VideoQ **VQPT** is the cloud-based **QA/QC** and **transcoding workflow optimization tool**, that answers this challenge
2. Periodic testing of the **workflow health** should be combined with permanent checking of input and output **AV content parameters**
3. The most efficient methodology of such QA/QC operations is the creation of machine readable **reports** built by automated program modules and the subsequent review of these reports by a **human operator**
4. Storage of such reports (aka **E**xtended **T**echnical **M**etadata) in the **centralized database**, that allows the **remote access** by authorized users, is vitally important for the efficient **management** of the whole content delivery process

VQPT Packages

VQPT program modules can be used separately or grouped for the following typical applications:

Pack 1. Target Application: **Workflow Health Tests**

VQMINF – Media File Info Report Generator
VQCBA – VideoQ Color Bars Analyzer, companion program for **VQCB** test patterns
VQCSA – Compression Stress Analyzer, companion program for **VQCST** test patterns
VQMA – Video Analyzer for objective video processing chain integrity tests: https://videoq.com/vqma.html

Pack 2. Target Application**: AV Files Conversion, Encoding and Transcoding**

VQMINF – Media File Info Report Generator
VQBIF – BIF (Base Index Frames) Files Verifier
VQBLA – Bitrate Ladder Analyzer
VQC – HDR-SDR Files Converter: https://videoq.com/vqc.html
VQCSA – Compression Stress Analyzer, companion program for **VQCST** test patterns
VQLPN – Audio Loudness Profiler and Normalizer: https://videoq.com/vqlpn.html
VQTSF – Transcoding Segments Finder

Pack 3. Target Application**: AV Content Analysis**

VQMINF – Media File Info Report Generator
VQCFA – Captions Files Analyzer
VQFP – Video Frames Profiler
VQLPC – Loudness Profiles Correlator, companion program for **VQLPN** module
VQLPN – Audio Loudness Profiler and Normalizer: https://videoq.com/vqlpn.html
VQPLA – Picture Levels Analyzer

Pack 1: VQPT Modules for Workflow Tests

- VQMINF – Media File Info Report Generator

  VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in UTF8 JSON format.
  It can read HDR-PQ, HDR-HLG and SDR video as well as LOG video metadata.
  VQMINF also calculates unique vqminfEssenceID string and input file MD5 hash string.

- VQCBA – VideoQ Color Bars Analyzer, companion program for **VQCB** test patterns.

  http://www.videoq.com/vqcba.html http://www.videoq.com/vqcb.html

  This module reads VQCB test pattern file at particular workflow test point, then checks video and audio levels, detects color space distortions, dynamic range conversion footprints and measures AV Sync errors

- VQCSA – Compression Stress Analyzer, companion program for **VQCST** test patterns.

  http://www.videoq.com/Downloads/VideoQ_StressTracker_Training_Presentation_PPT.pdf

  This module reads VQCST test pattern file at particular workflow test point, measures VMAF scores for particular Compression Stress parameters and codecs settings, then builds Compression Stress Response Profiles

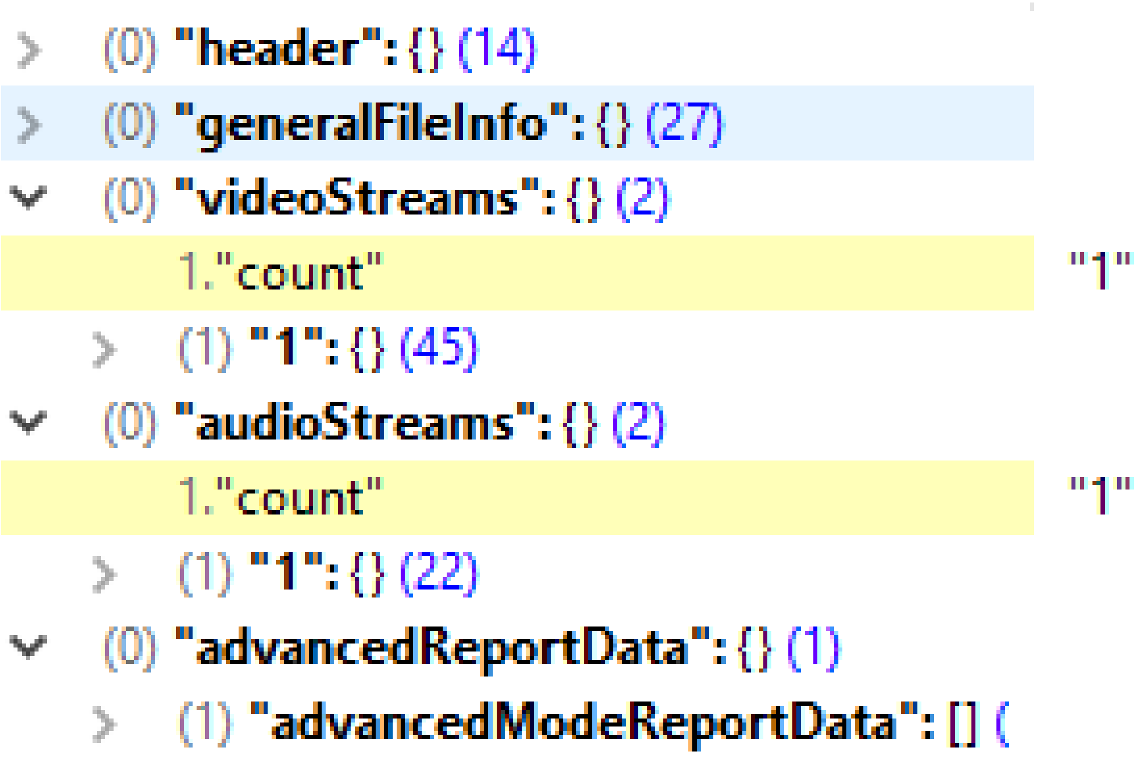
- VQMA – Video Analyzer for objective video processing chain integrity tests.

  http://www.videoq.com/vqma.html

  This program reads VQMA test pattern file at particular workflow test point, then checks multiple video parameters and checks them vs. pre-configured tolerance values

VQMINF – Media File Info Report Generator

- VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in **UTF8 JSON** format
- It can read **HDR-PQ**, **HDR-HLG** and **SDR** video as well as **LOG** video metadata
- VQMINF uses standard SHA3 and MD5 libraries to calculate unique **vqminfEssenceID** string and input file **MD5 hash** string
- Report file contains the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **textStreams**: technical parameters of text stream(s)
  - **imageStreams**: technical parameters of image stream(s)
  - **advancedReportData**: (*all raw media info strings, useful for manual analysis of complicated cases*)

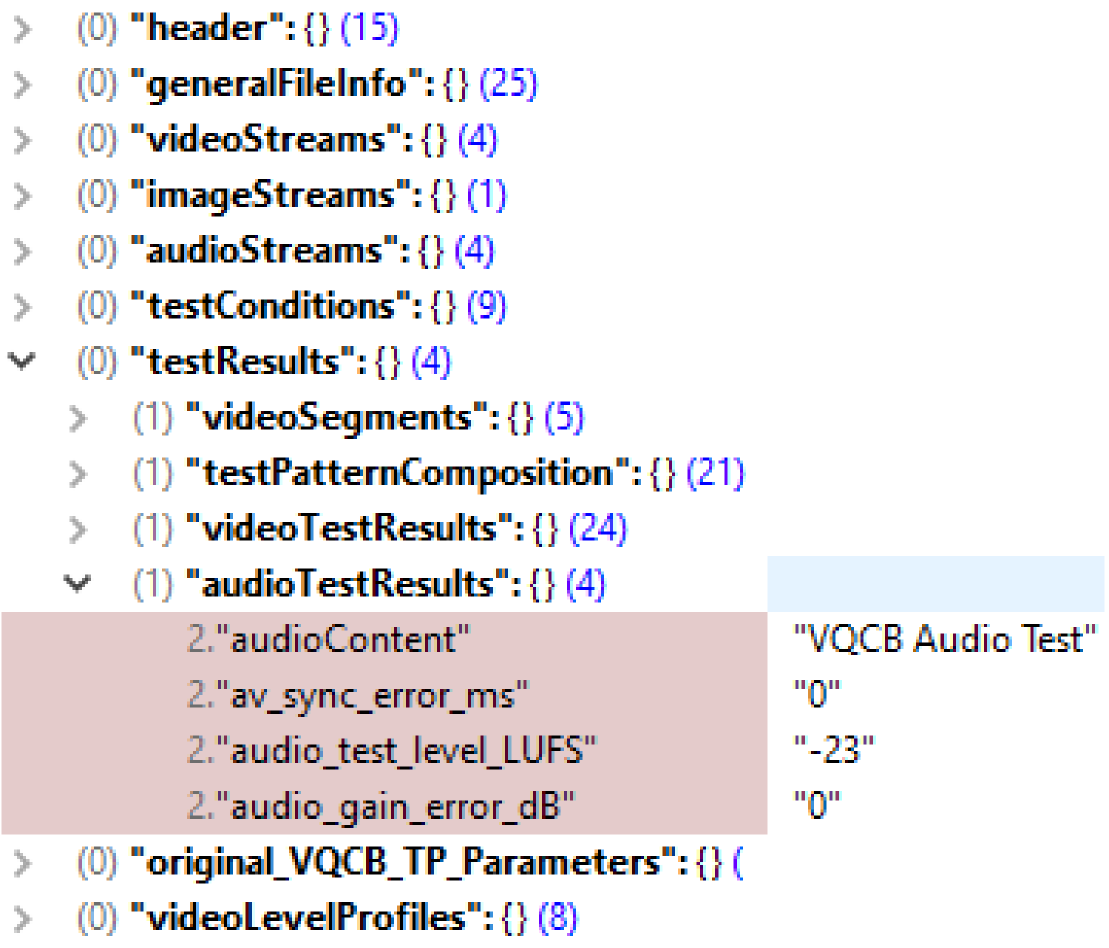


```
> (0) "header": {} (14)
> (0) "generalFileInfo": {} (27)
v (0) "videoStreams": {} (2)
      1."count"                                   "1"
    > (1) "1": {} (45)
v (0) "audioStreams": {} (2)
      1."count"                                   "1"
    > (1) "1": {} (22)
v (0) "advancedReportData": {} (1)
    > (1) "advancedModeReportData": [] (
```

CLI interface: **vqminf** [-noard] [-nohash] -i inFilePath [-o or -o outFilePath]

VQCBA – VideoQ Color Bars Analyzer

- VQCBA reads media file, containing at least one video stream (.AVI, .MOV, .MP4, etc.)
- It measures video frames parameters and creates machine-readable Report in JSON format showing the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **imageStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **testConditions**: user-selected and auto-configured test session parameters
  - **testResults**: several sub-sections containing all measured parameters
  - **original_VQCB_TP_Parameters**: decoded QR code string data
  - **videoLevelsProfile**: pixel-by-pixel YUV and RGB video data values (*for advanced analysis*)



```
> (0) "header": {} (15)
> (0) "generalFileInfo": {} (25)
> (0) "videoStreams": {} (4)
> (0) "imageStreams": {} (1)
> (0) "audioStreams": {} (4)
> (0) "testConditions": {} (9)
v (0) "testResults": {} (4)
    > (1) "videoSegments": {} (5)
    > (1) "testPatternComposition": {} (21)
    > (1) "videoTestResults": {} (24)
    v (1) "audioTestResults": {} (4)
            2."audioContent"             "VQCB Audio Test"
            2."av_sync_error_ms"         "0"
            2."audio_test_level_LUFS"    "-23"
            2."audio_gain_error_dB"      "0"
> (0) "original_VQCB_TP_Parameters": {} (
> (0) "videoLevelProfiles": {} (8)
```

VQCBA can analyze VideoQ VQCB test pattern, as well as other common types of Color Bars test pattern.

CLI interface: **vqcba** [-tm | -cd] [-short] [-tfe] [-sfe] -i inFilePath [-o] or [-o outFilePath]

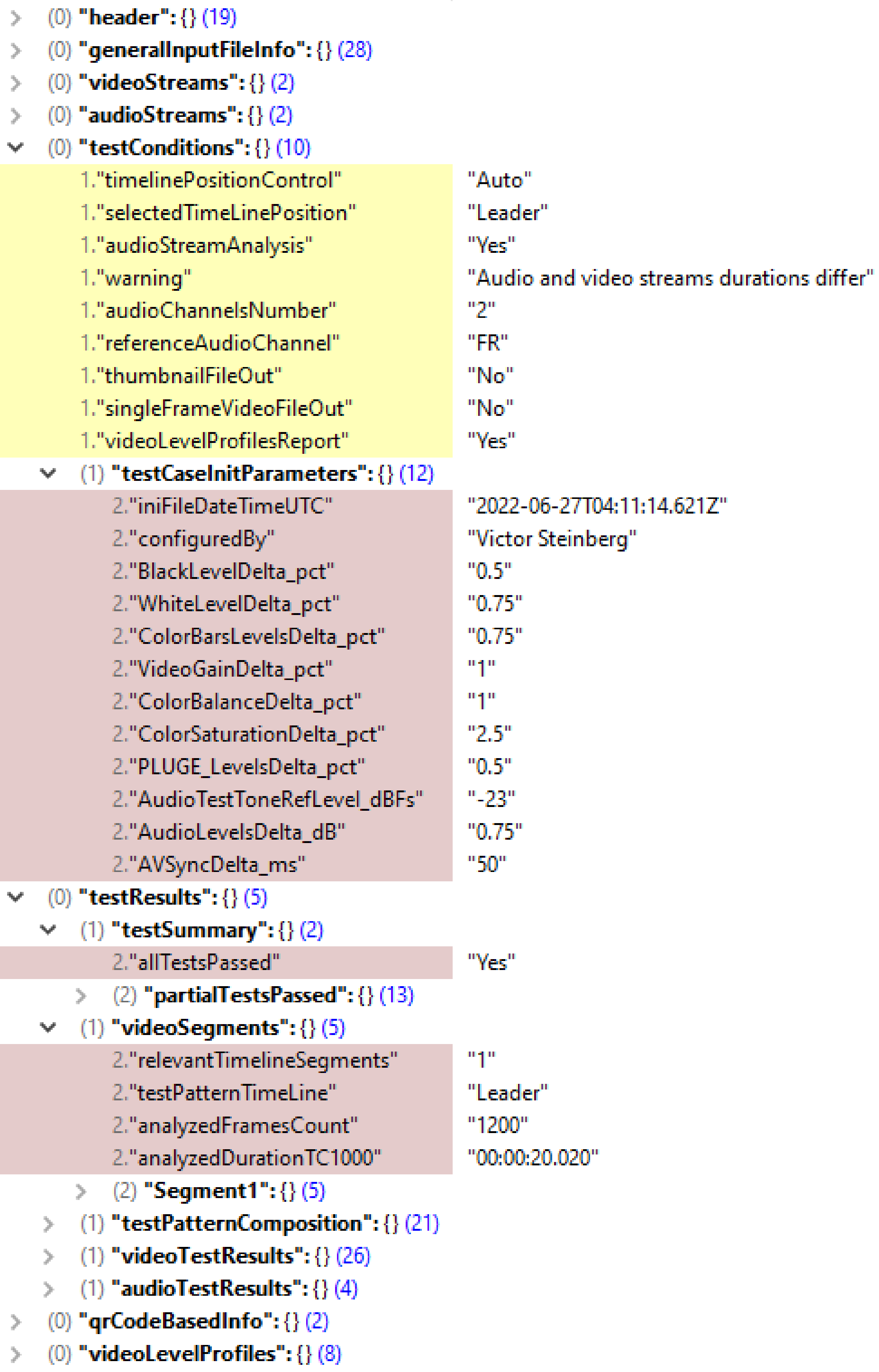
[-tm] flag enables ‘Trailer Mode’ (after main video segment), [-cd] flag enables ‘Captured Data’ asynchronous input mode

[-tfe] flag enables optional Thumbnail Frame Export to PNG file, [-sfe] flag enables optional Single Frame Export to lossless 16bpc MP4 file

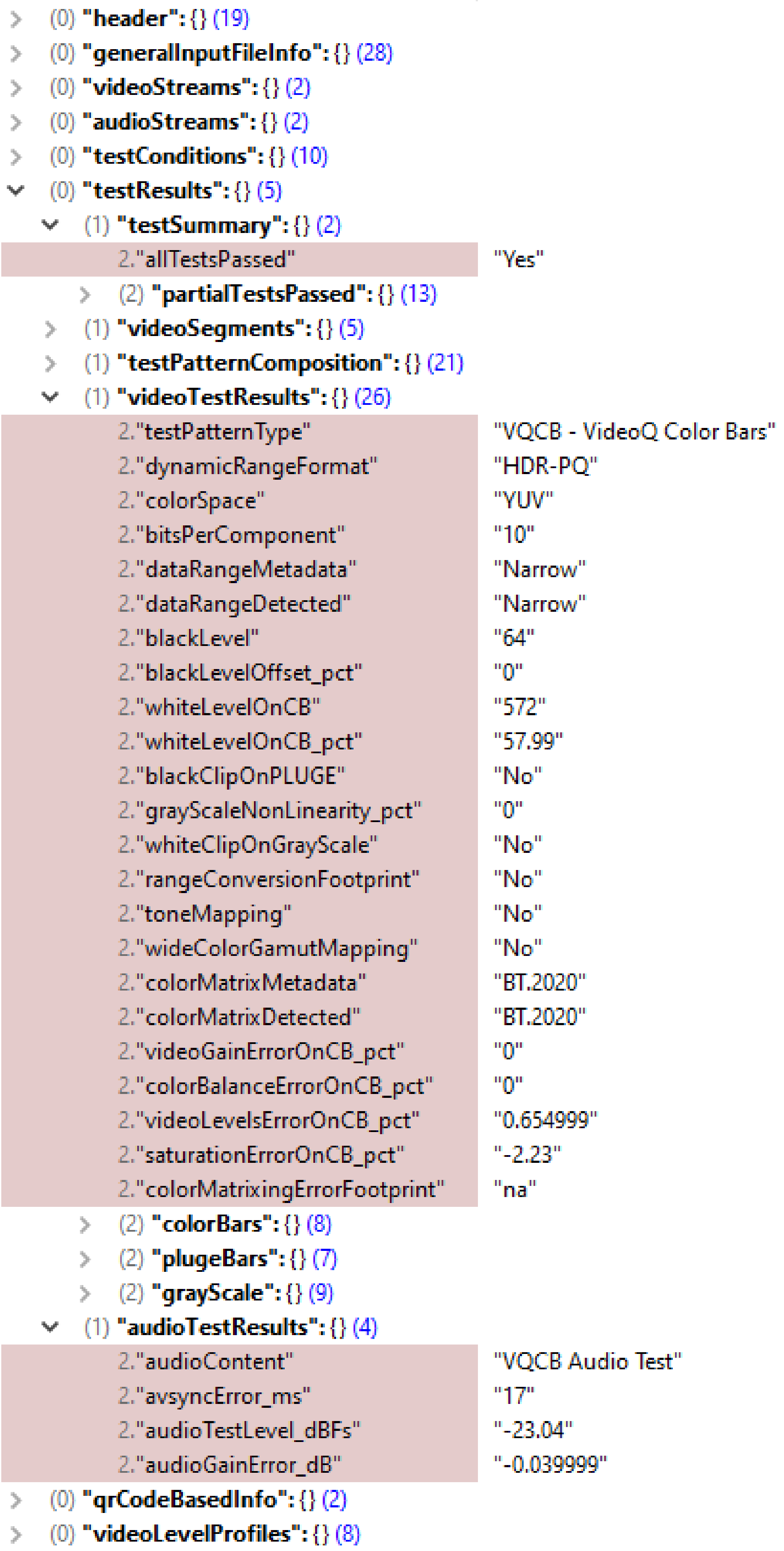
VQCBA – VideoQ Color Bars Analyzer Features

- VQCBA is especially useful when streaming in **multiple formats** or when **converting** between formats
- Supported frame sizes: from **480x270** to **8K UHD**
- VQCBA auto-detects and process **5 different types** of color bars tests
- Supported color bars test patterns types (with optional audio components):
  - **VideoQ VQCB** - dynamic **HDR** and **SDR AV** test patterns
  - **ITU BT.2111 HDR-PQ** and **HDR-HLG** color bars test patterns
  - **SMPTE RP219** test pattern
  - **SMPTE EG1** legacy test pattern
  - **Full frame color bars** - the most common video test pattern
- If input file contains continuous **audio test tone**, VQCBA detects this component and measures the audio test tone **level** in dBfs; result can be found in **audioTestResults** sub-section
- In case of VideoQ VQCB test analysis, VQCBA also provides:
  - throughput **audio gain** and **AV sync error** measurements results
  - decoded VQCB QR code data, shown in the **original_VQCB_Parameters** section of the Report

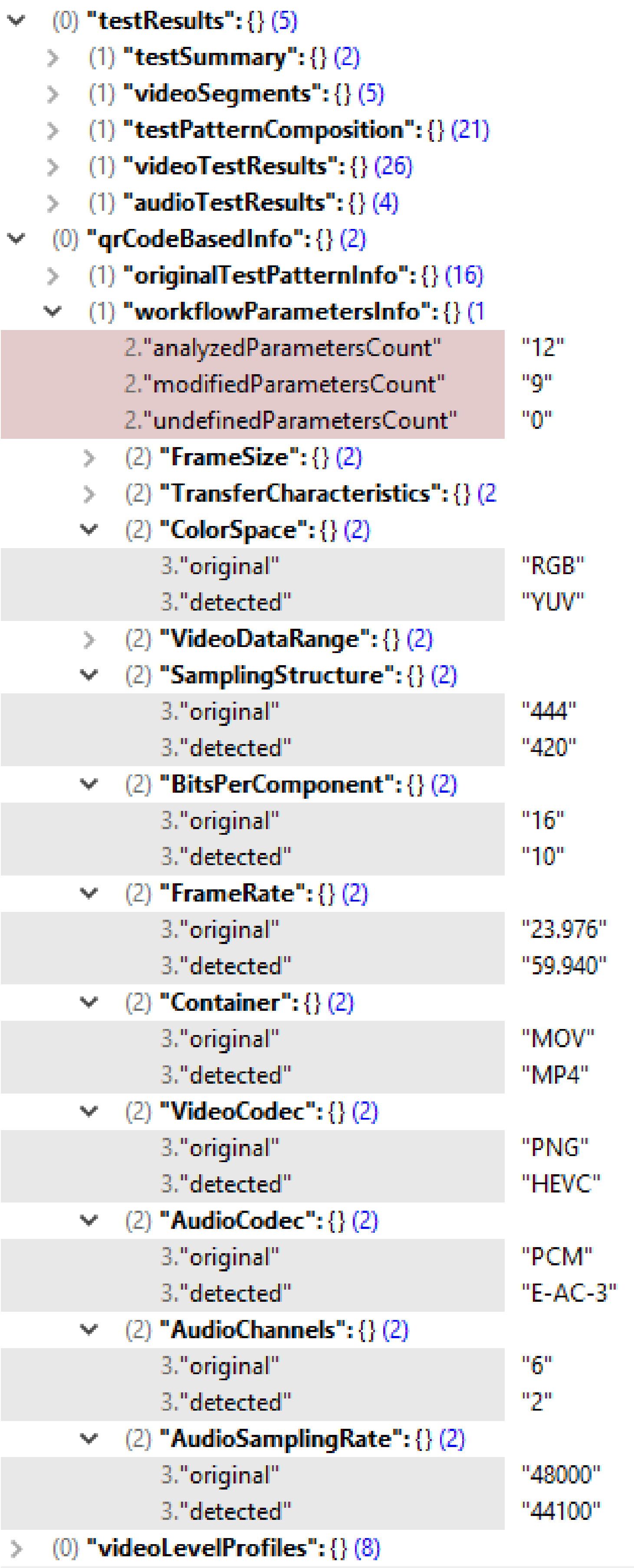
VQCBA – VideoQ Color Bars Analyzer Report Example



```
>  (0) "header": {} (19)
>  (0) "generalInputFileInfo": {} (28)
>  (0) "videoStreams": {} (2)
>  (0) "audioStreams": {} (2)
v  (0) "testConditions": {} (10)
       1."timelinePositionControl"          "Auto"
       1."selectedTimeLinePosition"         "Leader"
       1."audioStreamAnalysis"              "Yes"
       1."warning"                          "Audio and video streams durations differ"
       1."audioChannelsNumber"              "2"
       1."referenceAudioChannel"            "FR"
       1."thumbnailFileOut"                 "No"
       1."singleFrameVideoFileOut"          "No"
       1."videoLevelProfilesReport"         "Yes"
   v  (1) "testCaseInitParameters": {} (12)
          2."iniFileDateTimeUTC"            "2022-06-27T04:11:14.621Z"
          2."configuredBy"                  "Victor Steinberg"
          2."BlackLevelDelta_pct"           "0.5"
          2."WhiteLevelDelta_pct"           "0.75"
          2."ColorBarsLevelsDelta_pct"      "0.75"
          2."VideoGainDelta_pct"            "1"
          2."ColorBalanceDelta_pct"         "1"
          2."ColorSaturationDelta_pct"      "2.5"
          2."PLUGE_LevelsDelta_pct"         "0.5"
          2."AudioTestToneRefLevel_dBFs"    "-23"
          2."AudioLevelsDelta_dB"           "0.75"
          2."AVSyncDelta_ms"                "50"
v  (0) "testResults": {} (5)
   v  (1) "testSummary": {} (2)
          2."allTestsPassed"                "Yes"
      >  (2) "partialTestsPassed": {} (13)
   v  (1) "videoSegments": {} (5)
          2."relevantTimelineSegments"      "1"
          2."testPatternTimeLine"           "Leader"
          2."analyzedFramesCount"           "1200"
          2."analyzedDurationTC1000"        "00:00:20.020"
      >  (2) "Segment1": {} (5)
   >  (1) "testPatternComposition": {} (21)
   >  (1) "videoTestResults": {} (26)
   >  (1) "audioTestResults": {} (4)
>  (0) "qrCodeBasedInfo": {} (2)
>  (0) "videoLevelProfiles": {} (8)
```



```
>  (0) "header": {} (19)
>  (0) "generalInputFileInfo": {} (28)
>  (0) "videoStreams": {} (2)
>  (0) "audioStreams": {} (2)
>  (0) "testConditions": {} (10)
v  (0) "testResults": {} (5)
   v  (1) "testSummary": {} (2)
          2."allTestsPassed"                "Yes"
      >  (2) "partialTestsPassed": {} (13)
   >  (1) "videoSegments": {} (5)
   >  (1) "testPatternComposition": {} (21)
   v  (1) "videoTestResults": {} (26)
          2."testPatternType"               "VQCB - VideoQ Color Bars"
          2."dynamicRangeFormat"            "HDR-PQ"
          2."colorSpace"                    "YUV"
          2."bitsPerComponent"              "10"
          2."dataRangeMetadata"             "Narrow"
          2."dataRangeDetected"             "Narrow"
          2."blackLevel"                    "64"
          2."blackLevelOffset_pct"          "0"
          2."whiteLevelOnCB"                "572"
          2."whiteLevelOnCB_pct"            "57.99"
          2."blackClipOnPLUGE"              "No"
          2."grayScaleNonLinearity_pct"     "0"
          2."whiteClipOnGrayScale"          "No"
          2."rangeConversionFootprint"      "No"
          2."toneMapping"                   "No"
          2."wideColorGamutMapping"         "No"
          2."colorMatrixMetadata"           "BT.2020"
          2."colorMatrixDetected"           "BT.2020"
          2."videoGainErrorOnCB_pct"        "0"
          2."colorBalanceErrorOnCB_pct"     "0"
          2."videoLevelsErrorOnCB_pct"      "0.654999"
          2."saturationErrorOnCB_pct"       "-2.23"
          2."colorMatrixingErrorFootprint"  "na"
      >  (2) "colorBars": {} (8)
      >  (2) "plugeBars": {} (7)
      >  (2) "grayScale": {} (9)
   v  (1) "audioTestResults": {} (4)
          2."audioContent"                  "VQCB Audio Test"
          2."avsyncError_ms"                "17"
          2."audioTestLevel_dBFs"           "-23.04"
          2."audioGainError_dB"             "-0.039999"
>  (0) "qrCodeBasedInfo": {} (2)
>  (0) "videoLevelProfiles": {} (8)
```



```
v  (0) "testResults": {} (5)
   >  (1) "testSummary": {} (2)
   >  (1) "videoSegments": {} (5)
   >  (1) "testPatternComposition": {} (21)
   >  (1) "videoTestResults": {} (26)
   >  (1) "audioTestResults": {} (4)
v  (0) "qrCodeBasedInfo": {} (2)
   >  (1) "originalTestPatternInfo": {} (16)
   v  (1) "workflowParametersInfo": {} (1
          2."analyzedParametersCount"       "12"
          2."modifiedParametersCount"       "9"
          2."undefinedParametersCount"      "0"
      >  (2) "FrameSize": {} (2)
      >  (2) "TransferCharacteristics": {} (2
      v  (2) "ColorSpace": {} (2)
             3."original"                   "RGB"
             3."detected"                   "YUV"
      >  (2) "VideoDataRange": {} (2)
      v  (2) "SamplingStructure": {} (2)
             3."original"                   "444"
             3."detected"                   "420"
      v  (2) "BitsPerComponent": {} (2)
             3."original"                   "16"
             3."detected"                   "10"
      v  (2) "FrameRate": {} (2)
             3."original"                   "23.976"
             3."detected"                   "59.940"
      v  (2) "Container": {} (2)
             3."original"                   "MOV"
             3."detected"                   "MP4"
      v  (2) "VideoCodec": {} (2)
             3."original"                   "PNG"
             3."detected"                   "HEVC"
      v  (2) "AudioCodec": {} (2)
             3."original"                   "PCM"
             3."detected"                   "E-AC-3"
      v  (2) "AudioChannels": {} (2)
             3."original"                   "6"
             3."detected"                   "2"
      v  (2) "AudioSamplingRate": {} (2)
             3."original"                   "48000"
             3."detected"                   "44100"
>  (0) "videoLevelProfiles": {} (8)
```

VQCSA – Compression Stress Analyzer, Usage Example

Test conditions:

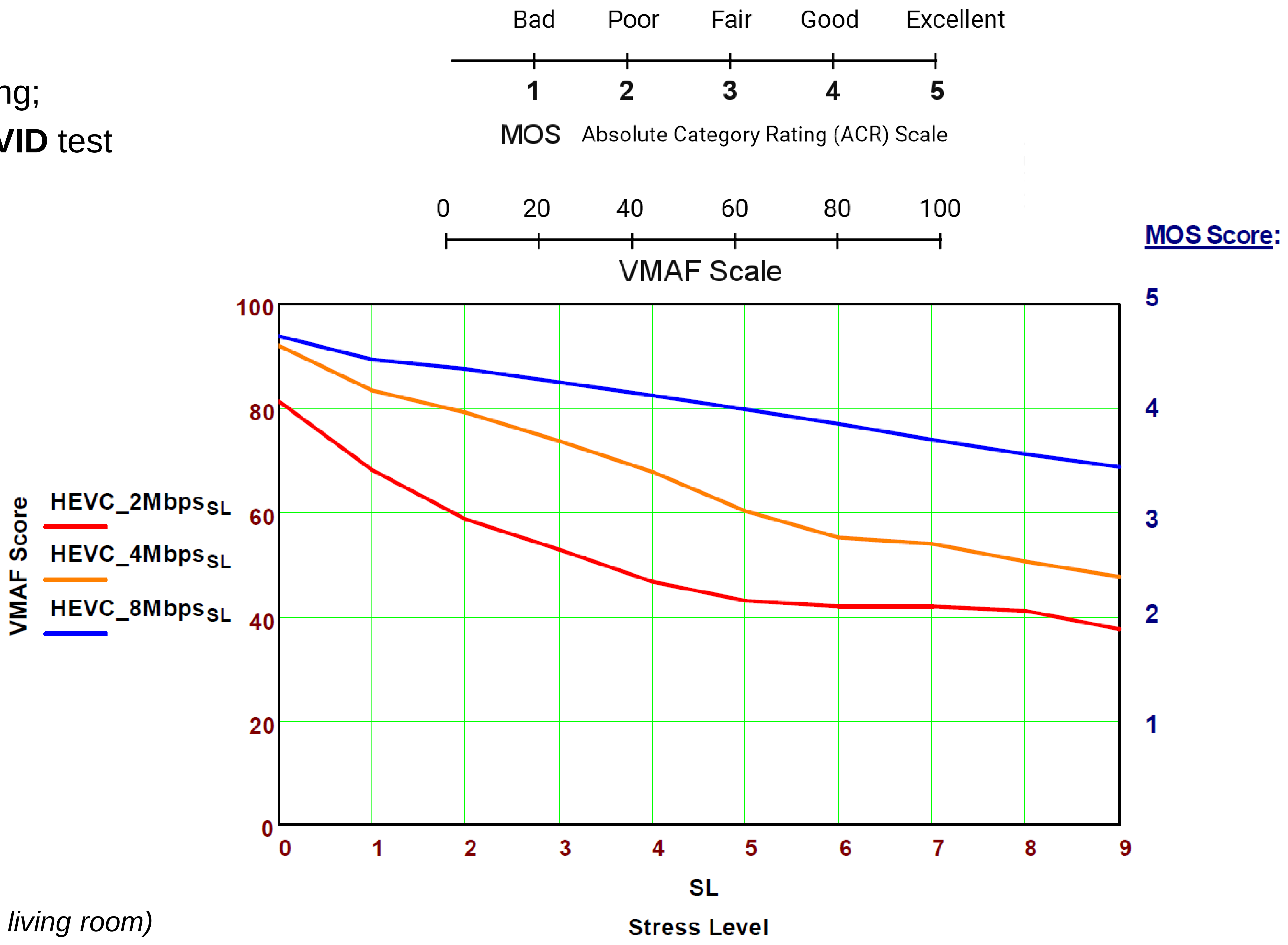
HD 60fps HEVC – **2**, **4, 8 Mbps** encoding;

Medium **S**tress **R**ange (MSR) **VQCST_VID** test

Bitrate, Mbps:

Stress Level:

$$\mathrm{VMAF} \equiv \begin{array}{ccc|c} 2 & 4 & 8 & \\ 81.3 & 92.0 & 93.8 & 0 \\ 68.1 & 83.4 & 89.3 & 1 \\ 58.7 & 79.1 & 87.5 & 2 \\ 52.8 & 73.7 & 85.0 & 3 \\ 46.7 & 67.7 & 82.4 & 4 \\ 43.0 & 60.2 & 79.7 & 5 \\ 42.0 & 55.1 & 77.0 & 6 \\ 41.9 & 54.0 & 73.9 & 7 \\ 41.1 & 50.5 & 71.1 & 8 \\ 37.6 & 47.6 & 68.7 & 9 \end{array}$$

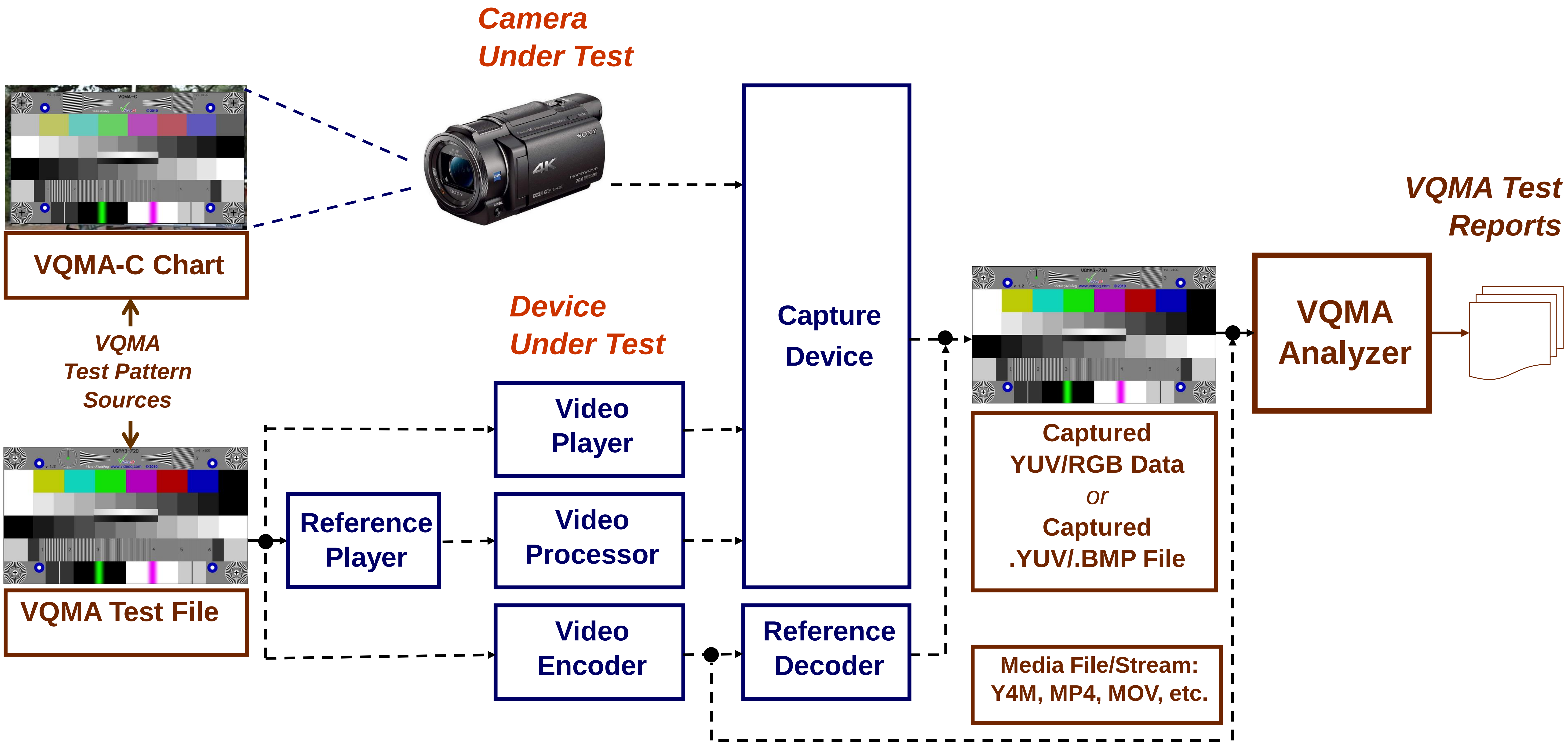


VMAF model used: Netflix vmaf_v0.6.1.pkl (HD, living room)

VQMA – Video Processing Chain Analyzer

- 4th generation of VideoQ best-selling software product,
  suitable for any video format, any frame size from ***192x108*** *to* ***4096x3072*** *(****4K versions****) or* ***7680x4320*** *(****8K versions****), any frame rate,* ***HDR & WCG support coming soon***
- Software executable under Windows^{TM} (XP, 7, 8, 10, 11)
- USB dongle copy-protected, dongle-per-workstation
- Automated analysis on the companion VQMA Matrix Test Pattern
- Variety of VQMA Test Pattern formats: Optical Chart, File, Signal, Stream
- Unique patented algorithms for accurate & fast measurements (typically 2-5 seconds)
- Built-in YUV/RGB Waveform Scope
- Noise Measurement and Waveform Scope work on any static image
- Windows GUI Mode for R&D and product verification
- Command Line Interface (Batch) Mode for automated QA/QC operation

VQMA Analyzer Workflow Variants



Camera Under Test

VQMA-C Chart

VQMA Test Pattern Sources

VQMA Test File

Reference Player

Device Under Test

Video Player

Video Processor

Video Encoder

Capture Device

Reference Decoder

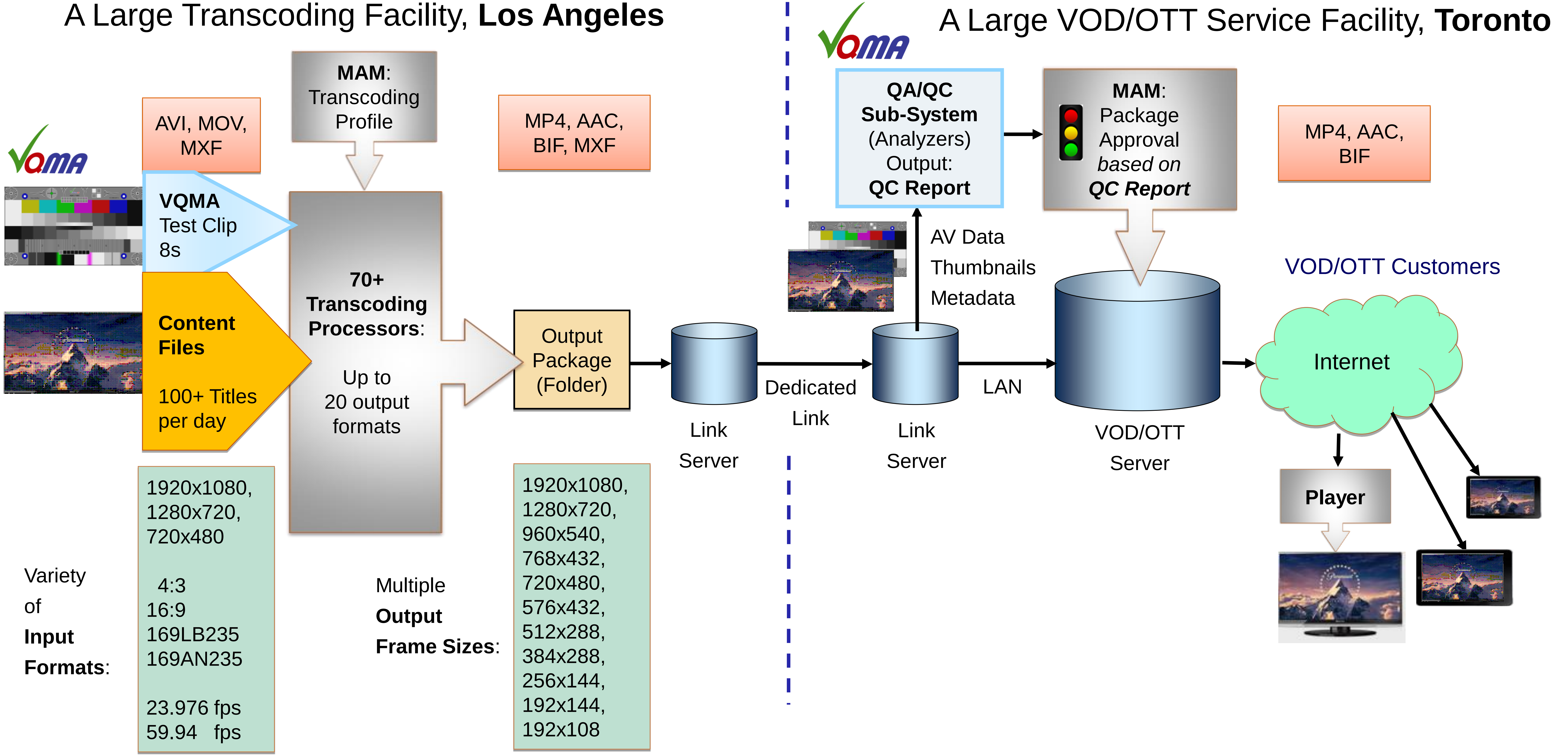
Captured YUV/RGB Data *or* Captured .YUV/.BMP File

Media File/Stream: Y4M, MP4, MOV, etc.

VQMA Analyzer

VQMA Test Reports

VQMA Analyzer Workflow Test Application Example



A Large Transcoding Facility, **Los Angeles**

A Large VOD/OTT Service Facility, **Toronto**

AVI, MOV, MXF

MAM: Transcoding Profile

MP4, AAC, BIF, MXF

VQMA Test Clip 8s

Content Files

100+ Titles per day

70+ Transcoding Processors:

Up to 20 output formats

Output Package (Folder)

Link Server

Dedicated Link

Link Server

LAN

VOD/OTT Server

QA/QC Sub-System (Analyzers) Output: **QC Report**

MAM: Package Approval *based on* ***QC Report***

MP4, AAC, BIF

AV Data

Thumbnails

Metadata

VOD/OTT Customers

Internet

Player

Variety of **Input Formats**:

1920x1080,
1280x720,
720x480

4:3
16:9
169LB235
169AN235

23.976 fps
59.94 fps

Multiple **Output Frame Sizes**:

1920x1080,
1280x720,
960x540,
768x432,
720x480,
576x432,
512x288,
384x288,
256x144,
192x144,
192x108

Pack 2: AV Files Conversion, Encoding and Transcoding

- **VQMINF** – Media File Info Report Generator

  VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in UTF8 JSON format.
- **VQBIF** **–** BIF (Base Index Frames) Files Verifier

  This module reads BIF thumbnail files, checks the parameters for compliance, and extracts numbered JPEG files as a reference decoder
- **VQBLA** – Bitrate Ladder Analyzer

  This module reads input media file, then checks multiple video parameters and builds the encoding quality score profiles for a given codec and preconfigured set of output frame sizes and bitrates
- **VQC** – HDR-SDR Converter

  https://videoq.com/vqc.html
- **VQCSA** – Compression Stress Analyzer, companion program for **VQCST** test patterns.

  http://www.videoq.com/Downloads/VideoQ_StressTracker_Training_Presentation_PPT.pdf

  This module reads VQCST test pattern file at particular workflow test point, measures VMAF scores for particular Compression Stress parameters and codecs settings, then builds Compression Stress Response Profiles
- **VQLPN** – Audio Loudness Profiler and Normalizer

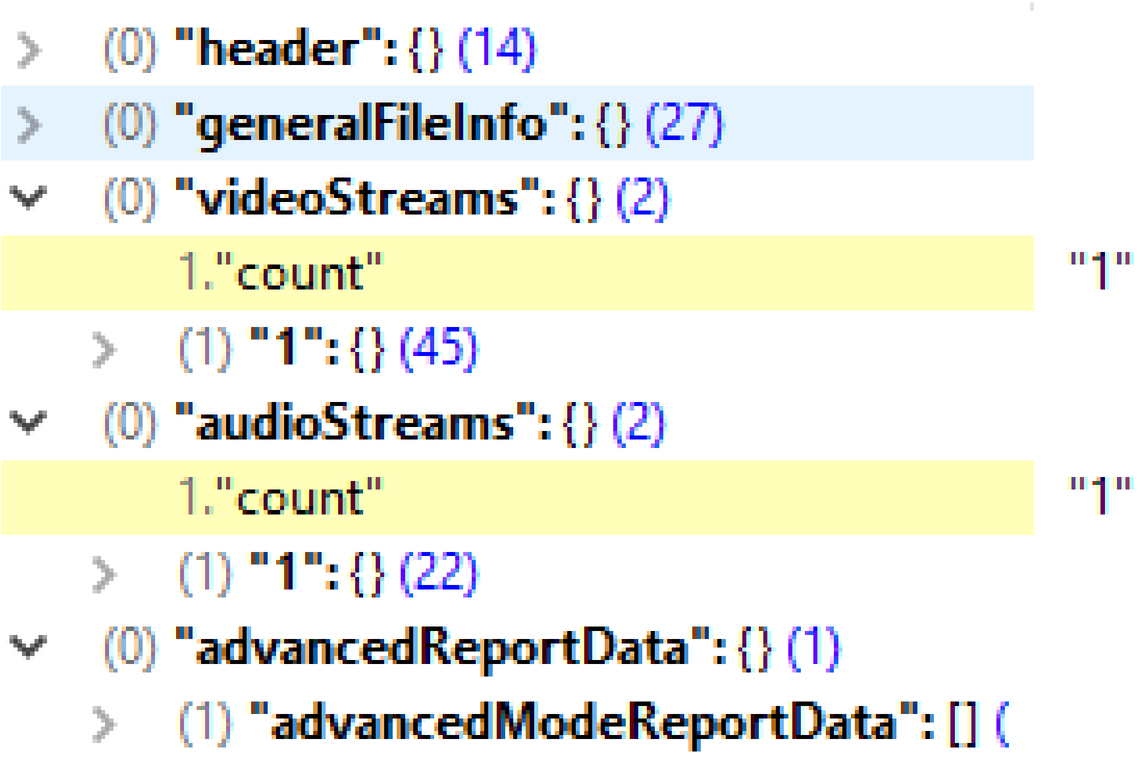
  https://videoq.com/vqlpn.html

  It measures and normalizes the audio stream loudness parameters in accordance with ITU-R BS.1770-4 (USA ATSC RP A85, EBU R128).
- **VQTSF** – Transcoding Segments Finder

  This module reads input media file and detects critical timeline segments positions (preferred key frames); it also measures content activities and builds related timeline profile

VQMINF – Media File Info Report Generator

- VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in **UTF8 JSON** format
- It can read **HDR-PQ**, **HDR-HLG** and **SDR** video as well as **LOG** video metadata
- VQMINF uses standard SHA3 and MD5 libraries to calculate unique **vqminfEssenceID** string and input file **MD5 hash** string
- Report file contains the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **textStreams**: technical parameters of text stream(s)
  - **imageStreams**: technical parameters of image stream(s)
  - **advancedReportData**: (*all raw media info strings, useful for manual analysis of complicated cases*)



```
>  (0) "header":{} (14)
>  (0) "generalFileInfo":{} (27)
v  (0) "videoStreams":{} (2)
        1."count"                              "1"
   >  (1) "1":{} (45)
v  (0) "audioStreams":{} (2)
        1."count"                              "1"
   >  (1) "1":{} (22)
v  (0) "advancedReportData":{} (1)
   >  (1) "advancedModeReportData":[] (
```

CLI interface: **vqminf** [-noard] [-nohash] -i inFilePath [-o or -o outFilePath]

VQBIF – BIF (Base Index Frames) Files Verifier

- VQBIF reads 1, 2 or 3 .BIF thumbnails files, and creates Report in JSON format.
- Validate BIF files for:
  - File integrity
  - Size [each frame/picture]
    - For high bitrate it should be …
    - For low bitrate it should be …
  - Coding quality [jpeg compression quality]
    - For high bitrate it should be …
    - For low bitrate it should be …
  - Offset (time offset between each grab)

- Check for **time offset correlation** (*function of the specified media file duration*)
- Check for **cumulative amount** of index frame images
- **Decode** the BIF file(s) as a **reference decoder** (*extract several sets of numbered .JPG files*)

CLI interface: **vqbif** -i in1.BIF [in2.BIF] [in3.BIF] [-o outPath] [-j/-q] [jpeg_folder] [-r Ref_Framewise_Separation_ms]

VQBLA – Bitrate Ladder Analyzer

- VQBLA reads .MOV, MP4, etc. media file, and analyzes its content enabling optimal downstream transcoder settings
- It measures Intra-frame and Inter-frame Activities
- VQBLA builds statistics (histograms) for a range of critical video parameters
- The key stage of VQBLA work is estimation of **expected levels of compression artifacts** and **down-scaling distortions**
- Finally, VQBLA creates machine-readable Report in JSON format showing:
    - Measured video parameters statistics: **Intra-frame Activity** and **Inter-frame Activity**
    - The BLA **Draft Bitrate Ladder** as an array of the expected **Quality Score** vs. **Bitrate** and **Frame Size**
- VQBLA also creates an optional .PNG image file showing (see next slide):
    - Activities Timeline Profile Plot
    - Activities Statistics Bargraphs
    - Bitrate Ladder Plot with critical bitrate values marked

CLI interface: **vqbla** [-cs NumOfSec] [-p] -i inFilePath [-o [outFilePath]]

TOC

VQBLA – Bitrate Ladder Analyzer, Plot Example

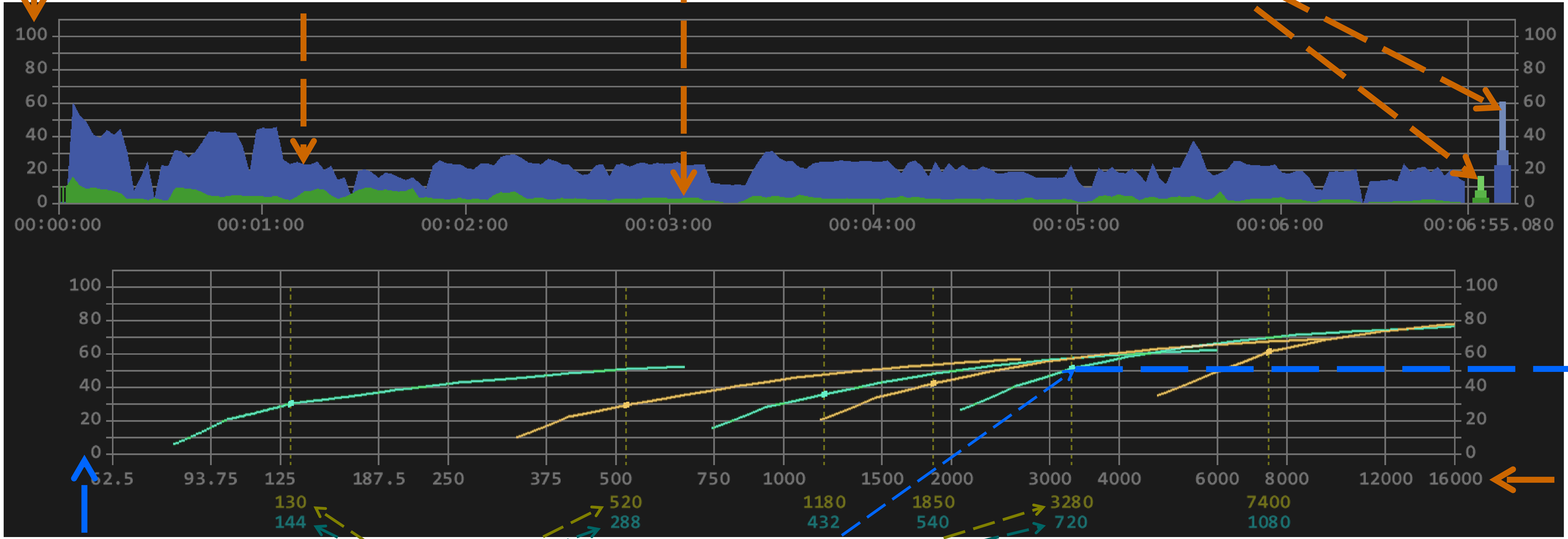
PTOC

Activity, %

Intra-frame Activity
(Static Details)

Inter-frame Activity
(Dynamic Details)

Activities Statistics Bargraphs



Quality Score 0 .. 100

Bitrate, kbps

Quality Score

Bitrate Ladder Items:
Critical Bitrates, kbps
vs. Frame Heights

Quality Score
and Critical Bitrate, kbps
for this Frame Height

Encoding with bitrates lower than critical results in compression artifacts, usually described as “Noticeable and annoying”, bitrates above critical – as “Noticeable, but not annoying”

VQC – HDR-SDR Converter

- VQC reads media file or image files sequence, then converts input file HDR-PQ, HDR-HLG, or SDR **dynamic range** and **color space format** to the specified output format
- Frame sizes: from **1280x720** to **8K UHD**
- Supported input and output color primaries:
  - **BT.709** (aka NCG = Narrow Color Gamut, *only for SDR*)
  - **BT.2020** (aka WCG = Wide Color Gamut)
  - **P3** (SMPTE432, aka ECG = Expanded Color Gamut)
- VQC internal workflow consists of several stages:
  - **Reading** media file or image files sequence
  - **Measurement** of video frames **parameters**
  - **Conversion** of video data to the specified **dynamic range** and **color space** format
  - **Encoding** of video data to the specified output file(s) format
  - Creation of machine-readable **JSON Report** and optional **PNG Plot** file

CLI interface: **vqc** [-c configFilePath] -i inFilePath -o outFilePath

VQCSA – Compression Stress Analyzer, Usage Example

Test conditions:

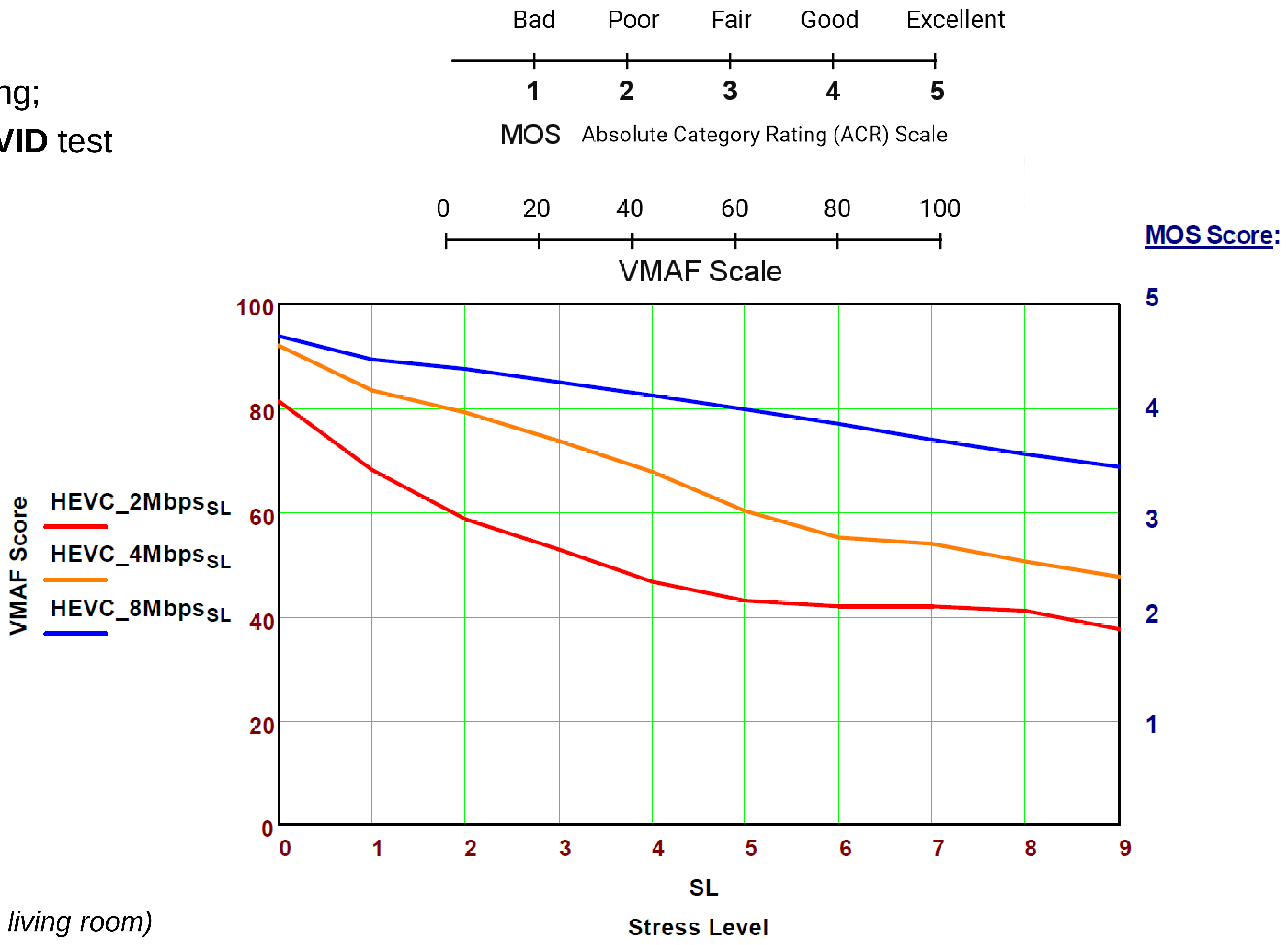
HD 60fps HEVC – **2**, **4, 8 Mbps** encoding;

Medium **S**tress **R**ange (MSR) **VQCST_VID** test

Bitrate, Mbps:

Stress Level:

$$\mathbf{VMAF} \equiv \begin{array}{ccc|c} 2 & 4 & 8 & \\ \hline 81.3 & 92.0 & 93.8 & 0 \\ 68.1 & 83.4 & 89.3 & 1 \\ 58.7 & 79.1 & 87.5 & 2 \\ 52.8 & 73.7 & 85.0 & 3 \\ 46.7 & 67.7 & 82.4 & 4 \\ 43.0 & 60.2 & 79.7 & 5 \\ 42.0 & 55.1 & 77.0 & 6 \\ 41.9 & 54.0 & 73.9 & 7 \\ 41.1 & 50.5 & 71.1 & 8 \\ 37.6 & 47.6 & 68.7 & 9 \end{array}$$



VMAF model used: Netflix vmaf_v0.6.1.pkl (HD, living room)

VQLPN – Audio Loudness Profiler & Normalizer

- VQLPN reads media file, containing audio stream(s), measures and normalizes audio loudness.
- VQLPN supports MP4, MOV, MXF, WAV, W64, AAC, AC3, EAC3, etc., and various audio stream formats:
  - any bit depth, bit rate and sampling rate, all audio codecs supported by ffmpeg
  - multi-channel and multi-track formats: **1.0, 2.0, 5.1, 7.1**
- It measures the audio stream loudness parameters in accordance with Recommendation ITU-R **BS.1770-4** (*USA **ATSC RP A85**, EBU **R128***). Editable *.INI file stores test configuration and target parameters.
- Sorts audio segments by types (**regular audio**, **test tone**, **mute**)
- Finally, VQLPN creates detailed Report in JSON format, including **Momentary Loudness Profile** data array at 100 ms step interval
- Configurable outputs:
  - **Audio file** in the **desired format,** optionally **normalized** to the desired **Integrated Loudness** target
  - **PNG image file** showing **momentary loudness** time-line profile **plot**, **loudness** statistics **bargraph**, **upper levels histogram**, as well as other useful markers and values
- Optional stand-alone utility modules: **VQLPC** *Correlator*, **VQLPP** *Plotter*, **VQILM** *IL Meter*

CLI interface: **vqlpn** [-j jsonFilePath] [-c configFilePath] -i inMediaFilePath [-o [outFilePath]]

TOC

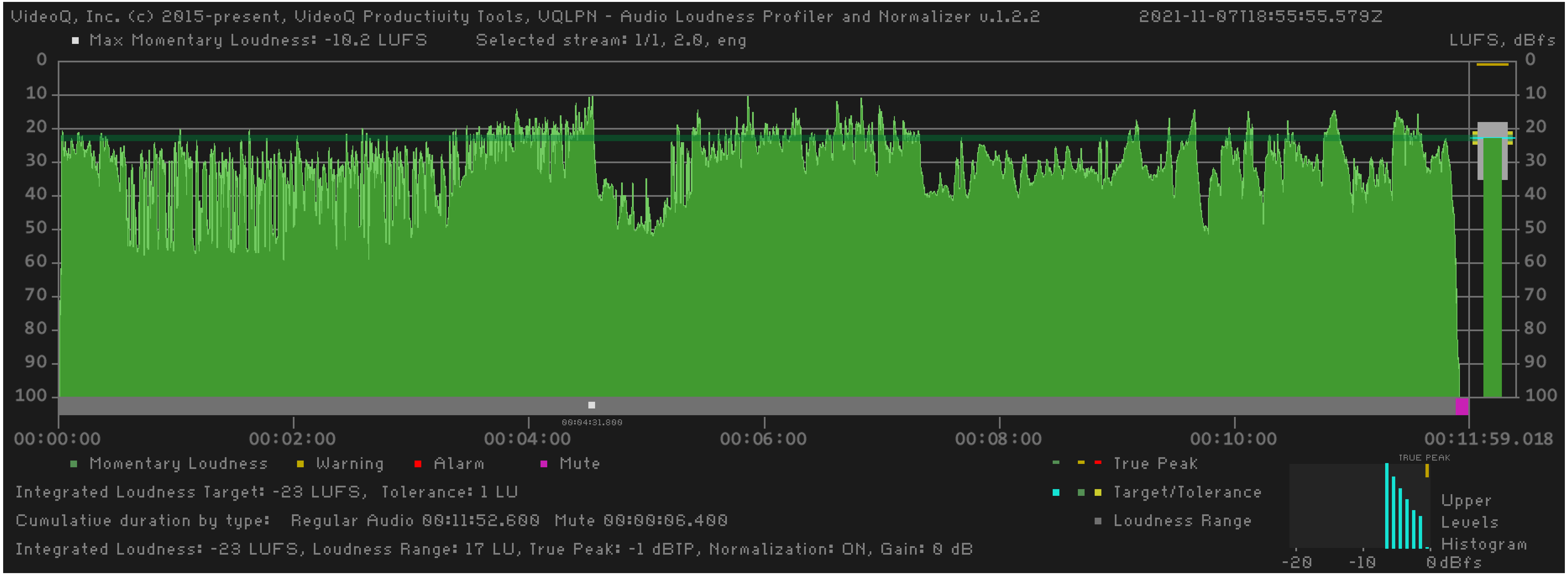
VQLPN – Loudness Profiler & Normalizer, Plot Example 1

PTOC

Professional clip with 6 seconds long Mute Fragment at the end

***Normalized audio stream** – **Integrated Loudness** is exactly equal to **-23 LUFS** target value*

True Peak** value is **quite high



VideoQ, Inc. (c) 2015-present, VideoQ Productivity Tools, VQLPN - Audio Loudness Profiler and Normalizer v.1.2.2 2021-11-07T18:55:55.579Z

■ Max Momentary Loudness: -10.2 LUFS Selected stream: 1/1, 2.0, eng

Integrated Loudness Target: -23 LUFS, Tolerance: 1 LU

Cumulative duration by type: Regular Audio 00:11:52.600 Mute 00:00:06.400

Integrated Loudness: -23 LUFS, Loudness Range: 17 LU, True Peak: -1 dBTP, Normalization: ON, Gain: 0 dB

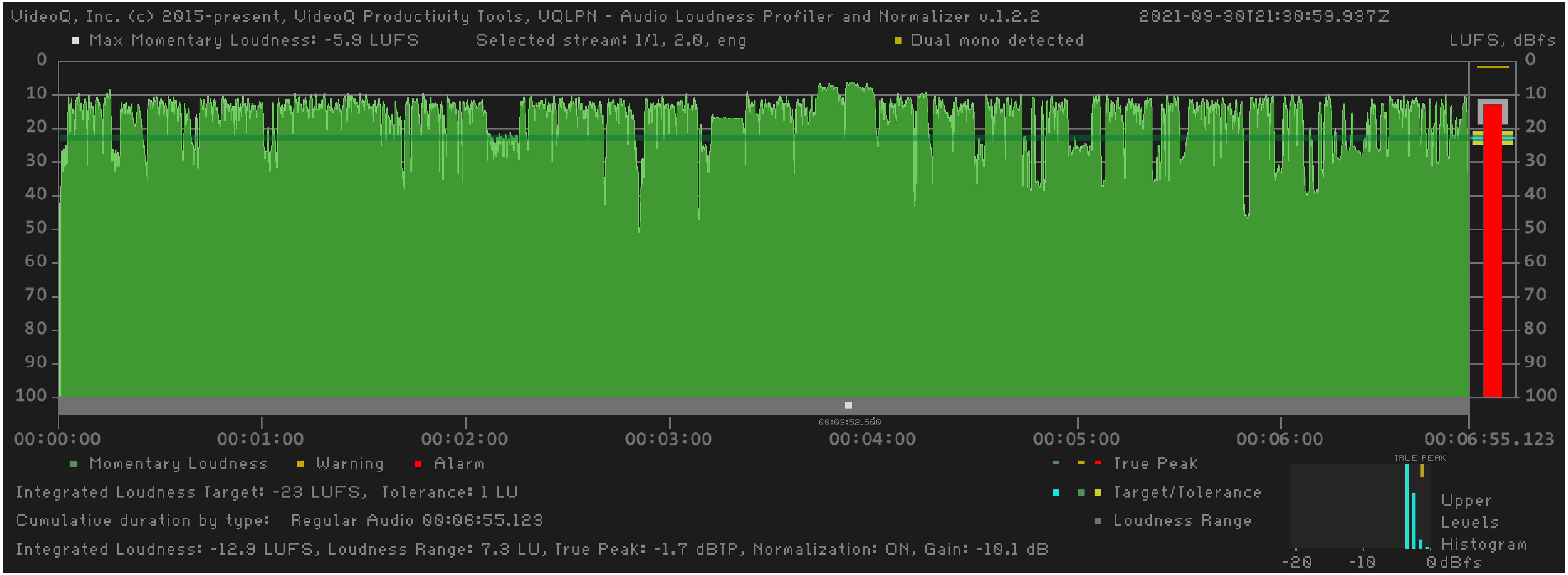
TOC

VQLPN – Loudness Profiler & Normalizer, Plot Example 2

PTOC

Legacy content:

Integrated Loudness** is **much higher** than **-23 LUFS** target value, and **True Peak** value is **quite high



```
VideoQ, Inc. (c) 2015-present, VideoQ Productivity Tools, VQLPN - Audio Loudness Profiler and Normalizer v.1.2.2          2021-09-30T21:30:59.937Z
■ Max Momentary Loudness: -5.9 LUFS        Selected stream: 1/1, 2.0, eng                    ■ Dual mono detected

Integrated Loudness Target: -23 LUFS,  Tolerance: 1 LU
Cumulative duration by type:  Regular Audio 00:06:55.123
Integrated Loudness: -12.9 LUFS, Loudness Range: 7.3 LU, True Peak: -1.7 dBTP, Normalization: ON, Gain: -10.1 dB
```

VQTSF – Transcoding Segments Finder

- VQTSF reads .MOV, MP4, etc. media file, and analyzes its content enabling optimal downstream transcoder settings. This is achieved via detection of critical timeline segments positions (**preferred key frames**). VQTSF also measures **content activities** and builds related timeline profile.
- VQTSF creates machine-readable Report in JSON format showing the following sections:
    - **header**
    - **generalFileInfo**
    - **videoStream**:
    - **videoSegments**
    - **activitiesStatistics**_dBfs
        - spatial
        - temporal
    - **timeLineActivitiesProfiles**_dBfs
        - spatial
        - temporal

CLI interface: **vqtsf** [-noplot] [-slo] -i inFilePath [-o [outFilePath]]

TOC

VQTSF – Transcoding Segments Finder, Plot Example

PTOC

File duration: 6min 55s. 74 segments found, segment durations from 0.64s to 25.2s**.**

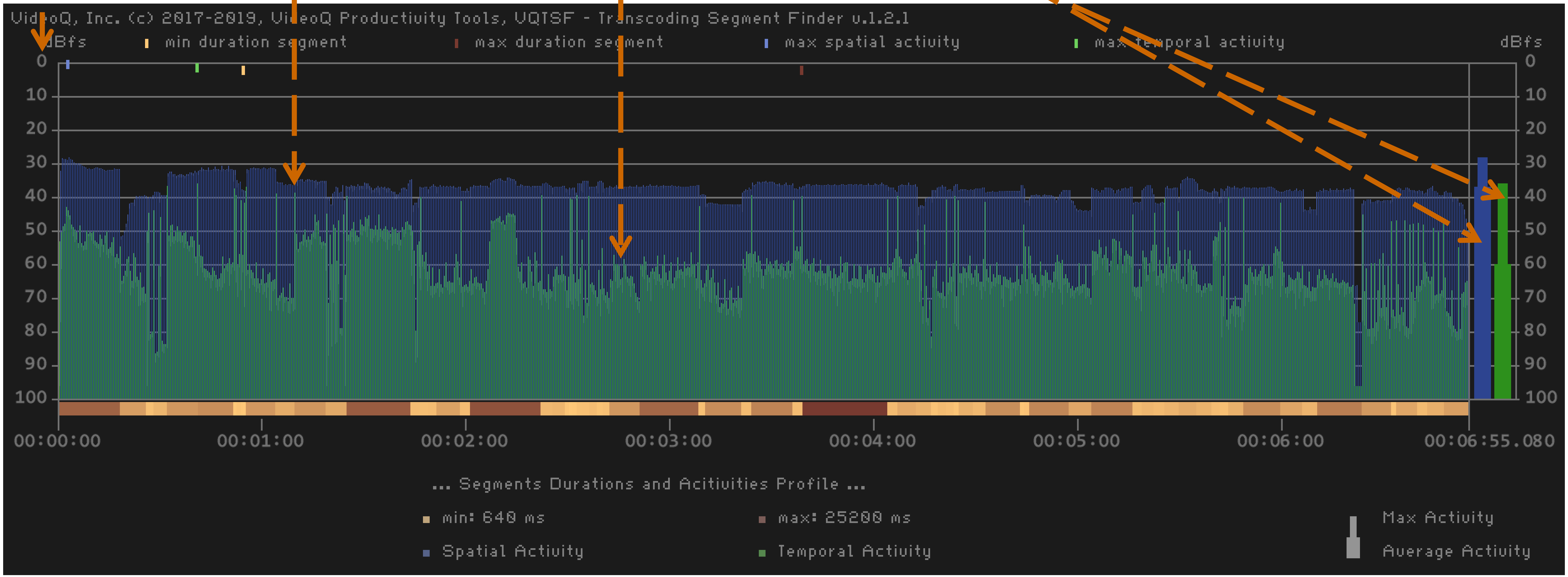
***Activity** profiles are of **medium** strength, so we can get relatively **good quality** at relatively **low bitrates**.*

Activity, %

Intra-frame Activity (Static Details)

Inter-frame Activity (Dynamic Details)

Activities Statistics Bargraphs



Pack 3: VQPT Modules for AV Content Tests

- VQMINF – Media File Info Report Generator

  VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in UTF8 JSON format.
  It can read HDR-PQ, HDR-HLG and SDR video as well as LOG video metadata.
  VQMINF also calculates unique vqminfEssenceID string and input file MD5 hash string.

- VQCFA – Captions Files Analyzer

  *VQCFA reads .**WebVTT**, .**VTT**, .**XML** or .**TTML** caption file, measures **caption parameters** and checks them against the predefined, auto-switchable, and/or CLI specified, thresholds; VQCFA also exports full captions event listing text file.*

- VQFP – Video Frames Profiler

  *VQFP reads media file, measures video frames active image formats, frame cadences, sharpness, noise level, video data and light level statistics, detects video segments, and calculate **timeline profiles** of video levels, light level, sharpness and details activity.*

- VQLPC – Loudness Profiles Correlator, companion program for **VQLPN** module

  *VQLPC reads two JSON report files created by **VQLPN** module (Reference and Test inputs), detects and compares momentary loudness time-line profiles of two inputs. Finally, VQLPC calculates the **correlation of two inputs**.*

- VQLPN – Audio Loudness Profiler and Normalizer

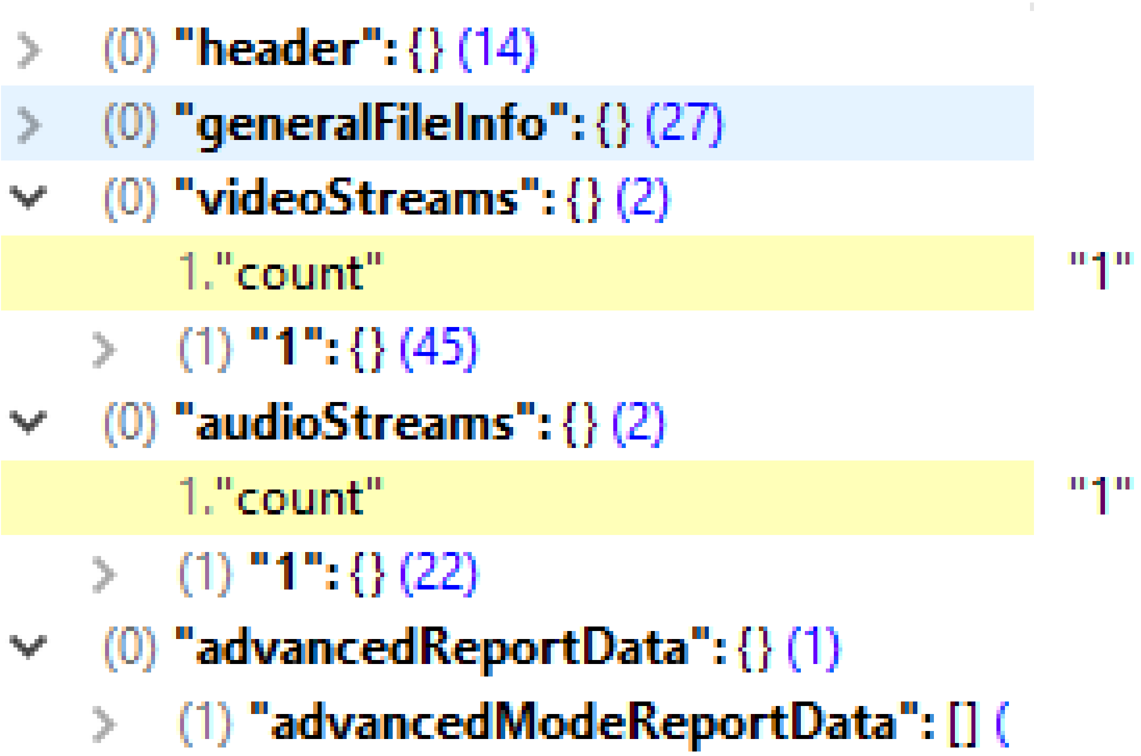
  https://videoq.com/vqlpn.html
  It measures and normalizes the audio stream loudness parameters in accordance with ITU-R BS.1770-4 (USA ATSC RP A85, EBU R128).

- VQPLA – Picture Levels Analyzer

  *VQPLA reads SDR/HDR media file, detects video segments, and calculates **timeline profiles** of **video and light levels**.*

VQMINF – Media File Info Report Generator

- VQMINF reads a wide variety of media files (MOV, MXF, WAV, JPG, etc.) and creates Report in **UTF8 JSON** format
- It can read **HDR-PQ**, **HDR-HLG** and **SDR** video as well as **LOG** video metadata
- VQMINF uses standard SHA3 and MD5 libraries to calculate unique **vqminfEssenceID** string and input file **MD5 hash** string
- Report file contains the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStreams**: encoding and format information
  - **audioStreams**: encoding and format information
  - **textStreams**: technical parameters of text stream(s)
  - **imageStreams**: technical parameters of image stream(s)
  - **advancedReportData**: (*all raw media info strings, useful for manual analysis of complicated cases*)



```
>  (0) "header": {} (14)
>  (0) "generalFileInfo": {} (27)
v  (0) "videoStreams": {} (2)
        1."count"                              "1"
     >  (1) "1": {} (45)
v  (0) "audioStreams": {} (2)
        1."count"                              "1"
     >  (1) "1": {} (22)
v  (0) "advancedReportData": {} (1)
     >  (1) "advancedModeReportData": [] (
```

CLI interface: **vqminf** [-noard] [-nohash] -i inFilePath [-o or -o outFilePath]

VQCFA – Captions Files Analyzer

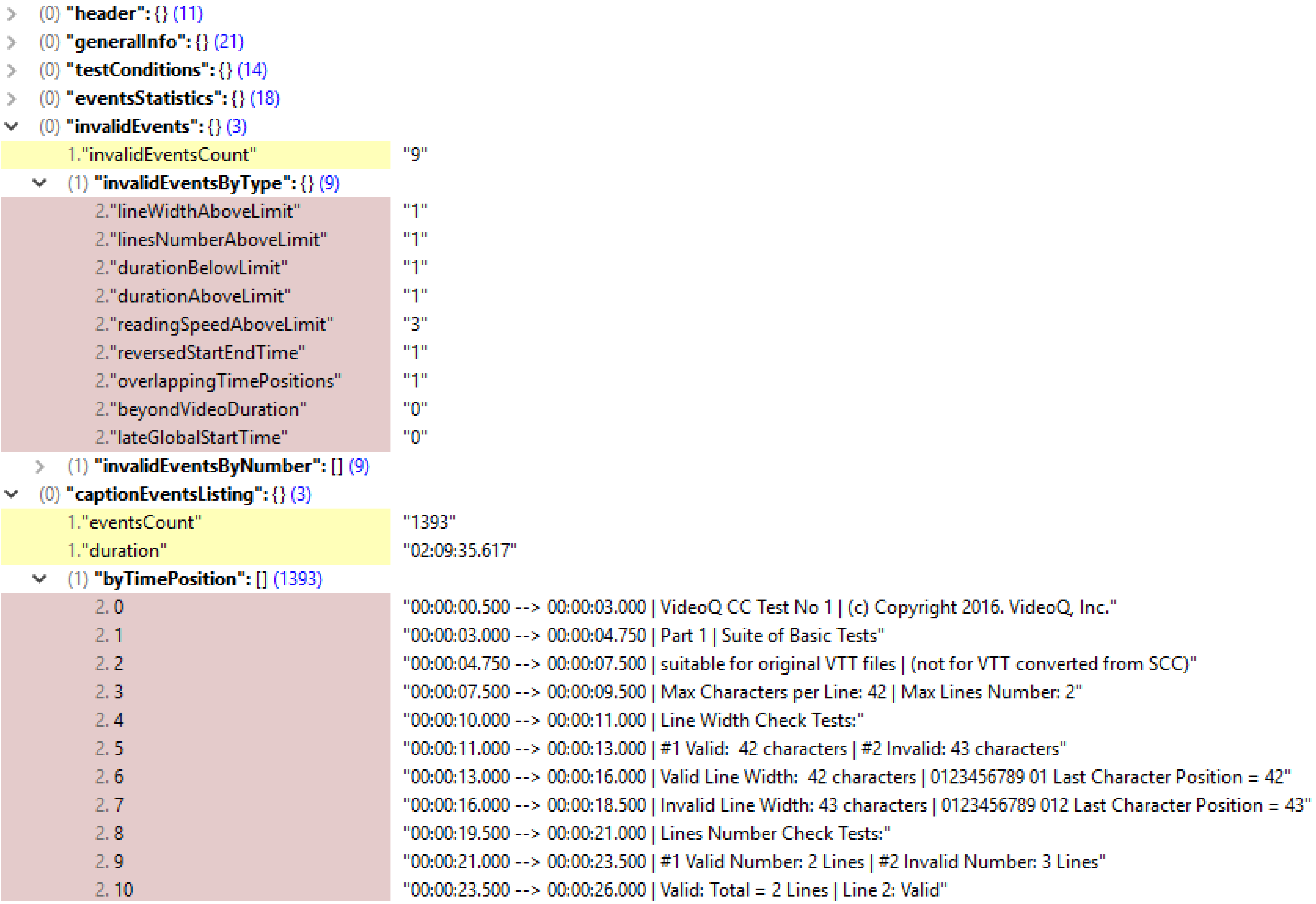
- VQCFA reads .**WebVTT**, .**VTT**, .**XML** or .**TTML** caption file
- It measures **caption parameters** and checks them against the predefined, auto-switchable, and/or CLI specified, **thresholds**.
- Finally, VQCFA creates machine-readable Report in JSON format showing the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **testConditions**: user-selected and auto-configured test session parameters
  - **eventsStatistics**: measured statistics
  - **invalidEvents**, which is further sub-divided into two sub-sections:
    - invalidEventsByType
    - invalidEventsByNumber
  - **Full Captions Event Listing** (*decoded unformatted UTF text lines with timestamps*)

CLI interface: **vqcfa** [-p int int] [-lw int -ln int] -i inFilePath [-o [outFilePath]]

TOC

VQCFA – Captions Files Analyzer, Report Example

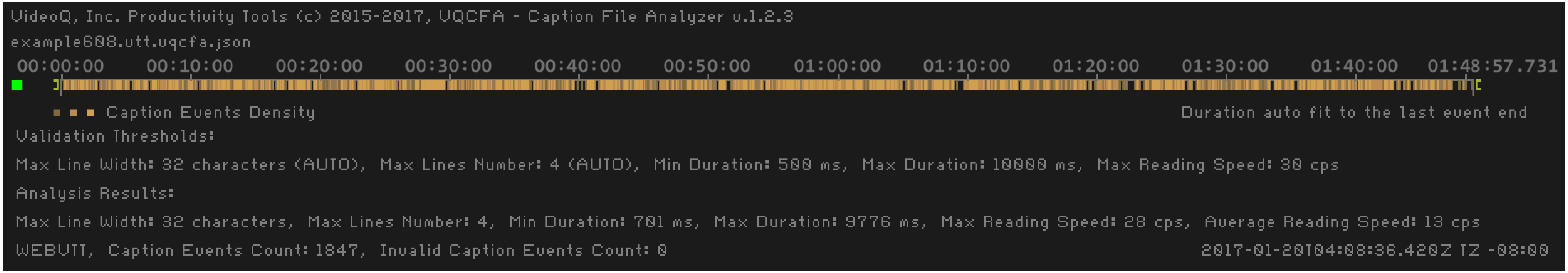
PTOC



```
>  (0) "header": {} (11)
>  (0) "generalInfo": {} (21)
>  (0) "testConditions": {} (14)
>  (0) "eventsStatistics": {} (18)
v  (0) "invalidEvents": {} (3)
       1."invalidEventsCount"              "9"
   v   (1) "invalidEventsByType": {} (9)
           2."lineWidthAboveLimit"         "1"
           2."linesNumberAboveLimit"       "1"
           2."durationBelowLimit"          "1"
           2."durationAboveLimit"          "1"
           2."readingSpeedAboveLimit"      "3"
           2."reversedStartEndTime"        "1"
           2."overlappingTimePositions"    "1"
           2."beyondVideoDuration"         "0"
           2."lateGlobalStartTime"         "0"
   >   (1) "invalidEventsByNumber": [] (9)
v  (0) "captionEventsListing": {} (3)
       1."eventsCount"                     "1393"
       1."duration"                        "02:09:35.617"
   v   (1) "byTimePosition": [] (1393)
           2. 0      "00:00:00.500 --> 00:00:03.000 | VideoQ CC Test No 1 | (c) Copyright 2016. VideoQ, Inc."
           2. 1      "00:00:03.000 --> 00:00:04.750 | Part 1 | Suite of Basic Tests"
           2. 2      "00:00:04.750 --> 00:00:07.500 | suitable for original VTT files | (not for VTT converted from SCC)"
           2. 3      "00:00:07.500 --> 00:00:09.500 | Max Characters per Line: 42 | Max Lines Number: 2"
           2. 4      "00:00:10.000 --> 00:00:11.000 | Line Width Check Tests:"
           2. 5      "00:00:11.000 --> 00:00:13.000 | #1 Valid:  42 characters | #2 Invalid: 43 characters"
           2. 6      "00:00:13.000 --> 00:00:16.000 | Valid Line Width:  42 characters | 0123456789 01 Last Character Position = 42"
           2. 7      "00:00:16.000 --> 00:00:18.500 | Invalid Line Width: 43 characters | 0123456789 012 Last Character Position = 43"
           2. 8      "00:00:19.500 --> 00:00:21.000 | Lines Number Check Tests:"
           2. 9      "00:00:21.000 --> 00:00:23.500 | #1 Valid Number: 2 Lines | #2 Invalid Number: 3 Lines"
           2. 10     "00:00:23.500 --> 00:00:26.000 | Valid: Total = 2 Lines | Line 2: Valid"
```

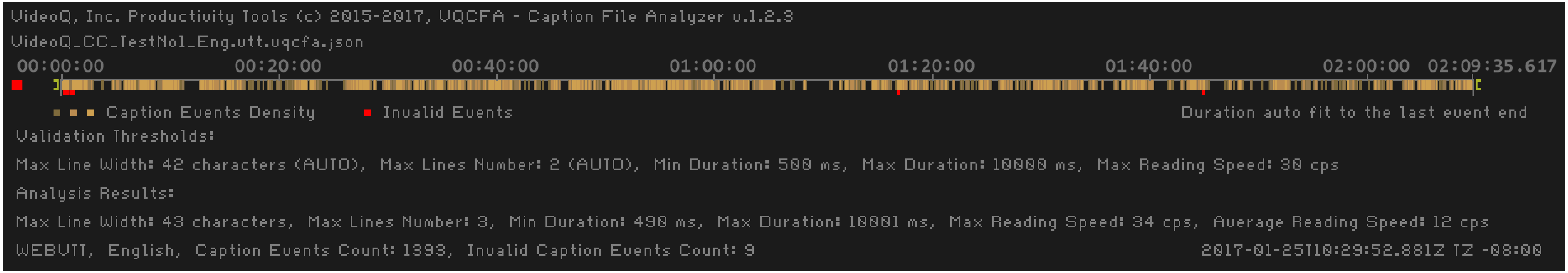
VQCFA – Captions Files Analyzer, Plot Examples

Normal Caption Events – **No problems found**



```
VideoQ, Inc. Productivity Tools (c) 2015-2017, VQCFA - Caption File Analyzer v.1.2.3
example608.vtt.vqcfa.json
00:00:00  00:10:00  00:20:00  00:30:00  00:40:00  00:50:00  01:00:00  01:10:00  01:20:00  01:30:00  01:40:00  01:48:57.731

    ■ ■ ■ Caption Events Density                                         Duration auto fit to the last event end
Validation Thresholds:
Max Line Width: 32 characters (AUTO), Max Lines Number: 4 (AUTO), Min Duration: 500 ms, Max Duration: 10000 ms, Max Reading Speed: 30 cps
Analysis Results:
Max Line Width: 32 characters, Max Lines Number: 4, Min Duration: 701 ms, Max Duration: 9776 ms, Max Reading Speed: 28 cps, Average Reading Speed: 13 cps
WEBVTT, Caption Events Count: 1847, Invalid Caption Events Count: 0                 2017-01-20T04:08:36.420Z TZ -08:00
```

Multiple Caption Events are **Out of Specs:**
Reading Speed, Min Duration, Max Duration,
Overlapping Events, Max Lines Number, Max Chars Per Line



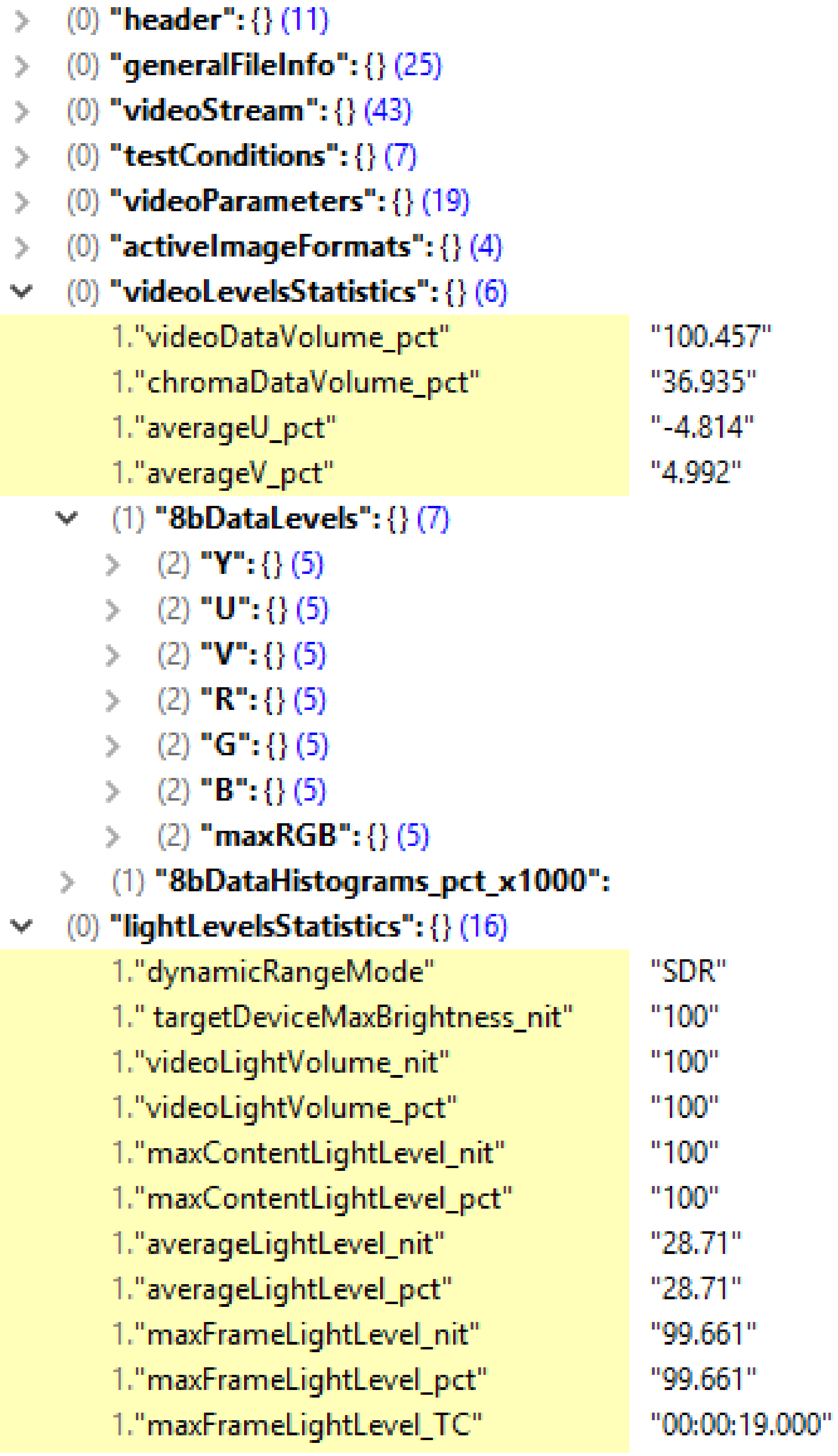
```
VideoQ, Inc. Productivity Tools (c) 2015-2017, VQCFA - Caption File Analyzer v.1.2.3
VideoQ_CC_TestNo1_Eng.vtt.vqcfa.json
00:00:00     00:20:00     00:40:00     01:00:00     01:20:00     01:40:00     02:00:00  02:09:35.617

    ■ ■ ■ Caption Events Density     ■ Invalid Events                    Duration auto fit to the last event end
Validation Thresholds:
Max Line Width: 42 characters (AUTO), Max Lines Number: 2 (AUTO), Min Duration: 500 ms, Max Duration: 10000 ms, Max Reading Speed: 30 cps
Analysis Results:
Max Line Width: 43 characters, Max Lines Number: 3, Min Duration: 490 ms, Max Duration: 10001 ms, Max Reading Speed: 34 cps, Average Reading Speed: 12 cps
WEBVTT, English, Caption Events Count: 1393, Invalid Caption Events Count: 9               2017-01-25T10:29:52.881Z TZ -08:00
```

VQFP – Video Frames Profiler

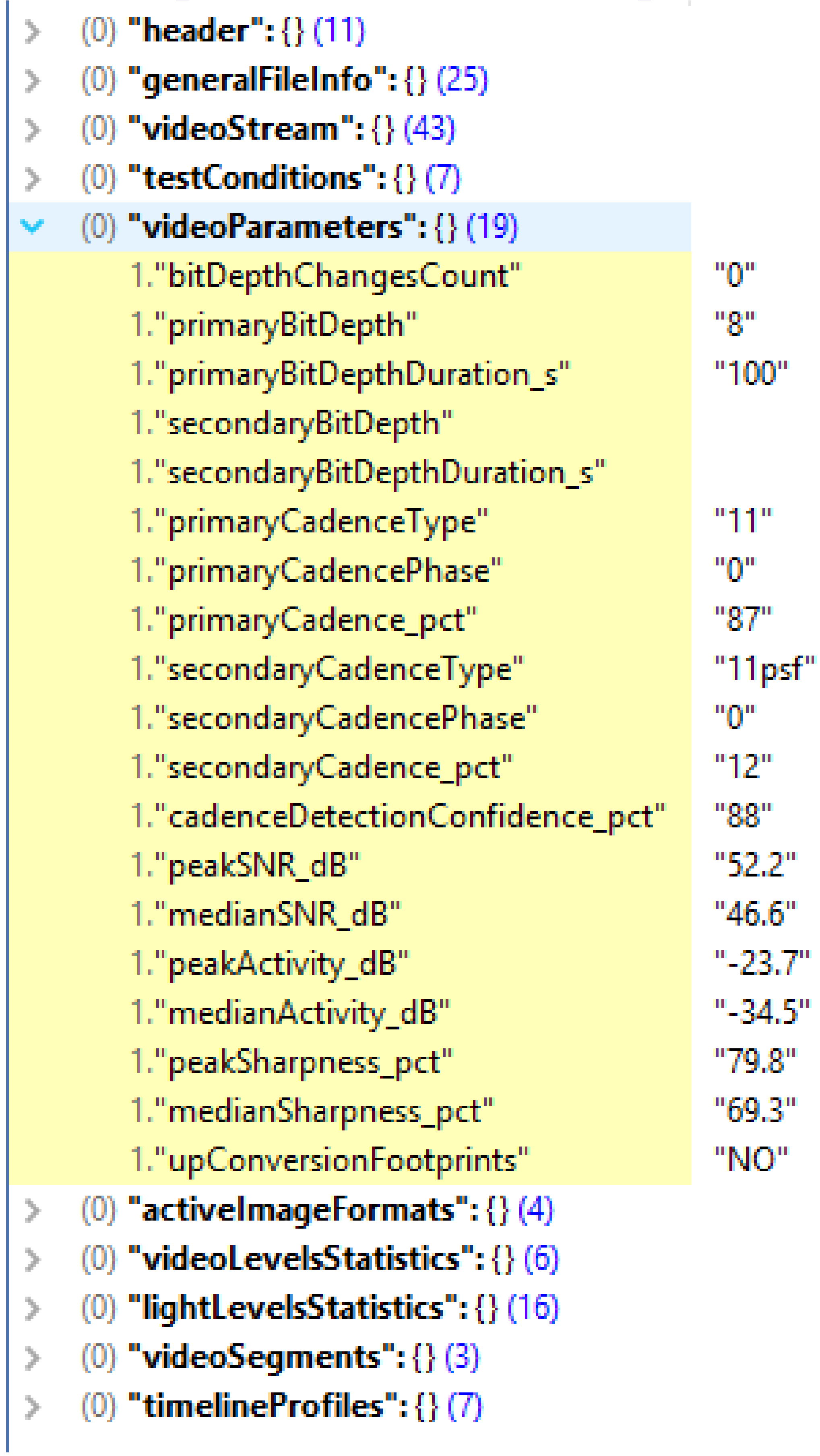
- VQFP reads media file, containing at least one video stream (.MOV, .MP4, etc.)
- It measures video frames parameters and creates machine-readable Report in JSON format showing the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStream**: encoding and format information
  - **testConditions**: user-selected and auto-configured test session parameters
  - **videoParameters**: bit depth variations, frame cadencies, SNR, sharpness, details activity, up-conversion footprints
  - **activeImageFormats**: integrated durations and active image sizes of all detected active image formats (**LetterBox**, **PillarBox**, etc.)
  - **videoLevelsStatistics**: Average and Max values in % and global histograms for Y,U,V,R,G,B and MaxRGB channels
  - **lightLevelsStatistics**: Average and Max values in nits and % of the specified TDMB
  - **videoSegments**, sorted by type and by number (in order of appearance)
  - **timeLineProfiles** of video levels, light level, sharpness and details activity

CLI interface: **vqfp** [-noplot] [-short] [-vsi] [-nr | -fr] [-sdr | -pq | -hlg] -i inFilePath [-o] or [-o outFilePath]

VQFP – Video Frames Profiler, JSON Report Examples



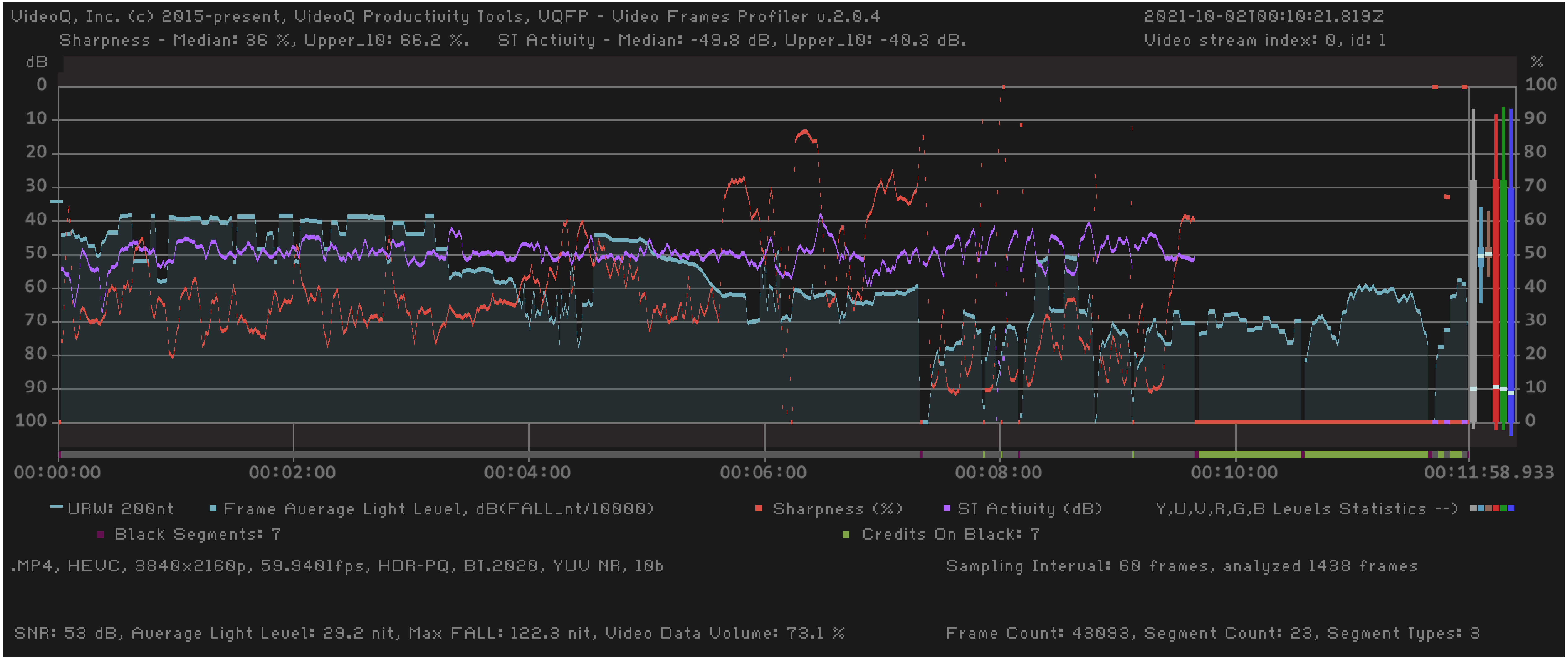
```
›  (0) "header": {} (11)
›  (0) "generalFileInfo": {} (25)
›  (0) "videoStream": {} (43)
›  (0) "testConditions": {} (7)
›  (0) "videoParameters": {} (19)
›  (0) "activeImageFormats": {} (4)
⌄  (0) "videoLevelsStatistics": {} (6)
        1."videoDataVolume_pct"             "100.457"
        1."chromaDataVolume_pct"            "36.935"
        1."averageU_pct"                    "-4.814"
        1."averageV_pct"                    "4.992"
   ⌄  (1) "8bDataLevels": {} (7)
      ›  (2) "Y": {} (5)
      ›  (2) "U": {} (5)
      ›  (2) "V": {} (5)
      ›  (2) "R": {} (5)
      ›  (2) "G": {} (5)
      ›  (2) "B": {} (5)
      ›  (2) "maxRGB": {} (5)
   ›  (1) "8bDataHistograms_pct_x1000":
⌄  (0) "lightLevelsStatistics": {} (16)
        1."dynamicRangeMode"                "SDR"
        1." targetDeviceMaxBrightness_nit"  "100"
        1."videoLightVolume_nit"            "100"
        1."videoLightVolume_pct"            "100"
        1."maxContentLightLevel_nit"        "100"
        1."maxContentLightLevel_pct"        "100"
        1."averageLightLevel_nit"           "28.71"
        1."averageLightLevel_pct"           "28.71"
        1."maxFrameLightLevel_nit"          "99.661"
        1."maxFrameLightLevel_pct"          "99.661"
        1."maxFrameLightLevel_TC"           "00:00:19.000"
```



```
›  (0) "header": {} (11)
›  (0) "generalFileInfo": {} (25)
›  (0) "videoStream": {} (43)
›  (0) "testConditions": {} (7)
⌄  (0) "videoParameters": {} (19)
        1."bitDepthChangesCount"             "0"
        1."primaryBitDepth"                  "8"
        1."primaryBitDepthDuration_s"        "100"
        1."secondaryBitDepth"
        1."secondaryBitDepthDuration_s"
        1."primaryCadenceType"               "11"
        1."primaryCadencePhase"              "0"
        1."primaryCadence_pct"               "87"
        1."secondaryCadenceType"             "11psf"
        1."secondaryCadencePhase"            "0"
        1."secondaryCadence_pct"             "12"
        1."cadenceDetectionConfidence_pct"   "88"
        1."peakSNR_dB"                       "52.2"
        1."medianSNR_dB"                     "46.6"
        1."peakActivity_dB"                  "-23.7"
        1."medianActivity_dB"                "-34.5"
        1."peakSharpness_pct"                "79.8"
        1."medianSharpness_pct"              "69.3"
        1."upConversionFootprints"           "NO"
›  (0) "activeImageFormats": {} (4)
›  (0) "videoLevelsStatistics": {} (6)
›  (0) "lightLevelsStatistics": {} (16)
›  (0) "videoSegments": {} (3)
›  (0) "timelineProfiles": {} (7)
```

VQFP – Video Frames Profiler, Plot Example

Typical HDR10 clip with 2 minutes long Credits On Black at the end
*Relatively **low** spatio-temporal **activity** and **medium** **sharpness***



```
VideoQ, Inc. (c) 2015-present, VideoQ Productivity Tools, VQFP - Video Frames Profiler v.2.0.4          2021-10-02T00:10:21.819Z
    Sharpness - Median: 36 %, Upper_10: 66.2 %.    SI Activity - Median: -49.8 dB, Upper_10: -40.3 dB.   Video stream index: 0, id: 1

.MP4, HEVC, 3840x2160p, 59.9401fps, HDR-PQ, BT.2020, YUV NR, 10b                     Sampling Interval: 60 frames, analyzed 1438 frames

SNR: 53 dB, Average Light Level: 29.2 nit, Max FALL: 122.3 nit, Video Data Volume: 73.1 %     Frame Count: 43093, Segment Count: 23, Segment Types: 3
```

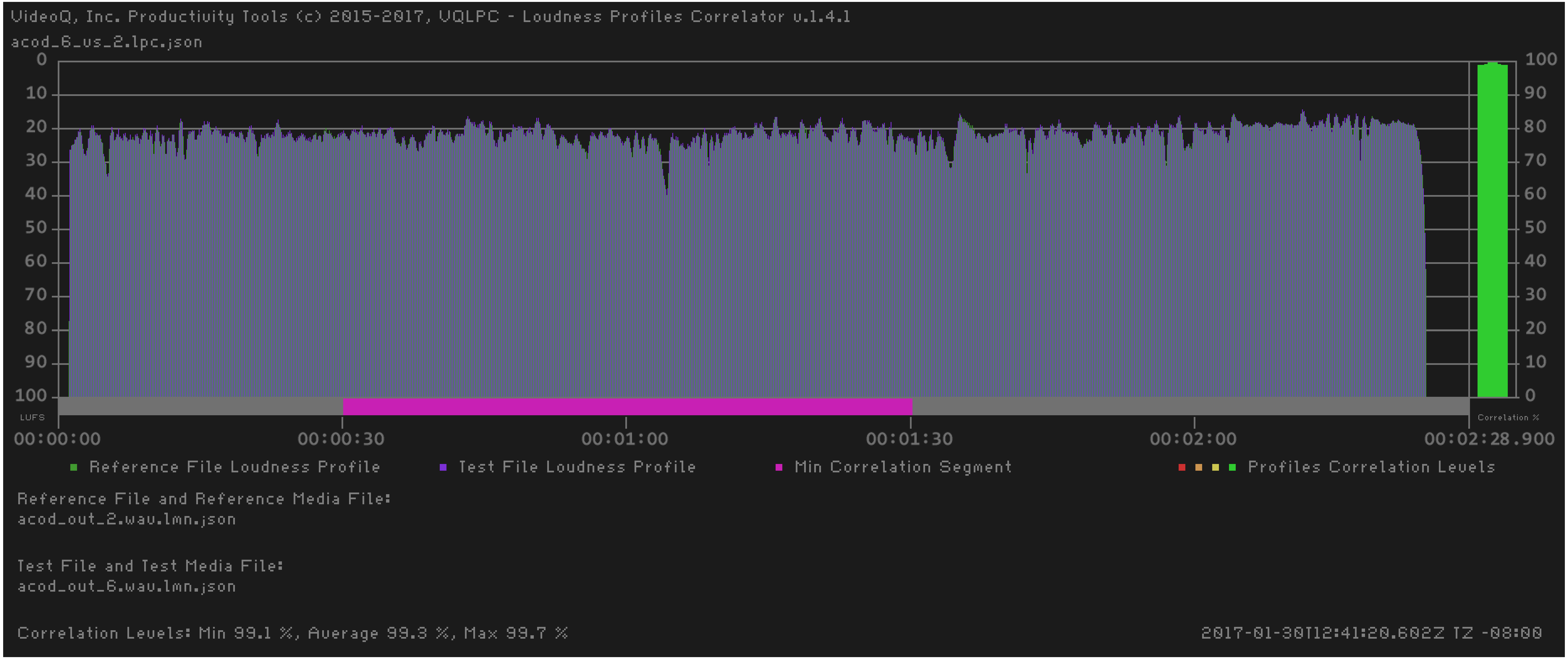
VQLPC – Loudness Profiles Correlator

- VQLPC reads **two** JSON report files created by **VQLPN** module (**Reference** and **Test** inputs)
- It detects and compares **momentary loudness** time-line profiles of two inputs, then creates Report in JSON format
- The most important output parameter is the **correlation value** in %:
    - Value of 100.0 means **perfect match**
    - Values above 99.2 means **small discrepancy**
    - Values below 85.0 mean **significant discrepancy**,
      e.g. caused by two different dialog languages combined with the same noise and music international soundtrack
    - Values below 55 (and down to 0) mean **very strong differences**, *e.g. two different episodes or two different clips*
- Optional output:
    - **PNG file** showing both inputs loudness time-line profiles plots, as well as global correlation parameters BarGraph

CLI interface: **vqlpc** [-p] -i refFilePath testFilePath [-o or -o outFilePath]

VQLPC – Loudness Profiles Correlator, Plot Example 1

Two inputs are two different versions (2.0 and 5.1) of the **same audio track**:
correlation is very high – about 100%



VideoQ, Inc. Productivity Tools (c) 2015-2017, VQLPC - Loudness Profiles Correlator v.1.4.1
acod_6_vs_2.lpc.json

0
10
20
30
40
50
60
70
80
90
100
LUFS
100
90
80
70
60
50
40
30
20
10
0
Correlation %
00:00:00
00:00:30
00:01:00
00:01:30
00:02:00
00:02:28.900
🟩 Reference File Loudness Profile
🟪 Test File Loudness Profile
🟪 Min Correlation Segment
🟥 🟧 🟨 🟩 Profiles Correlation Levels

Reference File and Reference Media File:
acod_out_2.wav.lmn.json

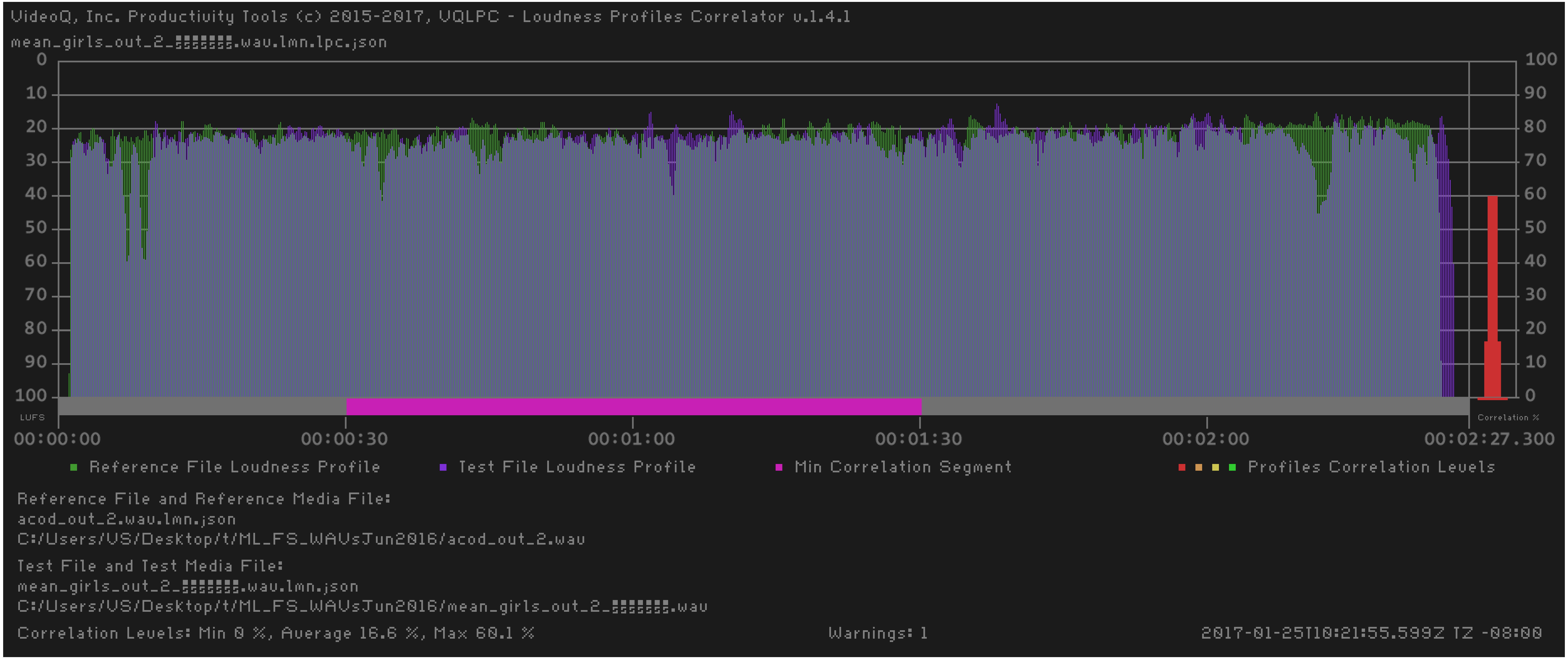
Test File and Test Media File:
acod_out_6.wav.lmn.json

Correlation Levels: Min 99.1 %, Average 99.3 %, Max 99.7 %

2017-01-30T12:41:20.602Z TZ -08:00

VQLPC – Loudness Profiles Correlator, Plot Example 2

Two inputs are in fact **two different audio tracks**
Loudness profiles and durations look similar, but actual **correlation value is very low**



0
10
20
30
40
50
60
70
80
90
100
LUFS
100
90
80
70
60
50
40
30
20
10
0
Correlation %
00:00:00
00:00:30
00:01:00
00:01:30
00:02:00
00:02:27.300
🟩 Reference File Loudness Profile
🟪 Test File Loudness Profile
🟪 Min Correlation Segment
🟥 🟧 🟨 🟩 Profiles Correlation Levels

```
VideoQ, Inc. Productivity Tools (c) 2015-2017, VQLPC - Loudness Profiles Correlator v.1.4.1
mean_girls_out_2_[redacted].wav.lmn.lpc.json

Reference File and Reference Media File:
acod_out_2.wav.lmn.json
C:/Users/VS/Desktop/t/ML_FS_WAVsJun2016/acod_out_2.wav

Test File and Test Media File:
mean_girls_out_2_[redacted].wav.lmn.json
C:/Users/VS/Desktop/t/ML_FS_WAVsJun2016/mean_girls_out_2_[redacted].wav

Correlation Levels: Min 0 %, Average 16.6 %, Max 60.1 %          Warnings: 1          2017-01-25T10:21:55.599Z TZ -08:00
```

VQLPN – Audio Loudness Profiler & Normalizer

- VQLPN reads media file, containing audio stream(s), measures and normalizes audio loudness.
- VQLPN supports MP4, MOV, MXF, WAV, W64, AAC, AC3, EAC3, etc., and various audio stream formats:
  - any bit depth, bit rate and sampling rate, all audio codecs supported by ffmpeg
  - multi-channel and multi-track formats: **1.0, 2.0, 5.1, 7.1**
- It measures the audio stream loudness parameters in accordance with Recommendation ITU-R **BS.1770-4** (*USA* ***ATSC RP A85***, *EBU* ***R128***). Editable *.INI file stores test configuration and target parameters.
- Sorts audio segments by types (**regular audio**, **test tone**, **mute**)
- Finally, VQLPN creates detailed Report in JSON format, including **Momentary Loudness Profile** data array at 100 ms step interval
- Configurable outputs:
  - **Audio file** in the **desired format,** optionally **normalized** to the desired **Integrated Loudness** target
  - **PNG image file** showing **momentary loudness** time-line profile **plot**, **loudness** statistics **bargraph**, **upper levels histogram**, as well as other useful markers and values
- Optional stand-alone utility modules: **VQLPC** *Correlator*, **VQLPP** *Plotter*, **VQILM** *IL Meter*

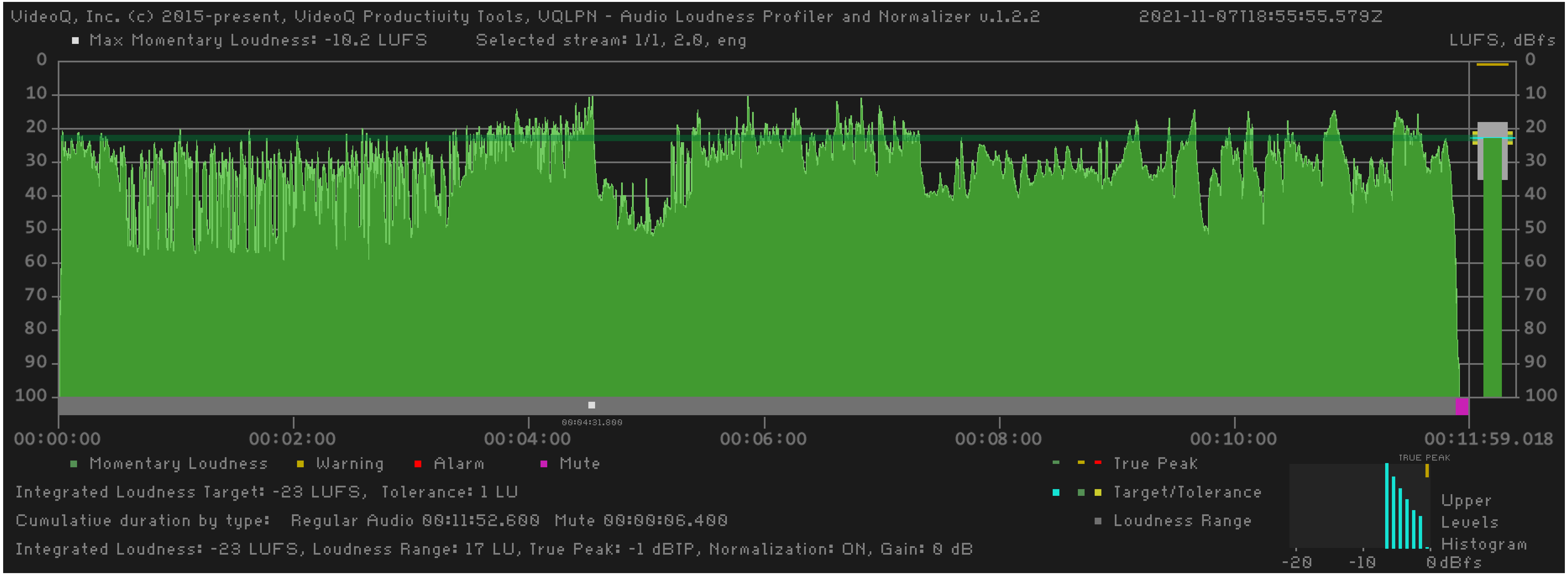
CLI interface: **vqlpn** [-j jsonFilePath] [-c configFilePath] -i inMediaFilePath [-o [outFilePath]]

VQLPN – Loudness Profiler & Normalizer, Plot Example 1

Professional clip with 6 seconds long Mute Fragment at the end

Normalized audio stream – ***Integrated Loudness*** *is exactly equal to* ***-23 LUFS*** *target value*

True Peak *value is* ***quite high***



VideoQ, Inc. (c) 2015-present, VideoQ Productivity Tools, VQLPN - Audio Loudness Profiler and Normalizer v.1.2.2 2021-11-07T18:55:55.579Z

■ Max Momentary Loudness: -10.2 LUFS Selected stream: 1/1, 2.0, eng

Integrated Loudness Target: -23 LUFS, Tolerance: 1 LU

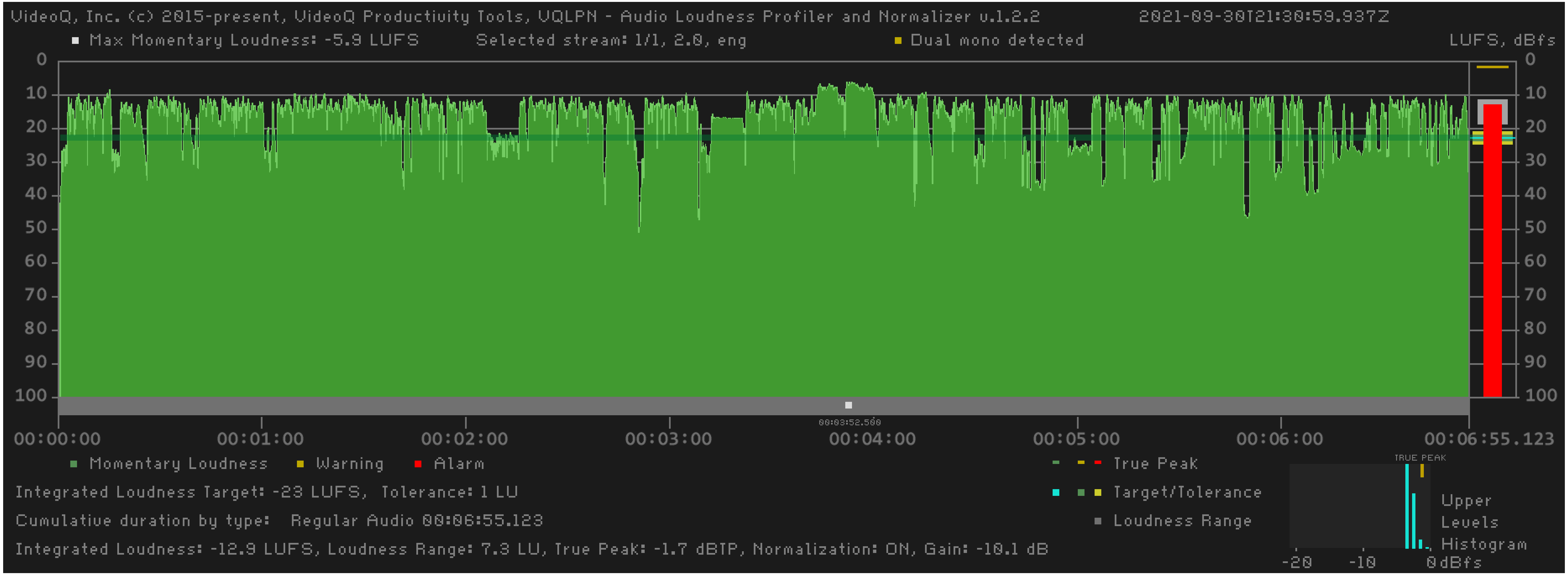
Cumulative duration by type: Regular Audio 00:11:52.600 Mute 00:00:06.400

Integrated Loudness: -23 LUFS, Loudness Range: 17 LU, True Peak: -1 dBTP, Normalization: ON, Gain: 0 dB

VQLPN – Loudness Profiler & Normalizer, Plot Example 2

Legacy content:

Integrated Loudness** is **much higher** than **-23 LUFS** target value, and **True Peak** value is **quite high



```
VideoQ, Inc. (c) 2015-present, VideoQ Productivity Tools, VQLPN - Audio Loudness Profiler and Normalizer v.1.2.2        2021-09-30T21:30:59.937Z
    ■ Max Momentary Loudness: -5.9 LUFS       Selected stream: 1/1, 2.0, eng                  ■ Dual mono detected

Integrated Loudness Target: -23 LUFS,  Tolerance: 1 LU
Cumulative duration by type:   Regular Audio 00:06:55.123
Integrated Loudness: -12.9 LUFS, Loudness Range: 7.3 LU, True Peak: -1.7 dBTP, Normalization: ON, Gain: -10.1 dB
```

VQPLA – Picture Levels Analyzer

- VQPLA reads media file, containing at least one video stream (.MOV, .MP4, etc.)
- It measures video frames parameters and creates machine-readable Report in JSON format showing the following sections:
  - **header**: Report timestamp and program version info
  - **generalFileInfo**: container parameters, including counts of media data streams
  - **videoStream**: encoding and format information
  - **testConditions**: user-selected and auto-configured test session parameters
  - **videoLevelsStatistics**: **Average** and **Max** values in % and **global histograms** for Y,U,V,R,G,B and MaxRGB channels
  - **lightLevelsStatistics**: **Average** and **Max** values in **nits** and **%** of the specified TDMB
  - **videoSegments**, sorted by type and by number (in order of appearance)
  - **timeLineProfiles** of **video levels** and **light level**

CLI interface: **vqpla** [-p] [-tlp] [-ffn] [-si] [-sn] [-r] [-DRMS] [-SRMS] -i inFilePath [-o] or [-o outFilePath]

TOC

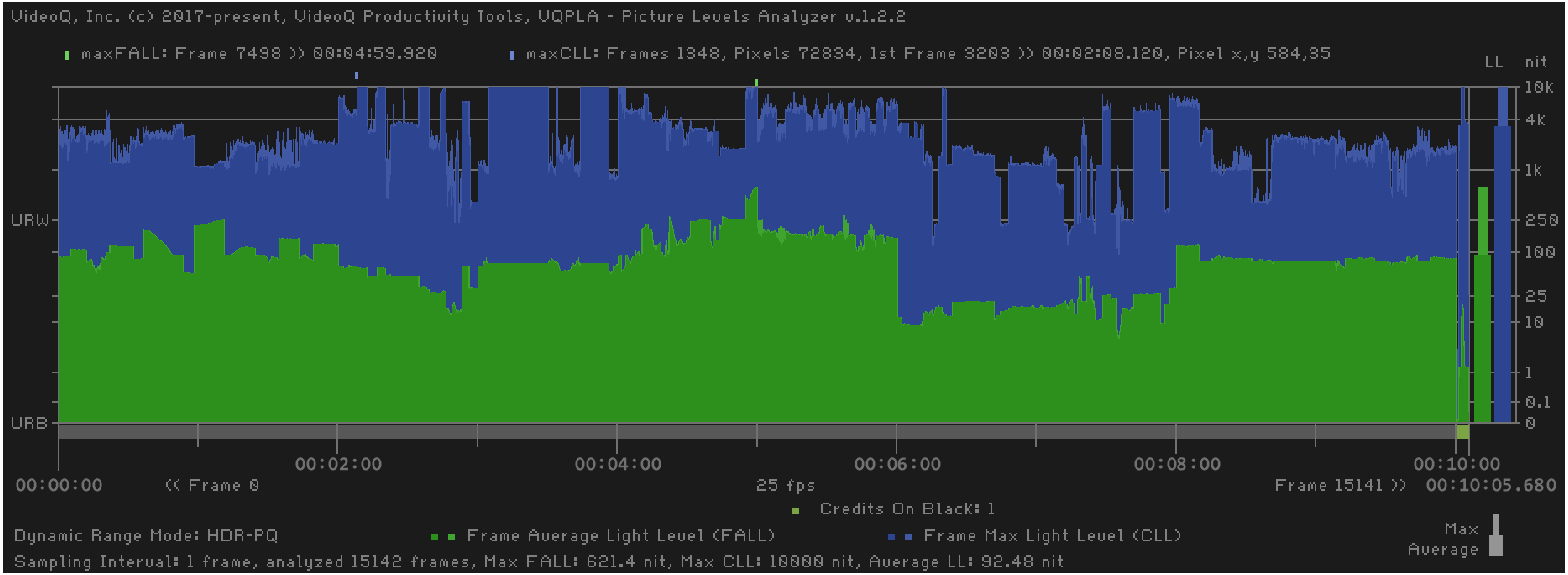
VQPLA – Picture Levels Analyzer, Plot Example

PTOC

HDR-PQ file analyzed. No serious problems found.

Warnings:

*a) Time position ~ 05:00:00 – **Frame Average Light Level** goes **above 600 nit**; this is too much for some displays*
*b) Time position ~ 06:00:00 – Light output **drops from 150 nit down to 10 nit**; this may cause visual discomfort*



VideoQ, Inc. (c) 2017-present, VideoQ Productivity Tools, VQPLA - Picture Levels Analyzer v.1.2.2

🟩 maxFALL: Frame 7498 >> 00:04:59.920

🟦 maxCLL: Frames 1348, Pixels 72834, 1st Frame 3203 >> 00:02:08.120, Pixel x,y 584,35

🟩 Credits On Black: 1

Dynamic Range Mode: HDR-PQ

🟩 🟩 Frame Average Light Level (FALL)

🟦 🟦 Frame Max Light Level (CLL)

Sampling Interval: 1 frame, analyzed 15142 frames, Max FALL: 621.4 nit, Max CLL: 10000 nit, Average LL: 92.48 nit

About VideoQ

Customers & Partners

Company History

- Founded in 2005
- Formed by an Engineering Awards winning team sharing between them decades of global video technology.
- VideoQ is a renown player in calibration and benchmarking of Video Processors, Transcoders and Displays, providing tools and technologies instantly revealing artifacts, problems and deficiencies, thus raising the bar in productivity and video quality experience.
- VideoQ products and services cover all aspects of video processing and quality assurance - from visual picture quality estimation and quality control to fully automated processing, utilizing advanced VideoQ algorithms and robotic video quality analyzers, including latest UHD and HDR developments.

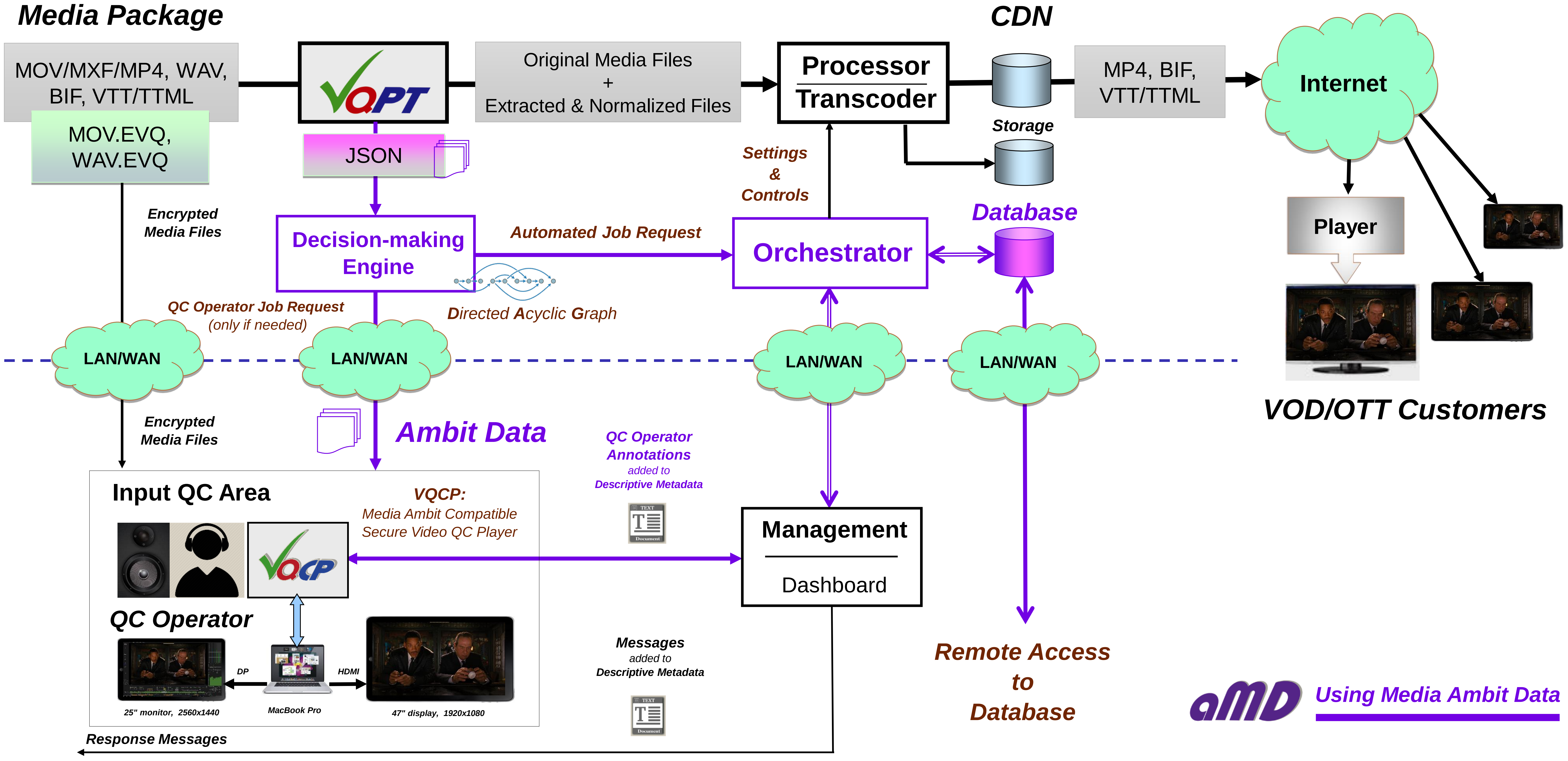
Operations

- Headquarters in CA, USA
- Software developers in Silicon Valley and worldwide
- Distributors and partners in several countries
- Sales & support offices in USA, UK

TOC

Appendix: Data Usage Workflow – VQPT and Media Ambit ™

PTOC



Media Package

MOV/MXF/MP4, WAV, BIF, VTT/TTML

MOV.EVQ, WAV.EVQ

VQPT

JSON

Original Media Files + Extracted & Normalized Files

Processor Transcoder

CDN

Storage

MP4, BIF, VTT/TTML

Internet

Player

VOD/OTT Customers

Encrypted Media Files

Decision-making Engine

Automated Job Request

Settings & Controls

Orchestrator

Database

Directed Acyclic Graph

QC Operator Job Request (only if needed)

LAN/WAN

LAN/WAN

LAN/WAN

LAN/WAN

Encrypted Media Files

Ambit Data

QC Operator Annotations *added to* *Descriptive Metadata*

Input QC Area

VQCP: Media Ambit Compatible Secure Video QC Player

QC Operator

DP

HDMI

25" monitor, 2560x1440

MacBook Pro

47" display, 1920x1080

Management

Dashboard

Messages *added to* *Descriptive Metadata*

Response Messages

Remote Access to Database

Using Media Ambit Data

Copyright VideoQ, Inc. – VQPT Training Presentation