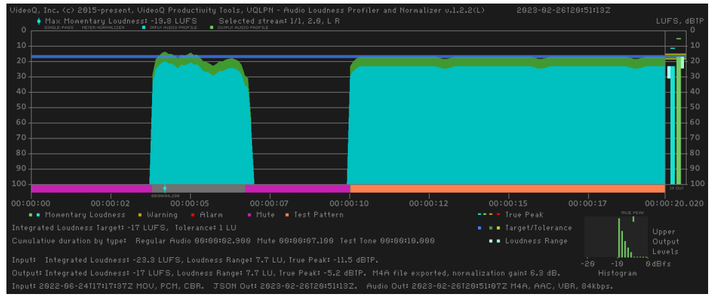


VQLPN

VideoQ Loudness Profiler & Normalizer

Training Presentation

December 2024

VideoQ Productivity Tools

A suite of software modules
for advanced video processing workflow

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

www.videoq.com/vqlpn.html

www.videoq.com

TOC <

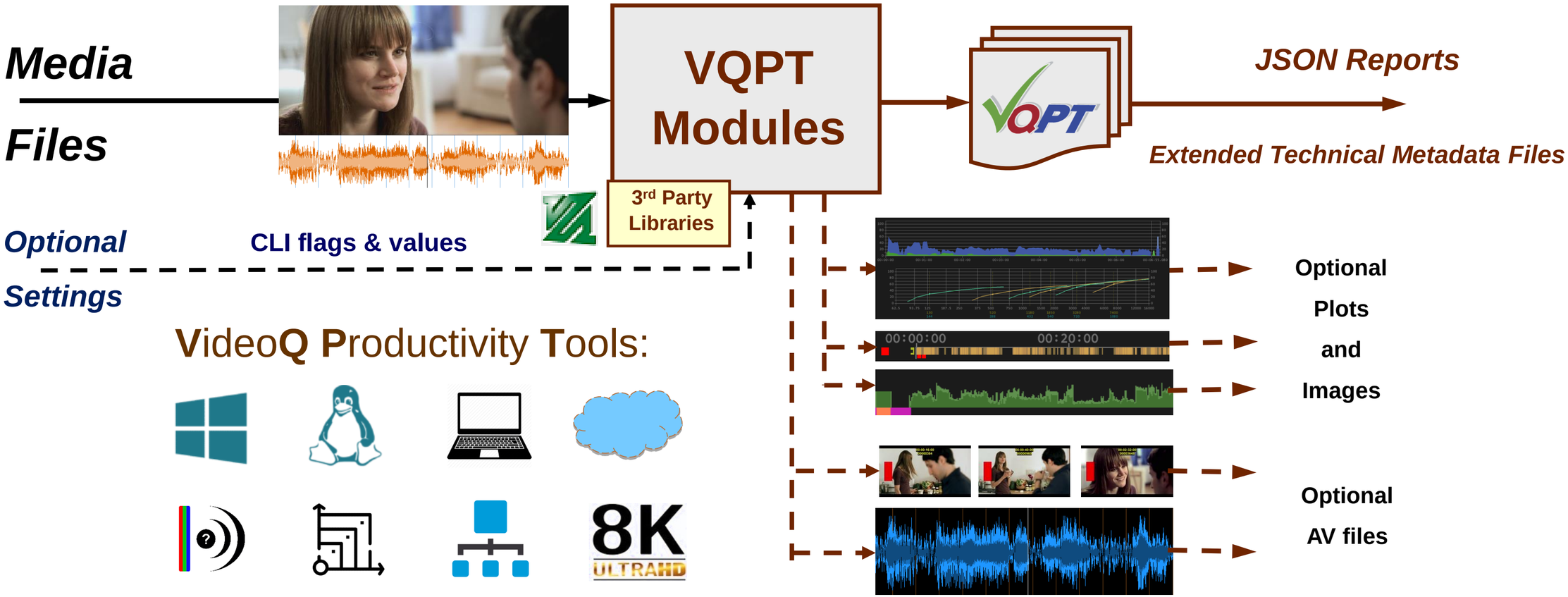
*Click on **TOC** n the upper-left corner of any slide for the **Table Of Contents***

Table of Contents

TOC

VQLPN and VQPT Automated Workflow

Video**Q** **L**oudness **P**rofiler and **N**ormalizer
is one of **VQPT** suite program modules used for AV Content Tests and Processing



TOC

VQLPN Features 1

- VQLPN can be used for production, post-production and distribution applications.
- VQLPN is a portable Windows/Linux CLI program for on premises and cloud computing.

  Portable here means that VQLPN does not require installation of any additional software.
- It measures and normalize the audio stream **loudness parameters** in accordance with Recommendation ITU-R **BS.1770-4** (*USA **ATSC RP A85**, EBU **R128***).
- Editable **.INI** files store test **configurations** and **target** parameters.
- VQLPN detects and sorts **audio segments** by types (**regular audio**, **test tone**, **mute**)
- Finally, VQLPN creates **Report** in machine-readable **JSON** format suitable for **large databases**

TOC

VQLPN Features 2

- VQLPN Report includes input file media info parameters as well as measured **Integrated Loudness**, **True Peak** level estimate, **Clipping Distortions** estimate, and **Loudness Range** values.
- VQLPN Report also includes **Momentary Loudness Profile** data array at 100 ms step interval.
- Optional stand-alone utility modules: **VQLPC** *Correlator*, **VQLPP** *Plotter*, **VQILM** *IL Meter*
- 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

TOC

Input Media File Formats and Parameters

- VQLPN reads media file, containing one or several audio stream(s)
- Input media file extensions:
  - TS, M2TS, MPG, MP4, MKV, MXF, MOV, AVI (with or without video stream)
  - AUDIO, AAC, AC3, EAC3, M4A, MKA, MP3, OGG, WAV, W64 (audio only formats)
- All audio codecs supported by ffmpeg 
- Any duration
- Any bit depth
- Any sampling rate
- Any bitrate
- Multi-channel formats: 1.0, 2.0, 5.1, 7.1
- Multi-track input files with the specified layout are auto-merged to 2.0, 5.1 or 7.1 multi-channel format

TOC

Usage Info Helper

Launching VQLPN executable without any parameters, or with –h flag, brings up the following help message:

Usage (see more in **VQLPN_ReadMe.TXT**):

vqlpn [-j jsonFilePath] [-c configFilePath] [-br bitRate_kbps] -i inFilePath [-o [outFilePath]]

Order of flags and parameters is mandatory and cannot be changed

[**-j jsonFilePath**] option enables **normalization only** mode, skipping the analysis stage

Other user controls and parameters are stored in the **.INI** config file

If [**-c configFilePath**] is omitted, then VQLPN uses **VQLPN.INI** file co-sited with vqlpn executable

If VQLPN.INI file is not found, then it will be auto-created with the default control values

[**-br bitRate_kbps**] option specifies **output audio bitrate** (only for aac, ac3, eac3 codecs)

If [-br bitRate_kbps] is omitted, then output audio bitrate is auto-selected

[**-o outFilePath**] option specifies full **JSON Report Path**: Path\ReportFileName.json

If [-o] is present but outFilePath omitted, outFilePath = inFilePath.vqlpn[options].json

If [-o] is **omitted**, Short Report will be sent **to console** (no JSON/PNG files created)

If Path\FileName contains spaces or special characters use double quotes

Optional **AudioOutFilePath** is auto-generated as **inFilePath.vqlpn[suffix].AudioOutFileExt**

Optional **PlotFilePath** always follows JSON Report outFilePath (*.**png** matching *.**json**)

File names, Report and Log contents are in multi-lingual UTF-8 encoding format

TOC

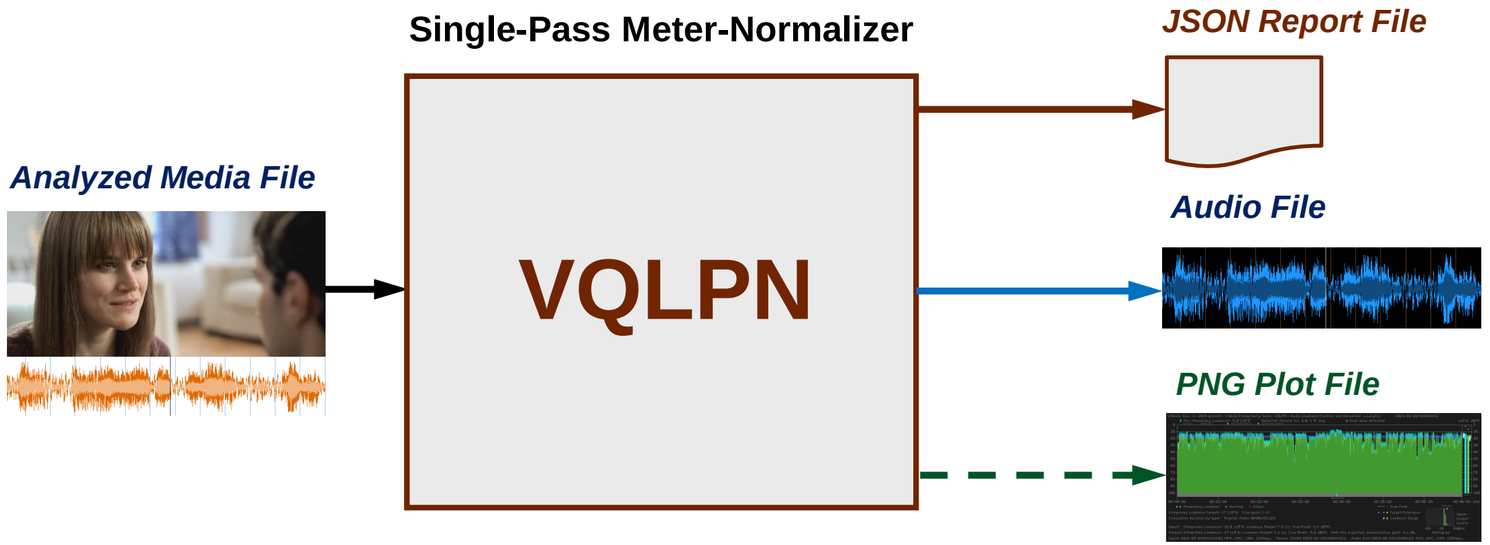
Single-pass Meter-Normalizer Mode

If the selected INI file specifies **TargetLoudness_LUFS,** which is **not equal** to **0**, ***and*** the user-specified Report file name string **does not** ends with _***master.json*** sub-string, then VQLPN works in the ***default*** Meter-Normalizer mode aka “**single-pass run**”.

In this mode VQLPN produces a **JSON Report** file, which describes the original loudness profile of the analyzed media file, **as well as** the loudness normalization process data matching the specified **TargetLoudness_LUFS** value.

In this mode VQLPN ***must*** export an **audio file** with the **normalized** loudness.

Also, VQLPN ***may*** export a **PNG plot file** – if requested in the INI configuration file.



TOC

2-pass Mode For Multi-Target Applications

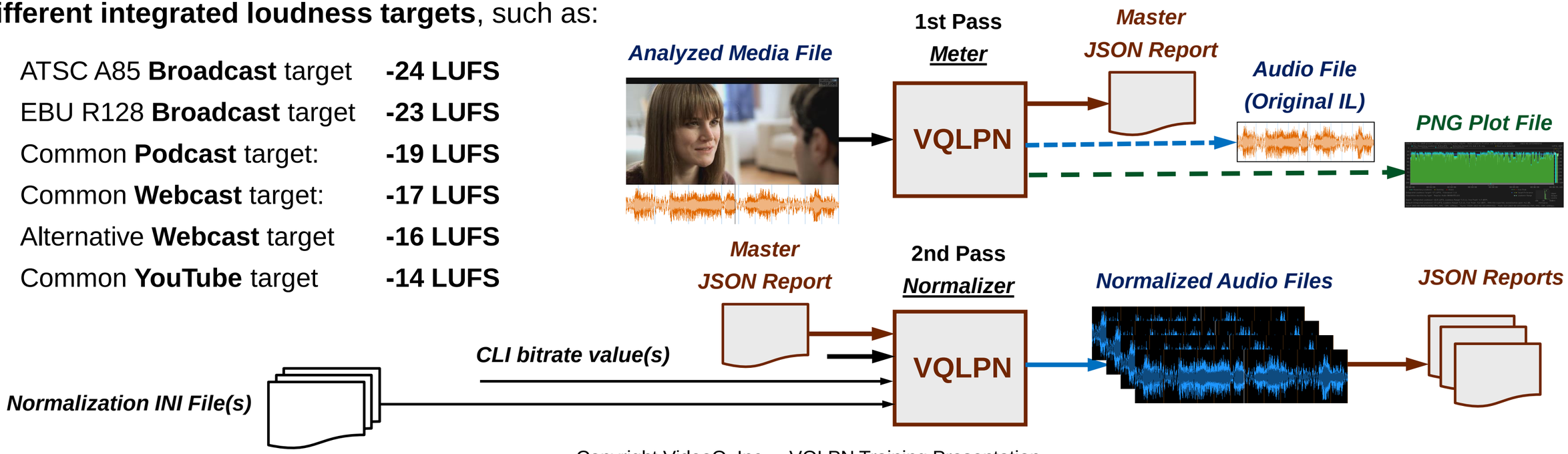
If the selected INI file specifies **TargetLoudness_LUFS=0**, ***or*** the user-specified Report file name string ends with the ***_master.json*** sub-string, then VQLPN works in special "measurement only" aka “**1st pass**” mode.

Important fact is that in this ***time-saving mode*** VQLPN **does not** export any **normalized audio file**. It produces *“master”* **JSON Report**, which describes the measured loudness profile of the analyzed media file, but does not contain any data related to the loudness normalization process.

Such JSON Report can be used in several 2nd pass runs, when the **[-j jsonFilePath]** CLI option enables "normalization only" aka “**2nd pass**” mode, skipping the analysis stage and specifying **one** or **several targets** for the **integrated loudness**.

The 2-pass mode provides for fast writing of **multiple** output **audio files** with different **formats/bitrates** and/or normalized to **different integrated loudness targets**, such as:

- ATSC A85 **Broadcast** target **-24 LUFS**
- EBU R128 **Broadcast** target **-23 LUFS**
- Common **Podcast** target: **-19 LUFS**
- Common **Webcast** target: **-17 LUFS**
- Alternative **Webcast** target **-16 LUFS**
- Common **YouTube** target **-14 LUFS**



TOC

VQLPN Configuration File Structure

```
;VideoQ VQLPN.INI file created 2022-05-23T23:10:27.937Z
;User can edit or replace this file as needed, add your note here:
;
[ConfiguredBy]
ConfiguredBy=Victor Steinberg
[AudioStreamIndex]
AudioStreamIndex=-1
[TargetLoudness_LUFS]
TargetLoudness_LUFS=-17
[LoudnessTolerance_LU]
LoudnessTolerance_LU=1
[MuteDurationThreshold_S]
MuteDurationThreshold_S=1
[AudioFileOut]
AudioFileOut=YES
[AudioFileExtension]
AudioFileExtension=MP4
[AudioFileCodec]
AudioFileCodec=AUTO
[TimelineProfile]
TimelineProfile=YES
[PlotFileOut]
PlotFileOut=YES
```

TOC

VQLPN Configuration File Usage 1

- Default VQLPN.INI config file is **co-sited** with the **vqlpn executable**.
- User can edit this config file **content** as needed, but “VQLPN.INI” file name must be preserved.
  Alternatively use “ -c ” CLI flag followed by full path to any custom *.INI config file name.
  It is advisable to create custom versions for particular tasks, e.g. “VQLPN_CFG1.INI”, and store the backup copies.
- **ConfiguredBy=** value is optional, empty string value after "ConfiguredBy=" is allowed.
  *However, it is recommended to use the **real person/company name**, this is especially important for large databases.*
  *It is also recommended to update the ISO formatted **UTC timestamp** of INI file creation within the topmost text line.*
- **AudioStreamIndex=** option specifies the selected Audio Stream Index (0, 1, etc.)
  AudioStreamIndex=-1 or empty string value means safe default, typically it means AudioStreamIndex=0
  If AudioStreamIndex=**-1**, then **multi-track** file audio streams will be **auto-merged** to the appropriate 2.0, 5.1 or 7.1 format
- **TargetLoudness_LUFS=** and **LoudnessTolerance_LU=** parameters are very important:
  - Default (**EBU R128**) values: TargetLoudness_LUFS=**-23**, LoudnessTolerance_LU=**1** (it means +/- 1 LU)
  - Other standards and custom applications values are also supported.
    *E.g. to comply with **ATSC A/85** standard user should specify in the config *.INI file:*
    *TargetLoudness_LUFS=**-24** and LoudnessTolerance_LU=**2** (it means +/- 2 LU).*

TOC

VQLPN Configuration File Usage 2

- **MuteDurationThreshold_S=** option specifies the Mute Segment Detection Duration Threshold,

  i.e. the detectable mute segment minimum duration value in integer seconds; default = 1 (one second)

- **AudioFileOut=**YES enables audio file export; *see next slides for details*
- **AudioFileExtension=** and **AudioFileCodec=** options allows specification of output audio file format;

  default string values: AUTO, see next slides for details

- **OutFileNormalization=**YES enables normalization of output file audio content **Integrated Loudness** to the specified TargetLoudness_LUFS value;

  user can set OutFileNormalization=NO to export the selected audio stream with the original loudness

- **TimelineProfile=**YES option enables Timeline Profile section within JSON Report file;

  user can set TimelineProfile=NO to reduce Report file size (not recommended)

- **PlotFileOut=**YES option enables Loudness Timeline Profile Plot PNG File output;

  user can set PlotFileOut=NO to save storage space (not recommended)

TOC

VQLPN Modes and Special File Names

You can configure the VQLPN for a special "1st pass" aka "Meter only" mode in two ways:

1. The first way is to use an **output JSON file name** containing a special "_**master**" substring, e.g. "Test.mp4.vqlpn_**master**.json"
2. Alternatively, you can use **any** output JSON file name, but specify **TargetLoudness_LUFS=0** within the VQLPN.INI file

When VQLPN **output audio file** is generated in **normalization** mode, the auto-generated output file name will contain a **special substring** ".vqlpn[suffix].EXT" at the end.

E.g., for input media file "Test.mp4", **-17** LUFS target level, aac codec, 192kbps bitrate, and AudioFileExtension=**MP4**, the output file name will be "Test.mp4.**vqlpn-17_aac_192.mp4**".

In "**Meter**" mode normalization is OFF, and an audio file output is **optional**, the output audio loudness is the same as input (not modified), and the auto-generated name suffix is reduced to ".**vqlpn.EXT**", e.g., "Test.mp4.**vqlpn.mp4**".

TOC

Setting Output Audio File Format 1

- There are two different ways to specify the output audio file format and parameters:
  1. **AudioFileCodec=AUTO** and **AudioFileExtension=AUTO** means that **output file codec** is set **equal** to the **input file codec**, and this selection also auto-defines the associated output file **extension**.
  2. Any **other** string value, e.g. **AudioFileCodec=AAC**, means explicit **manual selection** of output audio **codec**.
  3. In any case output audio file path is auto-generated as **inFilePath.vqlpn[suffix].AudioFileExt**,
     e.g. if input file path = c:\test\test.mp4 and automatic or manual extension selection = MP4,
     then the output audio file full path can be c:\test\test.mp4.vqlpn-17_aac_192.mp4
     Note that, If the duration of the resulting WAV file and the number of audio channels in it require the output file size greater than 4 GB, then VQLPN automatically switches the output format and extension to W64.
- Input sampling rate, language tag and channels layout metadata are preserved in the output;
  but in case of pcm codec the default sampling rate = 48kHz and default bit depth = 24bit are used.

TOC

Setting Output Audio File Format 2

Output file **codec** can be explicitly set in the .INI file, e.g. **AudioFileCodec**=**AAC**.

AudioFileCodec=**AUTO** means that **output file codec** is **auto-selected** as the detected **input file codec**.

In both cases **AudioFileExtension**=**AUTO** means that the container is auto-defined by the selected codec as shown in the table below:

Input and Output Codec	Extension (container)
aac	MP4(default)
ac3	MP4
eac3	MP4
flac	OGG
pcm	WAV *(W64 for files larger than 4GB)*

*If an **input file codec** is **not supported** by VQLPN, then the **output file codec** and **extension** are set to the **default** format: **AAC** codec and **MP4** extension*

TOC

Setting Output Audio File Format 3

Output audio file **container** can be **explicitly** set in the .INI file, e.g. **AudioFileExtension**=**MP4**,

or it can be set **automatically**: **AudioFileExtension**=**AUTO**.

The table below shows the supported extensions and codecs associated with them in **AudioFileCodec**=**AUTO** mode:

Extension (container)	Output Codec
MP4	aac (default)
M4A	aac
WAV	pcm
OGG	flac
MKA	flac
AC3	ac3
EAC3	eac3
TS	eac3

*If an unsupported **codec + container** combination is requested (e.g. AAC + WAV), then VQLPN will **exit** and the corresponding **error message** will be sent to console.*

TOC

Output Audio File Name

Audio Out File Path is auto-generated as inFilePath.vqlpn[suffix].AudioFileExt.

Suffix substring contains the selected normalization target level, codec, and (optionally) bitrate.

If the selected codec is **aac**, or **ac3**, or **eac3**, and user specified the output bitrate in **kbps** after **-br** flag, then the output file name suffix is extended to include this bitrate.

E.g., if input file path = c:\test\test.mp4 and automatic or manual output **extension** selection = MP4, loudness normalization **target** = -17 LUFS, **codec** = AAC, and **bitrate** = 192kbps, then the output audio file full path is c:\test\test.mp4.vqlpn**-17_aac_192**.mp4

In case of **AAC** codec, to preserve the **CBR mode**, VQLPN automatically **limits actual bitrate** kbps value:

AudioChannelsNumber	AAC_MaxAudioBitRate
1	96
2	192
6	384
8	512

TOC

Loudness Range, True Peak, and Upper Levels Histogram

Both VQLPN JSON Report and optional PNG plot image, in addition to the **most important parameter** – **Integrated Loudness** (aka **I** or **IL**), provide the measured values related to other important parameters.

*A lot of the time, when we talk about "loudness" and "loudness normalization" we mean **integrated loudness**, effectively **ignoring** other parameters*.

It should be noted that:

- **Loudness Range** (aka **LRA**) parameter serves mainly for information; it does describe the dynamics of the audio content along the timeline, but typically LRA is **not specified** in any **compliance** test procedures
- True Peak (aka **TP**) value is typically **specified** in formal **compliance** procedures, as a ceiling (max) limit, e.g., -2dBTP.
  But, in practice, TP issues are more complicated. The negative consequences of high TP values depend on (***not specified***) **frequency of occurrence** and **distribution** of these unwanted events along the timeline. If along the audio track there are hundreds of such events and they are not **concentrated** in one relatively short fragment, then we may face serious **Clipping Distortions** issues.
  On the other hand, even formally compliant TP values can be dangerous in combination with low IL value, e.g., TP = -2dBTP and IL = -28LUFS. After the mandatory IL normalization to -23LUFS (implying audio gain = +5dB), TP value will jump to unacceptable level of +3dBTP!
- **Upper Levels Histogram** helps with the expected level of **Clipping Distortions** estimation.
  Gradual decrease of the histogram **bin counts** typically indicate good quality of non-clipped voice and music waveforms.
  High numbers of hits in the upper levels bins, especially the levels **close to TP level**, are symptoms of unusual distribution, most probably due to the audio signal **clipping**. The same applies to sharp drop in counts at some audio level, as opposite to regular gradual decrease.

Important: VQLPN does not ***measure*** clipping distortions or the related subjective quality loss, it only provides an ***estimate***, which can be used as a ***hint*** that with the measured ***statistics*** such distortions are ***possible***.

TOC

JSON Report Example, Part 1

- (0) **"header"**: {} (27)
  - 1."operatingSystem": "Linux"
  - 1."programShortName": "VQLPN"
  - 1."programName": "Audio Loudness Profiler and Normalizer"
  - 1."version": "1.2.2"
  - 1."copyright": "VideoQ, Inc. (c) 2015-present"
  - 1."license": "Trial Demo version, expires 2023-01-15"
  - 1."configuredBy": "Victor Steinberg"
  - 1."reportDateTimeUTC": "2022-12-29T18:36:11.275Z"
  - 1."reportDateTimeLocal": "2022-12-29T18:36:11.275"
  - 1."localTimeZone": "UTC+00:00, GMT"
  - 1."elapsedTime_ms": "35315"
  - 1."elapsedTime_TC1000": "00:00:35.315"
  - 1."applicationFolder": "/home/steinv/vqtest"
  - 1."launchedFromFolder": "/home/steinv/vqtest"
  - 1."jsonInputFileName": "na"
  - 1."inputConfigFileName": "VQLPN.INI"
  - 1."inputMediaFileName": "/mnt/l/Mexicana.mp4"
  - 1."reportFileNameMode": "Specified in command line"
  - 1."reportFileName": "/mnt/l/Mexicana.mp4.vqlpn.json"
  - 1."plotFileName": "/mnt/l/Mexicana.mp4.vqlpn.png"
  - 1."outputAudioFile": "Yes"
  - 1."outputAudioFileName": "/mnt/l/Mexicana.mp4.vqlpn-17.m4a"
  - 1."outputAudioNormalization": "Yes"
  - 1."outputAudioCodec": "aac"
  - 1."outputAudioBitRateControl": "Specified in command line"
  - 1."specifiedAudioBitRate_kbps": "256"
  - 1."selectedAudioBitRate_kbps": "192"
- (0) **"inputMediaFileInfo"**: {} (2)
- (0) **"testConditions"**: {} (9)
- (0) **"outputAudioFileInfo"**: {} (2)
- (0) **"inputMediaFileLoudness"**: {} (15)
- (0) **"outputAudioFileLoudness"**: {} (15)
- (0) **"audioSegments"**: {} (3)
- (0) **"timeLineProfile"**: {} (3)

- (0) **"header"**: {} (24)
- (0) **"inputMediaFileInfo"**: {} (2)
  - (1) **"generalFileInfo"**: {} (25)
  - (1) **"audioStreams"**: {} (2)
    - 2."count": "1"
    - (2) **"1"**: {} (22)
      - 3."audioStreamIndex": "0"
      - 3."streamID": "2"
      - 3."title": "na"
      - 3."encodedDateTimeUTC": "na"
      - 3."taggedDateTimeUTC": "na"
      - 3."languageISO639Alpha2": "en"
      - 3."languageISO639Alpha3": "eng"
      - 3."channelsNumber": "2"
      - 3."channelLayout": "L R"
      - 3."duration_ms": "415123"
      - 3."duration_TC1000": "00:06:55.123"
      - 3."startDelay_TC1000": "na"
      - 3."samplingRate": "48000"
      - 3."bitsPerComponent": "na"
      - 3."samplesCount": "19925904"
      - 3."framesCount": "19459"
      - 3."bitRate_bps": "128000"
      - 3."bitRateMode": "CBR"
      - 3."streamSize_byte": "6642006"
      - 3."codecID": "mp4a-40-2"
      - 3."encodingFormat": "AAC"
      - 3."encodingProfile": "na"
- (0) **"testConditions"**: {} (8)
- (0) **"outputAudioFileInfo"**: {} (2)
- (0) **"inputMediaFileLoudness"**: {} (15)
- (0) **"outputAudioFileLoudness"**: {} (15)
- (0) **"audioSegments"**: {} (3)
- (0) **"timeLineProfile"**: {} (3)

TOC

JSON Report Example, Part 2

- (0) **"header"**: {} (24)
- (0) **"inputMediaFileInfo"**: {} (2)
- (0) **"testConditions"**: {} (8)
- (0) **"outputAudioFileInfo"**: {} (2)
- (0) **"inputMediaFileLoudness"**: {} (15)
  - 1."integratedLoudness_LUFS" "-12.9"
  - 1."targetLoudness_LUFS" "-17"
  - 1."integratedLoudness_LU" "4.1"
  - 1."normalizationGain_dB" "-4.1"
  - 1."truePeak_dBTP" " -1.7"
  - 1."clippingDistortionsLevel" "Undetectable"
  - 1."maxMomentaryLoudness_LUFS" " -5.9"
  - 1."maxMomentaryLoudness_ms" "232500"
  - 1."maxMomentaryLoudness_TC1000" "00:03:52.500"
  - 1."loudnessRange_LU" "7.3"
  - 1."loudnessRangeMin_LUFS" "-18.8"
  - 1."loudnessRangeMax_LUFS" "-11.5"
  - 1."dualMono" "Yes"
  - 1."stereoPhaseInversion" "No"
  - (1) **"upperLevelsHistogram"**: {} (2)
    - 2."binsCount" "4"
    - (2) **"values"**: [] (4)
      - (3) **0**: {} (1)
        - 4."0dBfs" "0"
      - (3) **1**: {} (1)
        - 4."-1dBfs" "338"
      - (3) **2**: {} (1)
        - 4."-2dBfs" "14140"
      - (3) **3**: {} (1)
        - 4."-3dBfs" "33922"
- (0) **"outputAudioFileLoudness"**: {} (15)
- (0) **"audioSegments"**: {} (3)
- (0) **"timeLineProfile"**: {} (3)

- (0) **"header"**: {} (24)
- (0) **"inputMediaFileInfo"**: {} (2)
- (0) **"testConditions"**: {} (8)
- (0) **"outputAudioFileInfo"**: {} (2)
- (0) **"inputMediaFileLoudness"**: {} (15)
- (0) **"outputAudioFileLoudness"**: {} (15)
- (0) **"audioSegments"**: {} (3)
  - 1."totalSegmentsCount" "1"
  - (1) **"segmentsByType"**: {} (5)
    - (2) **"RegularAudio"**: {} (3)
      - 3."count" "1"
      - 3."duration_ms" "415123"
      - 3."duration_TC1000" "00:06:55.123"
    - (2) **"Mute"**: {} (3)
    - (2) **"TestTone"**: {} (3)
    - (2) **"VQAUD1"**: {} (3)
    - (2) **"VQAUD2"**: {} (3)
  - (1) **"segmentsByNumber"**: {} (1)
    - (2) **"1"**: {} (7)
      - 3."type" "RegularAudio"
      - 3."duration_ms" "415123"
      - 3."duration_TC1000" "00:06:55.123"
      - 3."start_ms" "0"
      - 3."end_ms" "415123"
      - 3."start_TC1000" "00:00:00.000"
      - 3."end_TC1000" "00:06:55.123"
- (0) **"timeLineProfile"**: {} (3)
  - 1."meterMode" "EBU_R128_MomentaryLoudness"
  - 1."timeStep_ms" "100"
  - (1) **"momentaryLoudnessLUFS_x10"**:
    - 2. 0 -1000
    - 2. 1 -1000
    - 2. 2 -1000
    - 2. 3 -420

TOC

Plot Example 1 – Meter Mode

- **Measured Integrated Loudness:** IL = -12.9 LUFS. *For 1st pass (Meter Mode) the Target IL = 0, i.e. not specified.*
- **Measured True Peak** value: **-1.7 dBTP**
- **Created Master JSON** file can be used for several fast 2nd pass processors.

***Conclusion**: This clip can be normalized for all broadcasting/webcasting targets without any significant issues.*

Note the Cyan color of the audio profile plot, single IL bar, and absence of target markers

Small font messages designate 1st pass (Meter Mode)

TOC

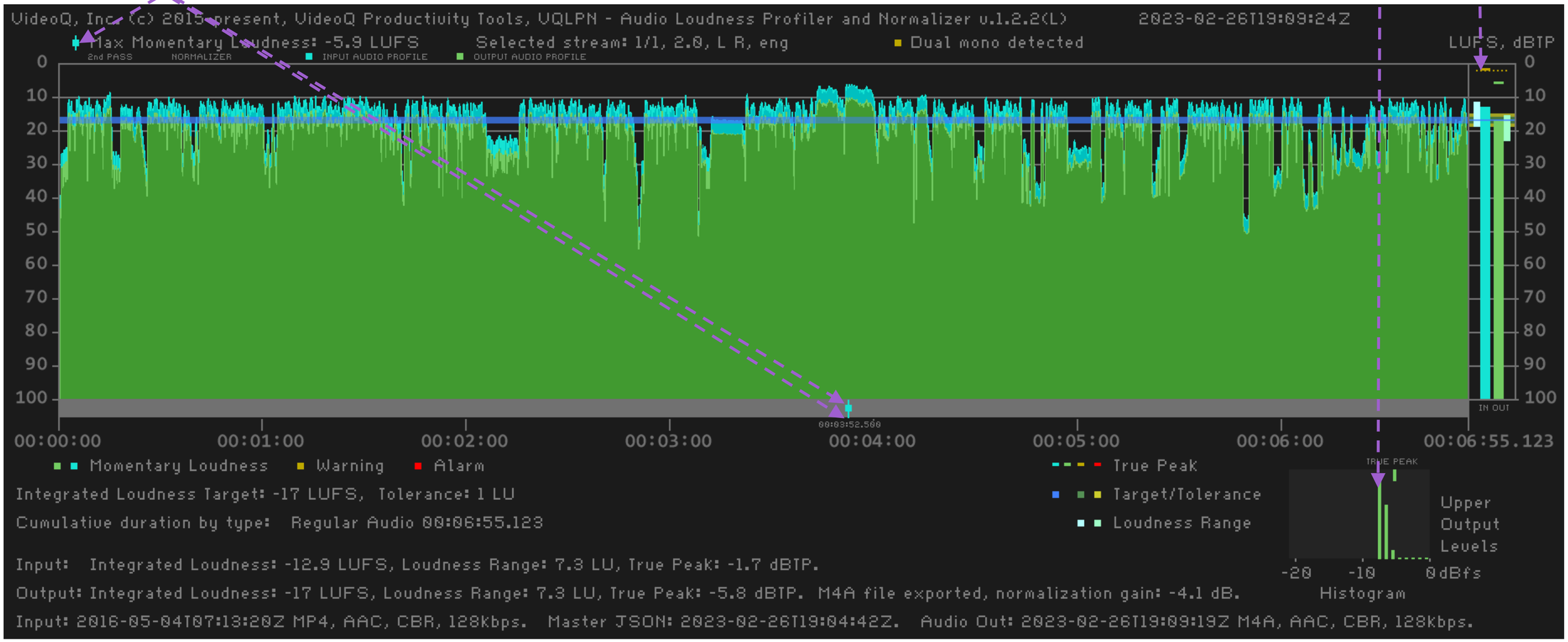
Plot Example 2 – Normalizer Mode

- **Integrated Loudness** is much higher than specified **-17 LUFS** target value. IL = **-12.9 LUFS**, probably legacy content
- **True Peak** value is **relatively high: -1.7 dBTP**
- **Dual Mono** (L=R) stereo content detected, probably upconverted from mono original
- **Output Upper Levels Histogram** shows no **Clipping Distortions**.

Input True Peak** level is **relatively high** but **Output Upper Levels Histogram** shows **no distortions

***Conclusion**: The clip is not suitable for broadcasting/webcasting, but can be normalized without any significant issues.*

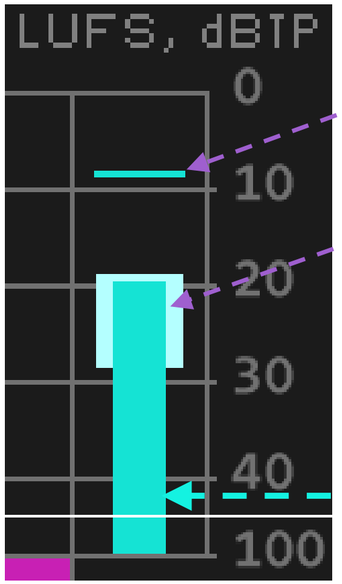
Small Cyan markers for Input Momentary Loudness max value and the corresponding time position



TOC

VQLPN Plot Image Details Explained 1

Meter Mode
(1st Pass Mode)



True Peak Level Marker(s)

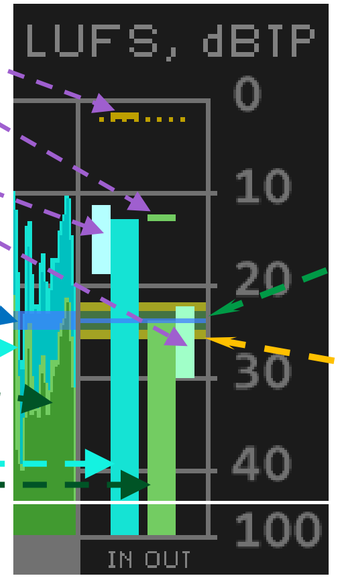
Loudness Range Zone(s)

Integrated Loudness Target

Momentary Loudness Profiles

Integrated Loudness Bar(s):

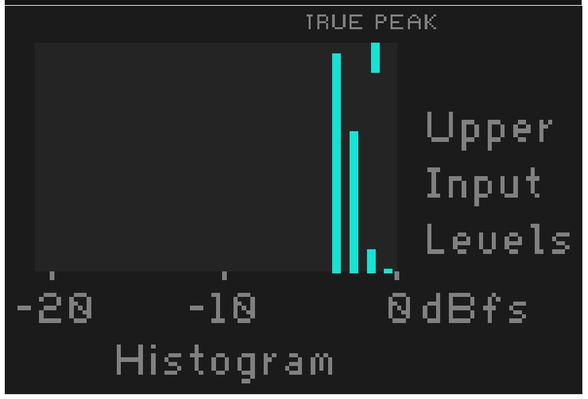
Meter-Normalizer & Normalizer Modes
(Single Pass or 2nd Pass)



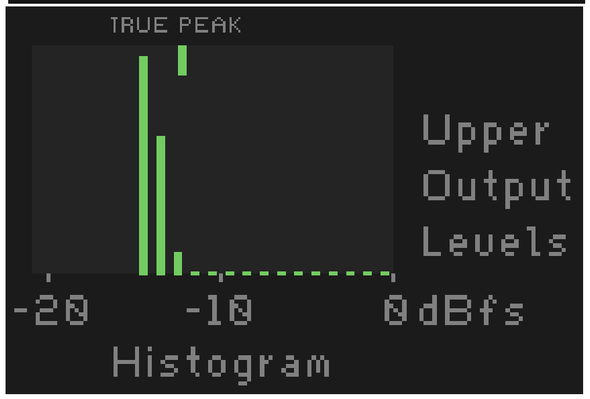
Tolerance Zone x1

Tolerance Zone x2

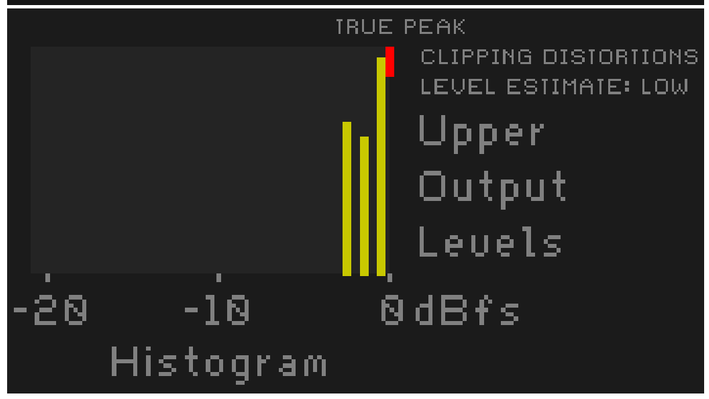
Good Upper Levels Histogram



Good Upper Levels Histogram

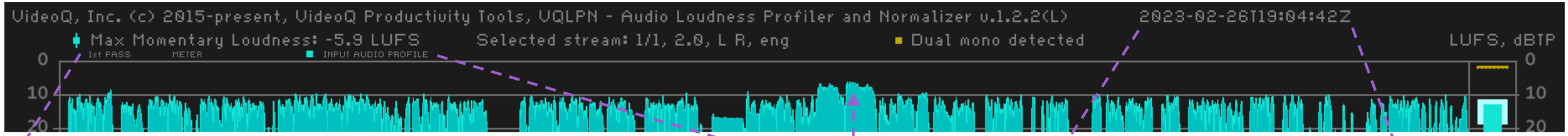


Upper Levels Histogram Warning



TOC

VQLPN Plot Image Details Explained 2



1st PASS METER INPUT AUDIO PROFILE

2023-02-10T23:23:24Z

Auto-generated info messages: VQLPN **Mode** and **Audio Profile Type**

VQLPN PNG creation UTC **Time Stamp**

Momentary Loudness Timeline Profiles: Cyan (Input), Green (Output)

Max Momentary Loudness: -5.9 LUFS ← Measured **Input Audio Max Momentary Loudness** value

2022-09-10T09:25:23.005Z

JSON creation UTC Time Stamp

Selected stream: 1/1, 2.0, L R, eng

Measured Stream **Index**

Channels Layout

Language Tag

Dual mono detected

Warning Messages

Specified **Target** and **Tolerance** values

Integrated Loudness Target: -17 LUFS, Tolerance: 1 LU

Windows OS

Linux OS

v.1.2.2(W)

v.1.2.2(L)

Program version

Detected audio content **segments**, sorted by **types** and **durations**

Cumulative duration by type: Regular Audio 00:00:02.900 Mute 00:00:07.100 Test Tone 00:00:10.000

Integrated Loudness: -12.9 LUFS, Loudness Range: 7.3 LU, True Peak: -1.7 dBTP Main IL, LRA, TP measurement results

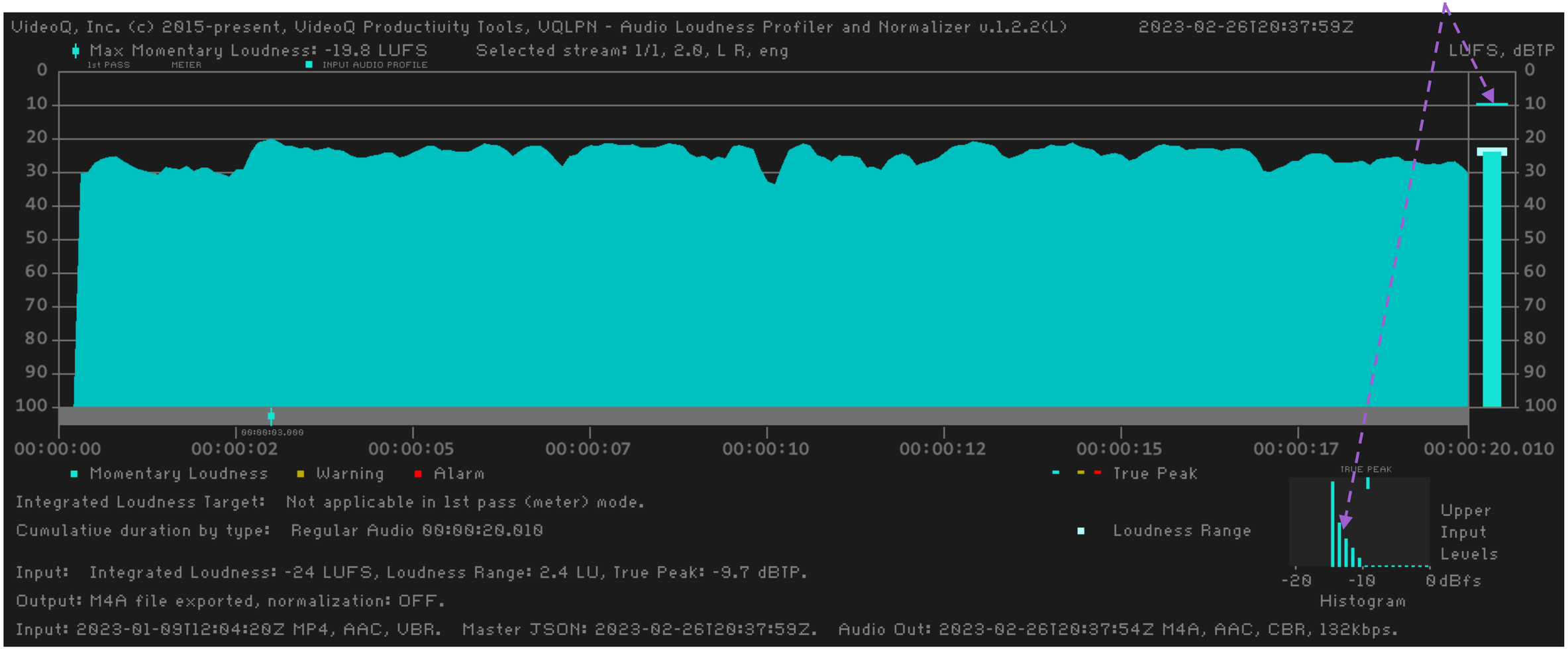
Normalized M4A file exported, gain: -4.1 dB Exported Audio file info message (if applicable)

TOC

Plot Example 3 – Professional Clip Checked

- Professional advert clip with very short mute segment at start, measured (checked) by VQLPN in “Meter” mode
- **Normalized audio stream** – **Integrated Loudness** is exactly equal to **-24 LUFS** (ATSC) standard value
- **True Peak** value is **good**, and **Upper Levels Histogram** looks **good**

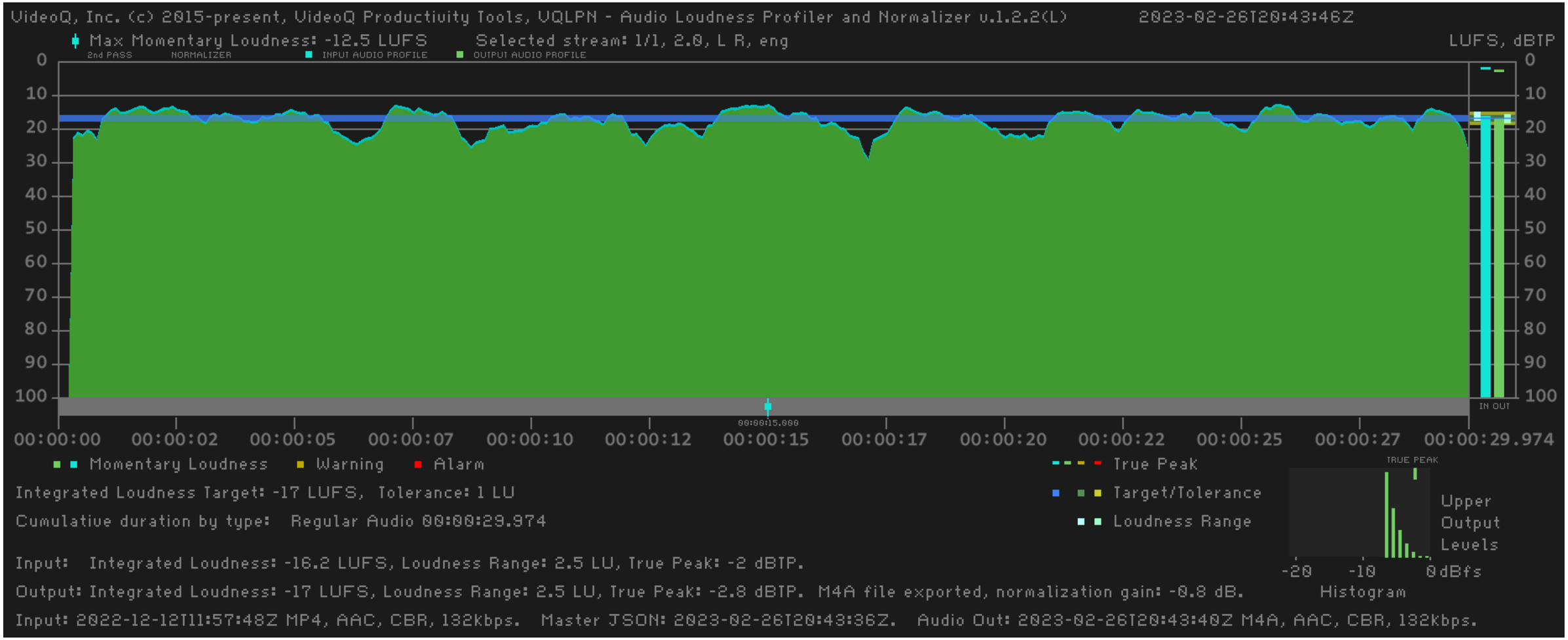
True Peak** level is **good
Upper Levels Histogram looks good



TOC

Plot Example 4 – 2-pass Processing

- Another advert clip, measured by VQLPN in "1st pass" (Meter) mode, then normalized in the "**2nd pass**" (Normalizer) mode
- **Input Integrated Loudness** is **-16.2 LUFS**, i.e. very close to -17 LUFS target value; normalization gain = 0.8 dB
- Both input and output **True Peak** values are **good**, and **Output Upper Levels Histogram** looks **good**

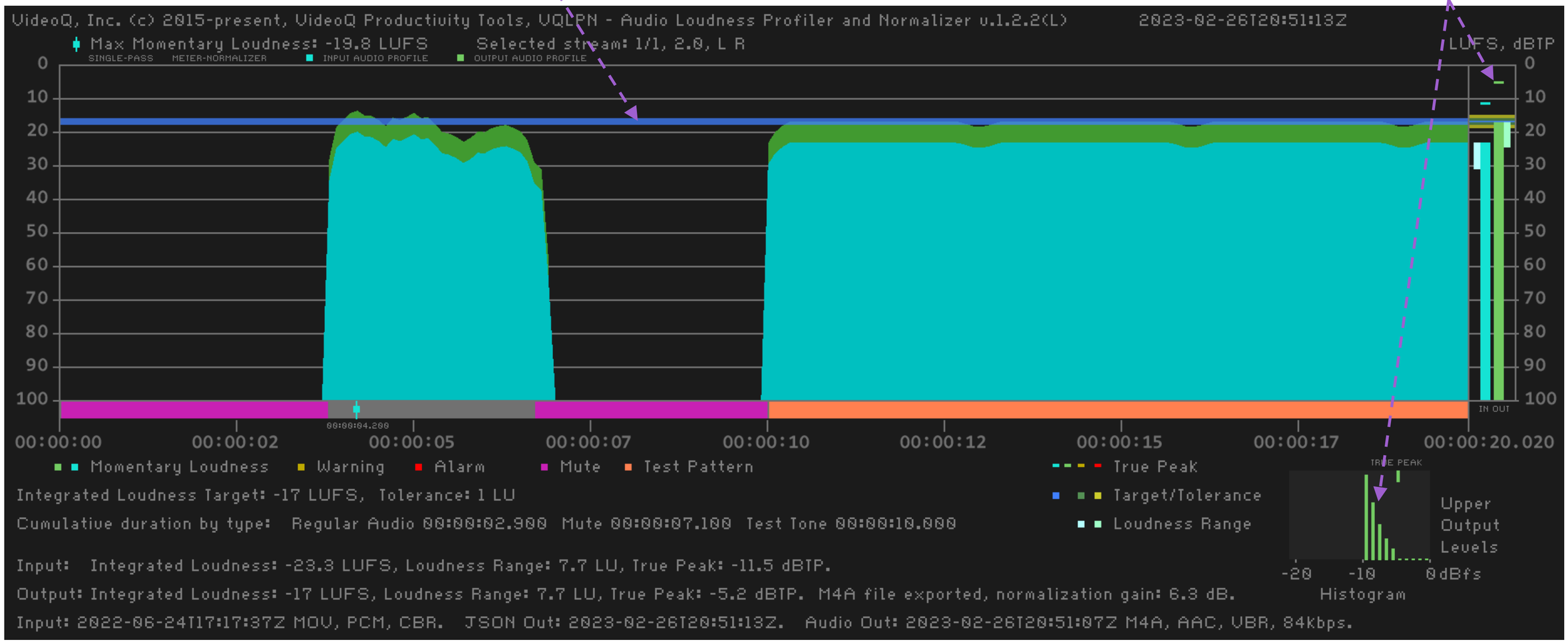


TOC

Plot Example 5 – Professional Clip Leader

- Professional clip “leader” with **voice tag segment** at start, followed by 4 seconds **mute segment** and 10 seconds of **test tone**
- Input is a broadcast quality audio – **Integrated Loudness -23.3 LUFS** is very close to **-23 LUFS** (EBU R128) specification
- **True Peak** value and **Upper Levels Histogram** look **good**
- VQLPN converted it to the specified **-17 LUFS target** in the **single-pass meter-normalizer** mode

***True Peak** levels and **Upper Levels Histogram** look good*



TOC

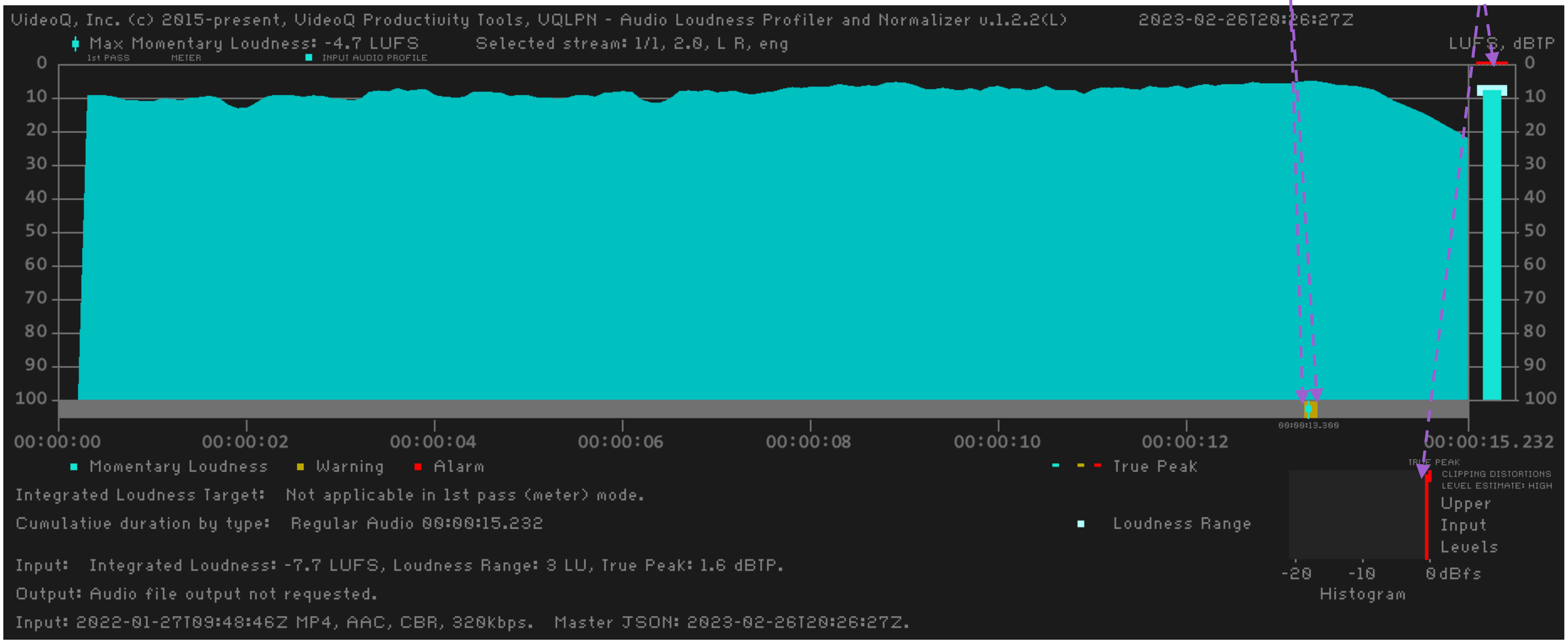
Plot Example 6a – Extremely Loud Audio

- **Integrated Loudness** is **very high: -7.7 LUFS**, probably attempting to get max possible loudness – “*up to the limit*”
- **True Peak** value is **very high: +1.6 dBTP** and the **affected timeline segment** is quite long
- **Upper Levels Histogram** shows that **Clipping Distortions** are possible
  Estimated Distortions Level: **High**, due to high histogram count at very high level

***Conclusion**: The clip is not suitable for broadcasting, but can be normalized, probably with some clipping issues*

Relatively long ***True Peak Alarm*** *segment*

True Peak** level is **very high**;* ***Upper Levels Histogram** **does not look good



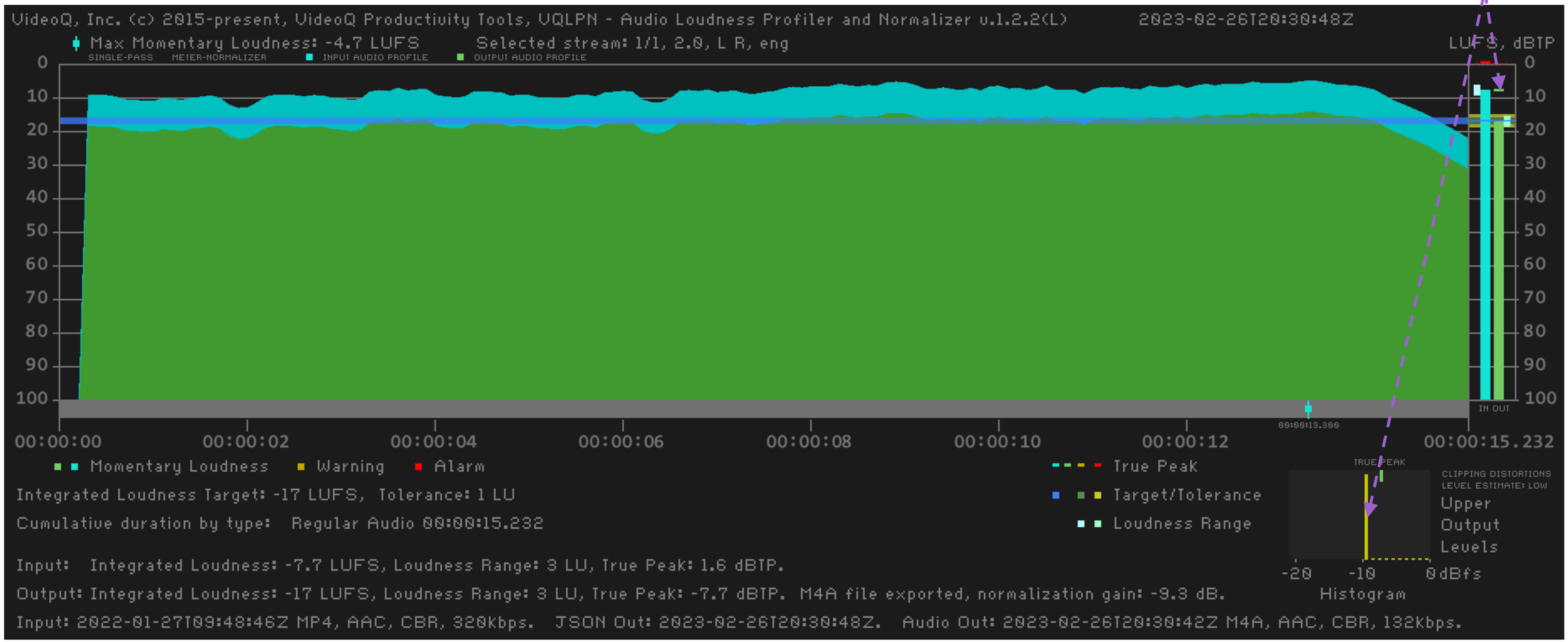
TOC

Plot Example 6b – Extremely Loud Audio Normalized

- **Integrated Loudness** after normalization is **exactly equal** to **-17 LUFS** (webcasting) target
- **Output True Peak** value looks **good**, but **Upper Levels Histogram** still shows **Low** level of possible **Clipping Distortions**

***Conclusion**: The normalized clip is suitable for webcasting, probably with minor clipping distortions issues*

***True Peak** level now **looks good**, but **Upper Levels Histogram** still shows possible problems*

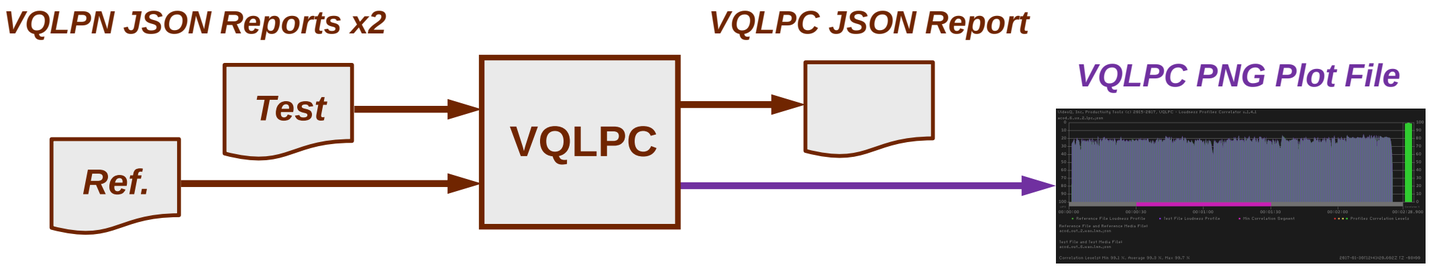


TOC

VQLPC – Loudness Profiles Correlator Utility

- 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

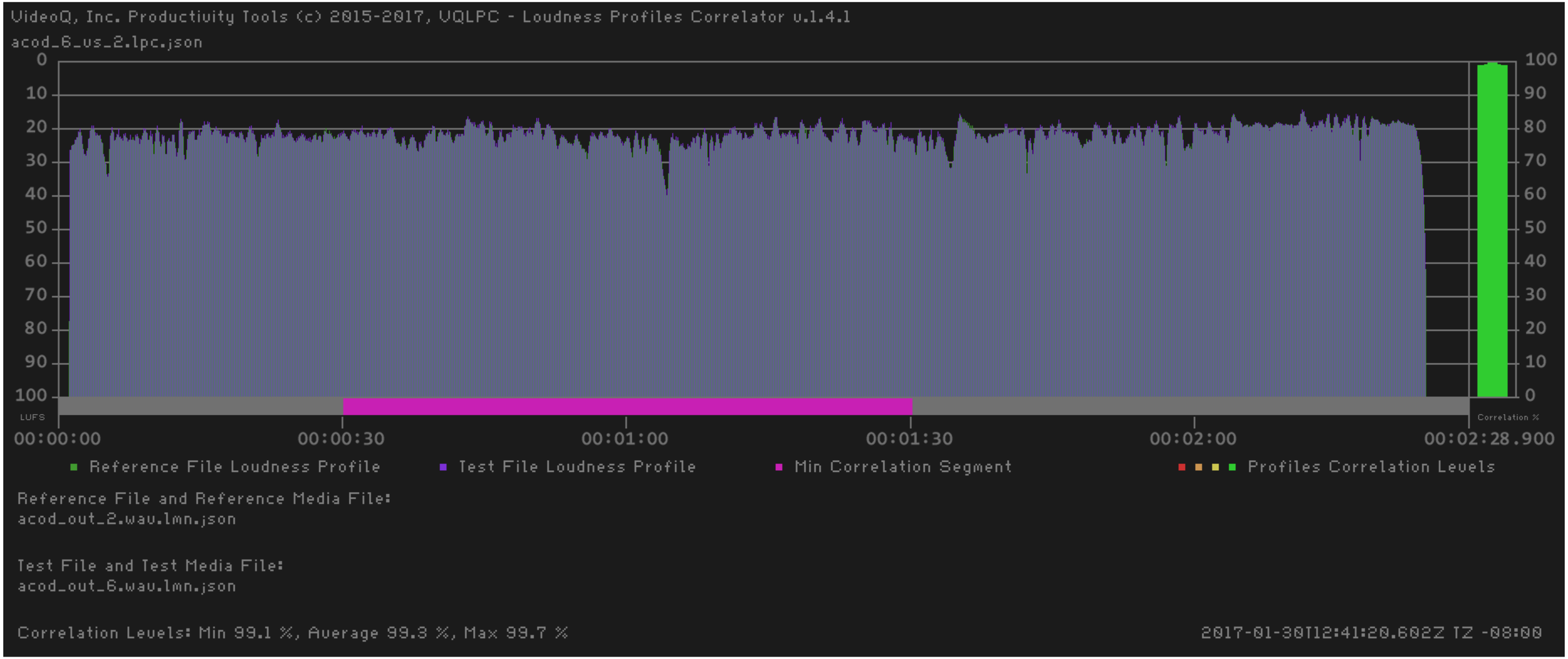
CLI interface: **vqlpc** [-p] -i refFilePath testFilePath [-o or -o outFilePath]



TOC

VQLPC 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%



TOC

VQLPC 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**

TOC

VQLPP – Loudness Profile Plotter Utility

This handy program module serves for **creation** or **re-creation** of **PNG** plot file using the **VQLPN JSON Report** file data. VQLPP is a companion program for main VQLPN module. Thus, the displayed VQLPP and VQLPN audio profiles are **identical**. Additionally, VQLPP offers **timeline zoom option**, providing more details about the selected timeline fragment.

Launching VQLPP executable without any parameters brings up the following help message:

Usage:

vqlpp [-zs TimeZoomStart -ze TimeZoomEnd] -i InFilePath.json [-o outFilePath.png]

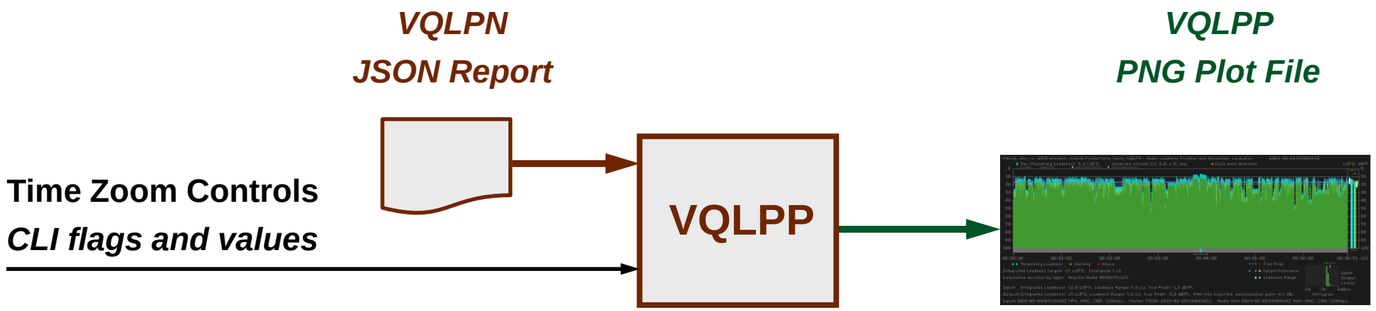
Order of flags and parameters is mandatory and cannot be changed

TimeZoom parameters specify a segment of loudness profile in float s.ms format

If TimeZoom parameters omitted, VQLPP plot shows full duration loudness profile

[-o outFilePath] option specifies full Path\FileName.png

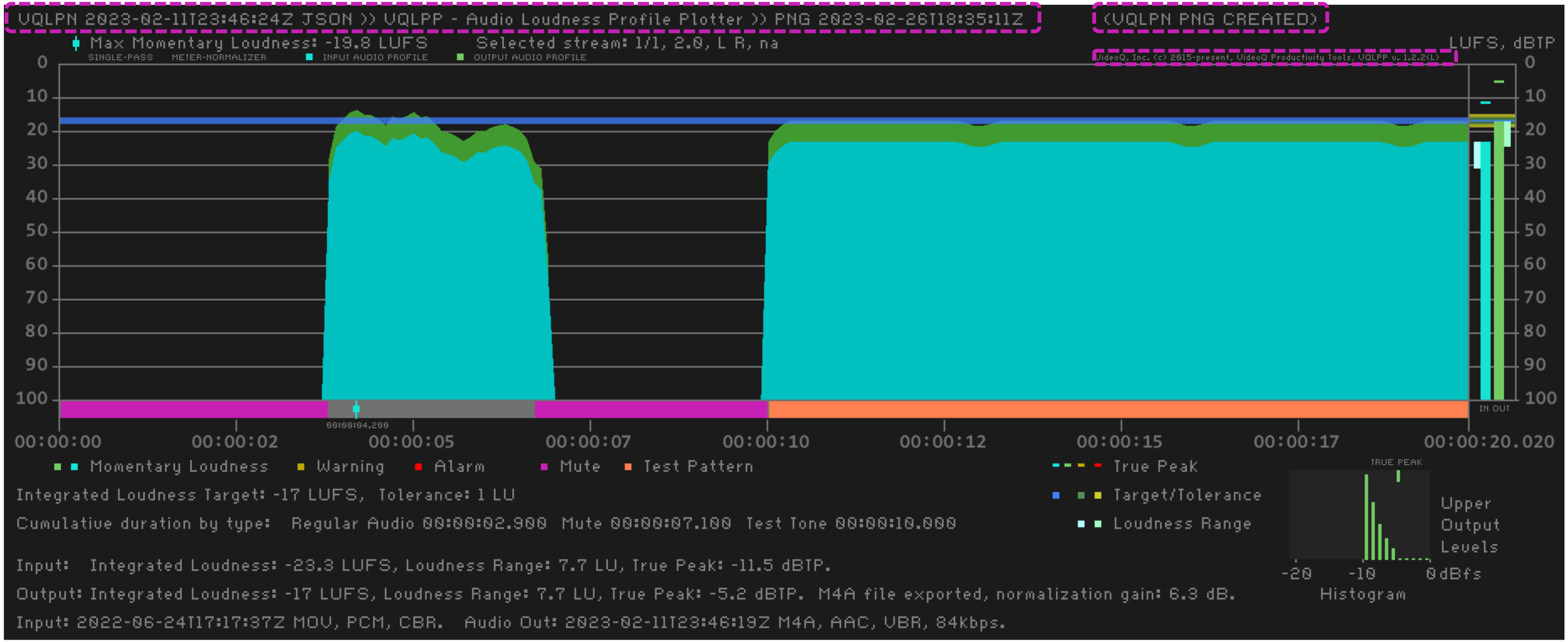
If [-o outFilePath] omitted, auto-generated outFilePath = inFilePath.vqlpp.png



TOC

VQLPP Plot Example

The components of the bitmap and all numerical values derived from VQLPP source file (VQLPN JSON Report) are identical to those of “Plot Image Example 5 – Professional Clip Leader” (see **slide #27**).
The only difference is three VQLPP message strings, designated here by Magenta boxes, and explained on the next slide.



TOC

VQLPP Plot Image Special Messages Explained

Source (VQLPN) JSON creation **UTC Time Stamp**

Program name

VQLPP PNG creation **UTC Time Stamp**

Information about the creation (i.e. the existence/availability) of **VQLPN PNG** file matching the **VQLPP PNG** output file

Copyright and program version info string

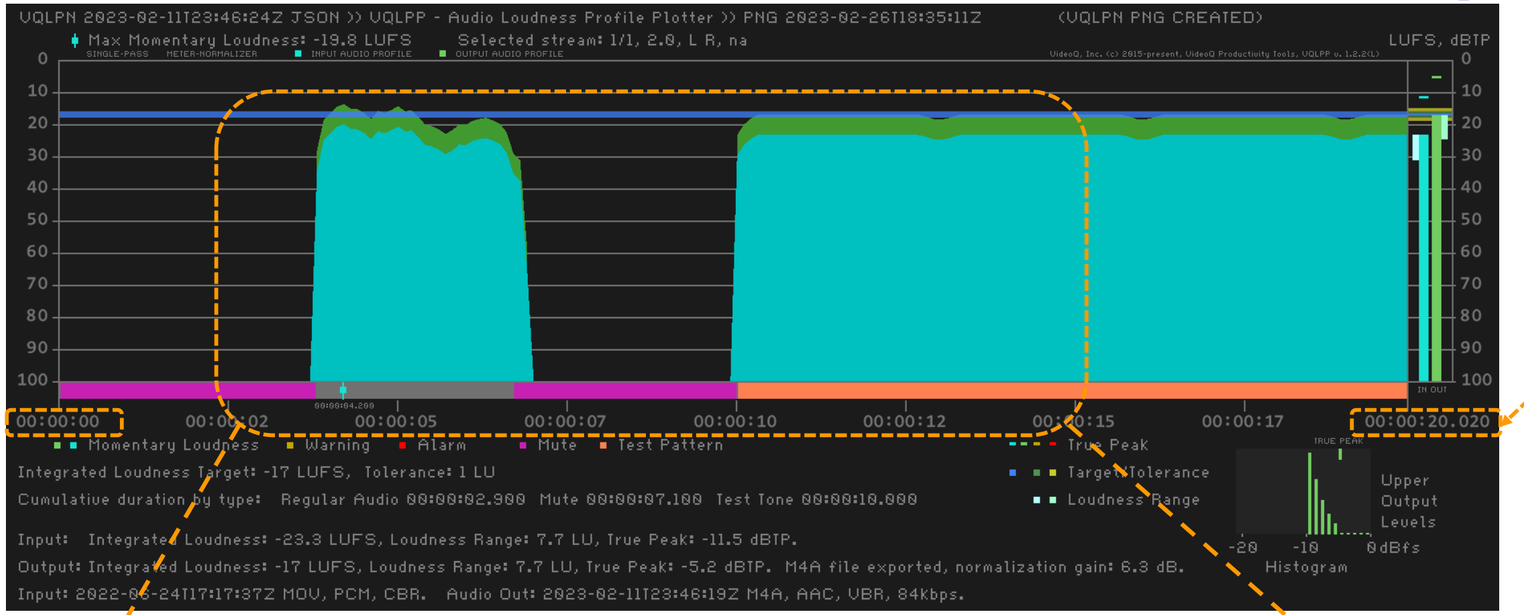
Windows OS

Linux OS

Program version

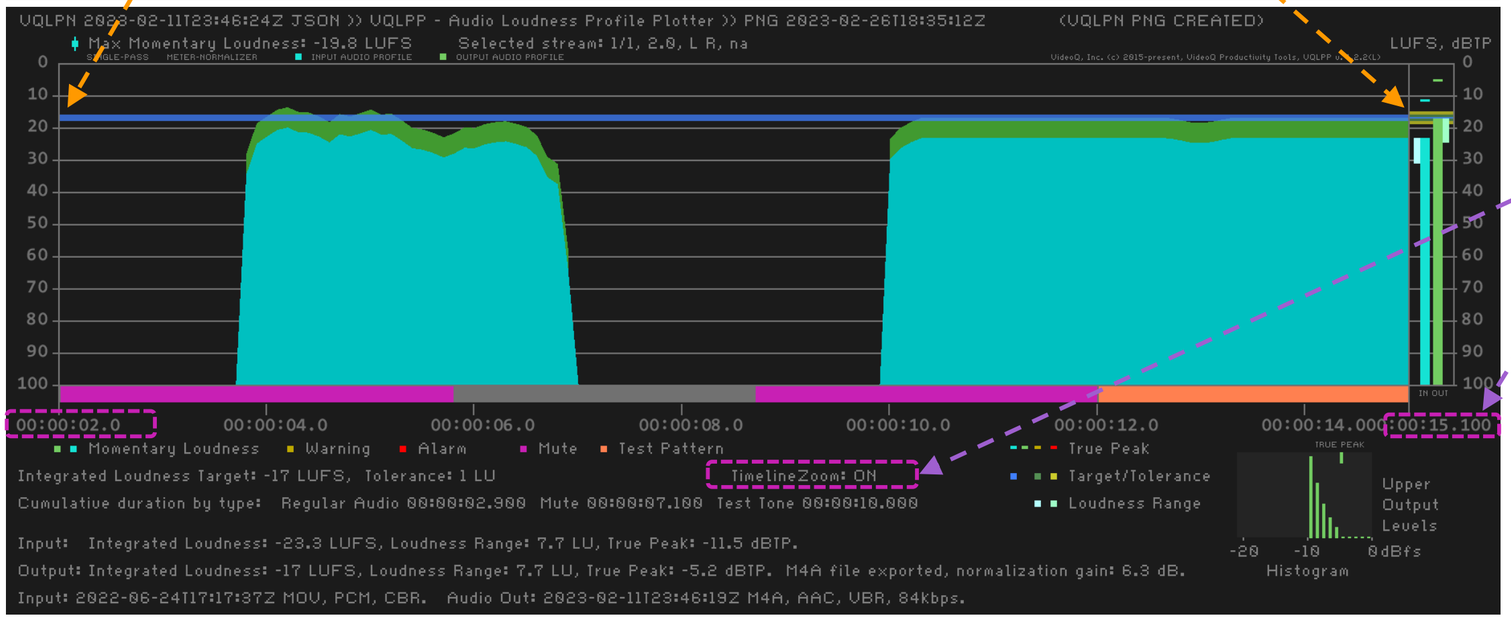
TOC

VQLPP Zoom Example



Full duration *(default mode)*:
from **0.0s** to **20.02s.**

Zoom OFF



User-selected timeline fragment:
from **2.0s** to **15.0s.**

Zoom ON

TOC

VQILM – Integrated Loudness Meter Utility

This program module serves for automated periodic **spot-checks** of **Audio Stream** loudness parameters.

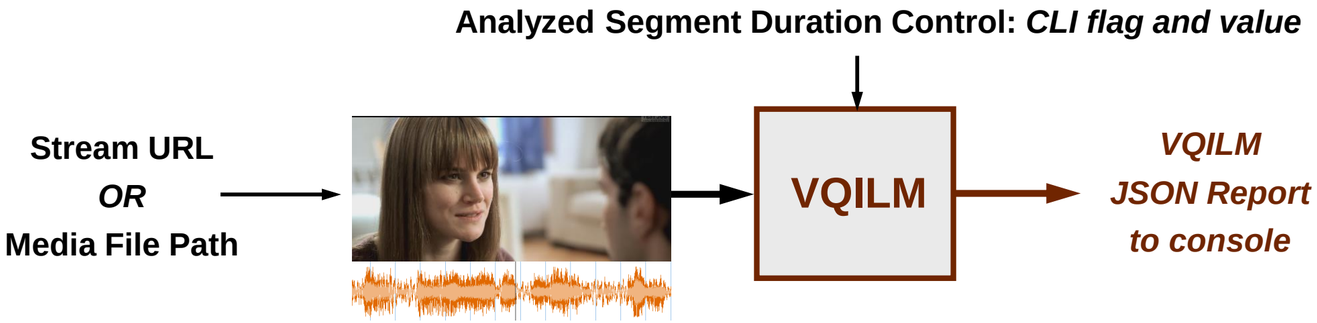
The most important parameter measured by VQILM is the **Integrated Loudness** value in **LUFS**.

JSON Report created by VQILM module is sent to console ***only***, and it is much ***shorter*** than VQLPN Report.
This provides for grouping together tens of reports, and number of groups stored in the database can be several thousands.

Significant advantage of VQILM is its capability to analyze AV Streams specified by webcast **URL**, such as " *https://….* ".

For given URL the **total duration** of concatenated streams **playout sequence** is typically ***unknown*** (so it should be treated as ***infinite***). Thus, in the **URL input mode** VQILM **must** use **Analyzed Segment Duration Control**.

In case of **file** input mode, stream **full duration** is **known**, thus the **segment duration control** is **optional**; VQILM can measure either **full file duration** or the specified timeline **segment** – from the file start to the specified segment duration end.



TOC

VQILM Usage Info

Launching VQILM executable without any parameters, or with –h flag, brings up the following help message:

Usage:

vqilm [-t ASD] -i sourceURL|sourceFilePath

Order of flags and parameters is mandatory and cannot be changed

VQILM auto-detects and switches source URL|File processing Mode

[-t ASD] option defines Analyzed Segment Duration (integer sec)

In URL Mode absence of [-t ASD] option means default ASD value = 90s

In File Mode absence of [-t ASD] option means ASD = full source file duration

Short JSON Report always sent to console (stdout wcout)

If sourceFilePath contains spaces or special characters use double quotes

VQILM supports Source URL|FilePath in multi-lingual UTF-8 encoding format

TOC

VQILM JSON Report Examples

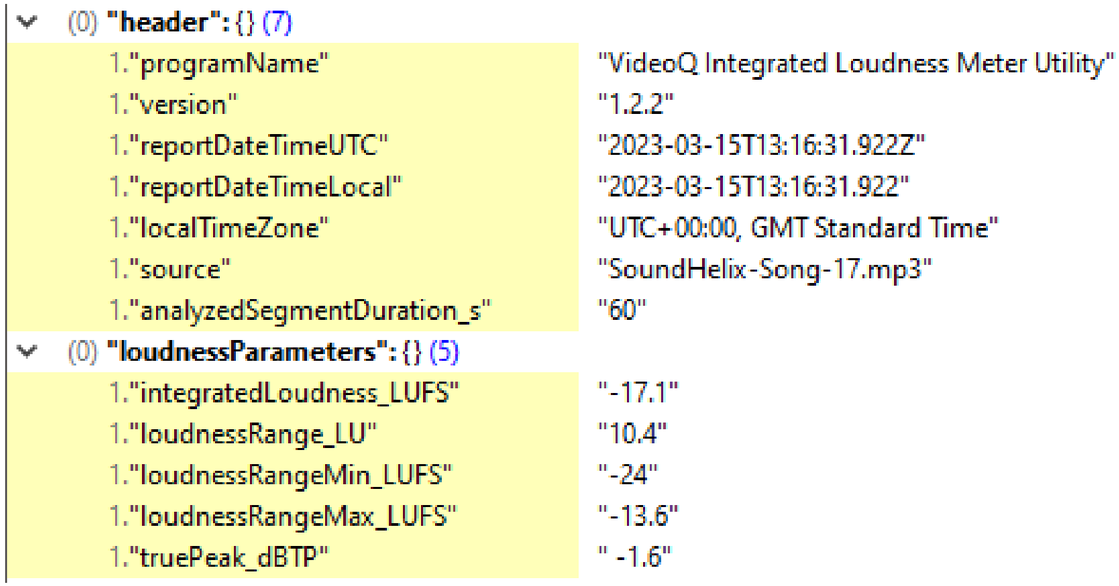
URL

```
C:\__VQILM_Mar2023>vqilm -t 60 -i "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-17.mp3"
{
    "header": {
        "programName": "VideoQ Integrated Loudness Meter Utility",
        "version": "1.2.2",
        "reportDateTimeUTC": "2023-03-15T13:12:04.806Z",
        "reportDateTimeLocal": "2023-03-15T13:12:04.806",
        "localTimeZone": "UTC+00:00, GMT Standard Time",
        "source": "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-17.mp3",
        "analyzedSegmentDuration_s": "60"
    },
    "loudnessParameters": {
        "integratedLoudness_LUFS": "-17.1",
        "loudnessRange_LU": "10.4",
        "loudnessRangeMin_LUFS": "-24",
        "loudnessRangeMax_LUFS": "-13.6",
        "truePeak_dBTP": " -1.6"
    }
}
```

File

```
C:\__VQILM_Mar2023>vqilm -t 60 -i "SoundHelix-Song-17.mp3"
{
    "header": {
        "programName": "VideoQ Integrated Loudness Meter Utility",
        "version": "1.2.2",
        "reportDateTimeUTC": "2023-03-15T13:16:31.922Z",
        "reportDateTimeLocal": "2023-03-15T13:16:31.922",
        "localTimeZone": "UTC+00:00, GMT Standard Time",
        "source": "SoundHelix-Song-17.mp3",
        "analyzedSegmentDuration_s": "60"
    },
    "loudnessParameters": {
        "integratedLoudness_LUFS": "-17.1",
        "loudnessRange_LU": "10.4",
        "loudnessRangeMin_LUFS": "-24",
        "loudnessRangeMax_LUFS": "-13.6",
        "truePeak_dBTP": " -1.6"
    }
}
```

Console stdout captured as JSON **File**



TOC

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