

Video Clarity



Tools for Video Analysis



ClearView Analyzers

Command Line Interface



TABLE OF CONTENTS

TABLE OF CONTENTS	2
1. CLEARVIEW VIDEO ANALYSIS SYSTEM	5
CAPTURE FEATURES:	5
ANALYSIS FEATURES:	5
<i>No Reference Metrics</i>	5
<i>Full Reference Metrics</i>	5
PLAYBACK FEATURES:	5
2. INTRODUCTION	7
SETUP	7
CVSERVER	7
CV	7
CONFIG	7
PATH	7
?	8
3. TEST SETUP	9
AUTOMATED VIDEO QUALITY TESTING PROCESS	9
SCRIPTED TEST OPERATION	9
4. NEW FEATURES FOR VERSION 7.0	11
<i>New</i>	11
<i>Updated</i>	11
5. COMMANDS	12
GLOBAL	12
<i>boardTemp</i>	12
<i>exit</i>	12
<i>freeFrames</i>	12
<i>shellCmd</i>	13
CONFIGURATION	13
<i>analogFormat</i>	13
<i>imageFormat</i>	14
<i>inOut</i>	14
<i>overlay</i>	14
<i>reset</i>	15
VANC	15
<i>videoFormat</i>	16
<i>File Movement</i>	16
<i>addLibrary</i>	16
<i>addSequence</i>	17
<i>configExport</i>	17
<i>export</i>	18
<i>import</i>	18
<i>libraryActivate</i>	19
<i>newLibrary</i>	19
<i>seqDelete</i>	19
VIEW MODE SETTINGS	20
<i>aMinusBConfig</i>	20
<i>viewmode</i>	20



PLAYBACK.....	21
<i>autoalign</i>	21
<i>configIntelligentAlign</i>	21
<i>first</i>	21
<i>fieldmode</i>	22
<i>goto</i>	22
<i>jogFwd</i>	22
<i>jogRev</i>	22
<i>last</i>	23
<i>mapA</i>	23
<i>mapB</i>	23
<i>pause</i>	24
<i>play</i>	24
<i>playmode</i>	24
<i>speed</i>	24
<i>stop</i>	25
<i>unmapA</i>	25
<i>unmapB</i>	25
OUTPUT	25
<i>customVideoFormat</i>	25
Notes: <i>none</i>	25
<i>dualOutput</i>	26
<i>outputHeight</i>	26
<i>OutputRefresh</i>	26
<i>outputWidth</i>	26
<i>videoOutput</i>	27
CAPTURE	27
<i>inputHeight</i>	27
<i>inputRefresh</i>	27
<i>inputWidth</i>	27
<i>videoInput</i>	28
<i>record</i>	29
METRICS.....	30
<i>audioMetricFreq</i>	30
<i>audioMetricPEAQ</i>	30
<i>configaFreq</i>	31
<i>configPEAQ</i>	31
<i>configDMOS</i>	32
<i>configPSNR</i>	32
<i>configJND</i>	32
<i>dmos</i>	33
<i>jnd</i>	34
<i>normalize</i>	34
<i>normalizeOffsets</i>	34
<i>spatialAlign</i>	35
<i>spatialOffsets</i>	35
<i>metricWindow</i>	35
<i>psnr</i>	35
<i>spatial</i>	36
<i>temporal</i>	37
6. ERRORS	38
NOT RECOGNIZED COMMAND	38

Video Clarity, Inc.
Phone: 408-379-6952
Fax: 408-379-6221
sales@videoclarity.com
<http://www.videoclarity.com>



Tools for Video Analysis



CONNECT() FAILED	38
ERROR OPENING CONFIG FILE	38



1. ClearView Video Analysis System

The ClearView Video Analysis systems (ClearView) provide video researchers, compression developers, hardware designers, and QA/QC engineers, broadcast, cable and IPTV operators with the unique ability to play, view, record, and objectively analyze video.

Capture Features:

The ClearView Command Line Interface only works with the Broadcast I/O module; thus, it allows the capture of video content from -- file, SDI, HD-SDI, Component, Composite, and S-Video. Regardless of the input, the video is converted, based on user choice, to fully uncompressed 4:2:2 Y'CbCr or RGBA. Any inputted video sequence, regardless of dimensions, can be cropped or matted to fit into the selected output raster.

Analysis Features:

Analysis begins on any two video sequences which share the same resolution and color space. The goal is to calculate the video quality without human intervention – termed objective analysis. ClearView calculates the pixel differences between the video sequences and displays them as A-B with threshold and add-back. Add-back shows where pixels are greater than the threshold. Without Add-back shows the actual value of the pixel differences. The Pixel Value tool shows the Y'CbCr or RGB values at the pixel location for each video sequence.

ClearView applies various objective metrics to each frame of the video sequences, generates graphs, applies thresholds and logs the results.

No Reference Metrics

- Luminance Value (PSNR to Black)
- Chrominance Value (PSNR to Baseline)
- Number of Edges (Spatial / Sobel Filter)
- Frame-to-Frame Differences (Temporal)

Full Reference Metrics

- PSNR
- PSNR with color/brightness Normalization
- Sarnoff's JND
- DMOS with MS-SSIM

ClearView can easily be programmed to display video sequences for the expert viewers; while recording the objective metric score.

To aid in subjective video analysis, ClearView displays the video sequences at any rate in side-by-side, seamless split, or split mirror.

Playback Features:

Output rates are independent from input rates; so any video sequence can be outputted at rates in excess of 120Hz. The user has control over shuttle rates, jog, color look-up tables, zoom/pan, and field display. The video sequences are previewed within the ClearView Interface and sent to HD-SDI, SDI, Component, S-Video, and Composite. Normally, the video sequences are shown on the same display,

Video Clarity, Inc.
Phone: 408-379-6952
Fax: 408-379-6221
sales@videoclarity.com
<http://www.videoclarity.com>

Video **Clarity**

Tools for Video Analysis



but each video sequence can be outputted via a separate HD-SDI/SDI link. Video Sequence or a portion of the video sequence can also be exported as uncompressed BMP, RAW or AVI files.

2. Introduction

Setup

The command line interface consists of 3 files that must be loaded before starting:

- CVServer
- CV
- Config

CVServer

CVServer resides on the machine running the ClearView software. It converts the CV commands to appropriate messages to start ClearView. CVServer must be activated before you can run a CV command. It is preferred to place CVServer in C:\Program Files\VideoClarity\ClearView since the path is already set.

When running CVServer from the command line there needs to be a port number and timeout. CVServer communicates through port 7. The command line should look like "cvserver<space>7<space>5".

CV

CV is the command line processor. It communicates via sockets to CVServer. It reads the Config file to find out where the ClearView machine sits on the network. A list of CV commands is in the table of contents above. Each command is detailed below.

It is preferred to place CV in C:\Program Files\VideoClarity\ClearView.

Config

Config holds the IP address of the machine running ClearView software.

This file should be placed in C:\Program Files\VideoClarity\ClearView.

Path

The system path is a list of folders, separated by a semicolon, which identifies the folders that the system should search when looking for files that are called from the Run dialog box, command line, or other processes. Normal program installation changes this path to include the program's installation path. To manually change the system path, perform these steps:

- Start the System Control Panel applet (Start → Settings → Control Panel → System).
- Select the Advanced tab.
- Click the Environment Variables button.
- Under System Variables, select Path, then click Edit.

Add the folder in which CV/CVServer and config reside, preferably C:\Program Files\VideoClarity\ClearView. (Remember to place a ";" before this new entry) Click OK.

Video Clarity, Inc.
Phone: 408-379-6952
Fax: 408-379-6221
sales@videoclarity.com
<http://www.videoclarity.com>



?

If you ever need a list of commands that are available in cvserver one can type "cv ?" and a list of commands will be listed.

If you would like to see how a command is used and do not have the CLI Documentation handy one can type `cv ? <command>`, example: `cv ? configjnd`.



3. Test Setup

Automated video quality testing process

Load/Capture Reference Sequence(s)	Source material for a reference sequence can be either imported from file or captured from ClearView's SDI Input. The "Reference Sequence" will be output as uncompressed SDI video to the DUT (Device Under Test) video input.
Output Source Video Sequence from ClearView HD/SD-SDI to DUT	ClearView can be told to start playing at anytime. If the DUT has a known startup delay, ClearView can first pause on frame 0 for x number of seconds before starting to play the sequence. This ensures that the captured result contains frame 0 of the reference sequence. Alternatively, measurements can be set to start at frame X instead of frame 0.
Device Under Test processes video and outputs uncompressed video	This is most likely a video encoder, video processor, or video decoder/STB. ClearView outputs uncompressed video to the DUT, and accepts uncompressed video input or compressed/uncompressed files.
Output from DUT is captured by ClearView to a new sequence (GoldResult)	ClearView can simultaneously playout and record up to 1080i (or more precisely anything that requires single-link SDI). If you exceed the single link requirements, then ClearView can play or record.
Perform Auto Alignment. Create safe inpoint/output for testing	After ClearView has recorded from the DUT, place the original sequence in Viewport B and the newly recorded sequence in Viewport A. Advance to the 2 nd frame (or beyond) of the original sequence and run automatic temporal then spatial alignment. In addition, you can run normalization to equalize the brightness/hue between the videos.
Visually Inspect GoldResult. Run Metrics to define testing thresholds	Run objective metrics on the original and recorded sequences and apply a threshold creating a pass/fail condition. Alternatively, you can simply view the results and make your own subjective analysis.

Scripted Test Operation

Output Source Video Sequence from ClearView HD-SDI	cv MapA ReferenceSequence 0 300 cv Play
Device Under Test processes video and outputs uncompressed video	Script commands sent to start DUT
Output from DUT is captured by ClearView to a new sequence (TestSeq)	cv record TestSeq number_frames
A defined portion of the captured clip is selected for measurement	cv MapA TestSeq cv MapB GoldResult
Perform Auto	cv inout 0 10 290

Video Clarity, Inc.
Phone: 408-379-6952
Fax: 408-379-6221
sales@videoclarity.com
<http://www.videoclarity.com>



Tools for Video Analysis



Alignment. Create safe inpoint/output for testing	cv autoalign
Quality Metrics are performed PSNR, SSIM, Temporal, JND	cv psnr TestSeq.psnr 10
Pass or Fail and Detailed results are logged.	Received: Threshold Failures = 0 Full Frame-by-Frame log sent to TestSeq.psnr



4. New Features for Version 7.0

The following commands have been updated. All previous command should run without change with exception to dmos.

New

- AddLibrary** – adds an existing library to Clearview
- AddSequence** – adds a sequence from one library to another
- AudioMetricFreq** – will run aFreq
- AudioMetricPEAQ** – will run PEAQ
- BoardTemp** – returns the temperature of the broadcast board
- ConfigAfreq** – configures aFreq metric
- ConfigIntelligentAlign** – configures intelligent alignment
- ConfigPEAQ** – configures PEAQ metric
- DualOutput** – will enable dualoutput
- Exit** – Closes ClearView
- NewLibrary** – creates a new library
- ShellCmd** –runs a command on the local machine remotely
- UnmapA** – removes a sequence from Viewport A
- UnmapB** – removes a sequence from Viewport B

Updated

- DMOS** – added ability to run Chorma
- AutoAlign** – added intelligent alignment option
- ConfigExport** – added ability to export QuickTime files
- VideoFormat** – 1080pB formats have been added. The “A” command still work “1080pXX” with this release though it is recommended to change them to “1080pXXA”
- ViewMode** – allow split points and direction (horizontal or vertical) on Side/Seamless/Mirror viewmodes

5. Commands

The Video Clarity Clear View Command Line Interface allows the user to control any ClearView machine, which can be seen (open socket call). The general command structure is as follows: CV CommandName CommandArguments. The command is echo-ed back to the console timestamped, and the results of the command are displayed (timestamped) on the next line.

Global

boardTemp

Description: Returns the temperature of the broadcast board
Syntax: boardTemp <iModule>
Input: **iModule** 0 – Broadcast Output Module 1
1 – Broadcast Output Module 2
Output: Received: Success
Received: Failure
Example: *cv boardTemp 0*
Notes: If the module is omitted 0 is expected.

exit

Description: Closes ClearView
Syntax: Exit
Input: **NONE**
Output: Received: Success
Received: Failure
Example: *cv exit*
Notes: none

freeFrames

Description: FreeFrames returns the total number of frames that are free based on the current video format
Syntax: FreeFrames <bMemory>
Input: **bMemory** 0 – Returns free frames on disk
1 – Returns free frames on memory
Output: Received: Success: Free Frames = <number of free frames>
Received: Failure
Example: *cv freeframes 1*
Notes: none



shellCmd

Description: ShellCmd runs a command on the local machine remotely
Syntax: shellCmd <cCmd>
Input: **cmd** Text Path and location of a command
Output: Received: Success
Received: Failure
Example: cv "C:\My Test Program.exe"
Notes: none

Configuration

analogFormat

Description: AnalogFormat will change the analog video format of the currently selected output device
Syntax: AnalogFormat <cFormat>
Input: **cFormat** **525ComponentRGB** – 525 Component RGB
525ComponentUS – 525 Component US
525ComponentBetaUS – 525 Component Beta US
525ComponentBetaJapan – 525 Component Beta Japan
525CompositeUS – 525 Composite US
525CompositeJapan – 525 Composite Japan
625ComponentRGB – 625 Component RGB
625ComponentSMPTE – 625 Component SMPTE
625Composite – 625 Composite
1080iRGB – 1080i RGB
1080psfRGB – 1080psf RGB
720pRGB – 720p RGB
1080iSMPTE – 1080i SMPTE
1080psfSMPTE – 1080i psf SMPTE
1080iXVGA – 1080i XVGA
1080psfXVGA – 1080psf XVGA
720pXVGA – 720p XVGA
Output: Received: Success
Received: Failure
Example: cv VideoFormat 1080iRGB
Notes: none



imageFormat

Description: ImageFormat will change the image format of the video

Syntax: ImageFormat <cFormat>

Input: **cFormat** **YCbCr8** – YCbCr 8bpc
 YCbCr10 – YCbCr 10bpc
 ARGB – ARGB 8bpc
 RGBA – RGBA 8bpc
 RGB8 – RGB 8bpc
 BGR8 – BGR 8bpc
 RGB10 – RGB 10bpc

Output: Received: Success
 Received: Failure

Example: cv ImageFormat YCbCr10

Notes: none

inOut

Description: InOut can be used to both change the first/last frame of a sequence loaded or to give the first/last frames of the sequence loaded

Syntax: InOut <eViewport> <iFirst> <iLast>

Input: **eViewport** **0** – Viewport A
 1 – Viewport B

iFirst

iLast

Output: Received: Success
 Received: Failure
 Received: Success: Viewport = <viewport>: First = <First>, Last = <Last>

Example: cv inout 0 ; returns current first/last frame
 cv inout 0 10 80 ; sets Viewport A, first frame = 10, last frame = 80

Notes: If <iFirst> AND <iLast> are omitted then the current frame set for first and last will be returned for the corresponding viewport, if <iFirst> AND <iLast> are used the first/last frame used for the corresponding viewport will be changed

overlay

Description: turns off/on overlay

Syntax: Overlay <bOverlay>

Input: **bOverlay** **1** – On
 0 – Off

Output: Received: Success
 Received: Failure

Example: cv overlay 1

Notes: none



reset

Description: Reset will perform the same actions as pressing "Reset" on the Clear View Graphical User Interface. All sequences will be removed and Clear View will be reset to the start up state

Syntax: reset
Input: **NONE**
Output: Received: Success
Received: Failure

Example: *cv reset*

Notes: none

VANC

Description: will turn on/off the VANC option

Syntax: VANC <bOn>

Input: **bOn** 1 = On
 0 = Off

Output: Received: Success
Received: Failure

Example: *cv VANC 1*

Notes: none



videoFormat

Description: VideoFormat will change the video format of the currently selected output device

Syntax: VideoFormat <cFormat>

Input: **cFormat** **525** – 525 59.95Hz
 625 – 625 50.00 Hz
 1080i50 – 1080i 50.00 Hz
 1080i59 – 1080i 59.94 Hz
 1080i60 – 1080i 60.00 Hz
 720p23 – 720p 23.98 Hz
 720p50 – 720p 50.00 Hz
 720p59 – 720p 59.94 Hz
 720p60 – 720p 60.00 Hz
 1080p23 – 1080p 23.98 Hz
 1080p24 – 1080p 24.00 Hz
 1080p25 – 1080p 25.00 Hz
 1080p29 – 1080p 29.97 Hz
 1080p30 – 1080p 30.00 Hz
 1080p50a – 1080p 50.00a Hz
 1080p59a – 1080p 59.94a Hz
 1080p60a – 1080p 60.00a Hz
 1080p50a – 1080p 50.00a Hz
 1080p59a – 1080p 59.94a Hz
 1080p60a – 1080p 60.00a Hz
 1080psf23 – 1080p sf 23.98 Hz
 1080psf24 – 1080p sf 24.00 Hz

Output: Received: Success
 Received: Failure

Example: cv VideoFormat 1080i59

Notes: none

File Movement

addLibrary

Description: AddLibrary adds an existing library to Clearview

Syntax: AddLibrary <cPath>

Input: **cPath** File path to a valid existing Clear View library

Output: Received: Success
 Received: Failure

Example: cv AddLibrary "E:\720p YCbCr 8bpc"

Notes: If the path has spaces the path needs to be enclosed by double quotes
 One should not include the "\"



addSequence

Description: AddSequence adds a sequence from one library to another
Syntax: AddSequence <cDstLibrary> <cSequence> <cSrcLibraryPath>
Input: **cDstLibrary** File path to library in which new sequence will be added
cSequence Name of new sequence
cSrcLibrary Any path to where the sequences file exists
Output: Received: Success
Received: Failure
Example: cv AddSequence "G:\TV" RTM_20100929_03_00_00 "C:\Temp":
Notes: If the path has spaces the path needs to be enclosed by double quotes
One should not include the "\"
This does NOT copy the video or audio files – it assumes the calling script does this

configExport

Description: configExport will configure the export functionality
Syntax: Export <cType> <framerate> <b16Bit> <bMultiframefile> <b420> <bPlanar>
<bPlanar>
ConfigExport MOV <b8AudioChannels>
Input: **cType** BMP = BMP
AVI = AVI
RAW = RAW
MOV=Quicktime
iFrameRate OPTIONAL - Only needed for AVI
b16Bit 0 = off, use current bit format
1 = on, pad out to 16 bits
bMultiFrameFile 0 = off, single file per frame
1 = on, single file for all frames
b420 0 = off, 4:2:2
1 = on, 4:2:0
bPlanar 0 = off, interleaved format
1 = on, planar format
b8AudioChannels 0 = off, 2 channel audio export
1 = on, 8 channel audio export
Output: Received: Success
Received: Failure
Example: cv configExport AVI 60
cv configExport RAW 0 0 1 1 0
cv configExport MOV 0
Notes: iFormatRate is used for AVI and when bMultiFrameFile is set
b16Bit, bMultiFrameFile, b420 and bPlanar are only applicable to RAW exports
If b16Bit is set one cannot set bMultiFrameFile, b420 or bPlanar
bPlanar can only be set if exporting b420
Note there are two different syntaxes for ConfigExport - one for Mov and one for the other export types:



export

Description: Export will export the given sequence
Syntax: Export <cSequenceName> <iFirst> <iLast> <cDestFile>
Input: **cSequenceName** Any sequence name that conforms to Clear View sequence naming rules
iFirst First frame to be exported
iLast Last frame to be exported
cDestFile Full path and name of file to be created
Output: Received: Success
Received: Failure
Example: cv export "Impairments 1080 YCbCr 8-bit" 0 149 C:\Output\MyFile.avi
Notes: Export path must be from root, i.e. C:\My Clips\Image.avi not \Image.avi
If the source path or sequence name has spaces they need to be enclosed by double quotes

import

Description: Import will import the given files, playlists or logs
Syntax: Import <cSrcPathFile> <cSequenceName> <bToMemory> <iFirst> <iLast>
Import <cSrcPathFile>
Input: **cSrcPathFile** Any file path, including file name
cSequenceName Any sequence name that conforms to Clear View sequence naming rules
bToMemory **1** – record to memory
0 – record to disk
iFirst OPTIONAL – first frame to import
iLast OPTIONAL – last frame to import
Output: Received: Success
Received: Failure
Example: cv Import "E:\Clips\football\YCbCr 8-bit Football 8Mbps 1080 YCbCr 8-bit.avi" MyImport
cv Import "E:\List\Test.psnr"
Notes: There are two different syntaxes for Import. The syntax for playlists and logs does not have any more parameters than the location of the file. The syntax for files includes where to load the file as well as first/last frame to import.
Path must be from root, i.e. C:\My Clips\Image.avi not \Image.avi
If the source path or sequence name has spaces they need to be enclosed by double quotes
If the first and last frame parameter are omitted all frames will be imported
The first frame must be less than or equal to the last frame.



libraryActivate

Description: LibraryActivate changes the current active library
Syntax: LibraryActivate <cPath>
Input: **cPath** File path to a valid existing Clear View library
Output: Received: Success
Received: Failure
Example: cv libraryActivate "E:\720p YCbCr 8bpc"
Notes: Path must be from root, i.e. E:\My Clips\ not \My Clips
If the path has spaces the path needs to be enclosed by double quotes
One should not include the "\"

newLibrary

Description: NewLibrary creates a new library
Syntax: NewLibrary <cPath>
Input: **cPath** File path to a valid existing Clear View library
Output: Received: Success
Received: Failure
Example: cv newLibrary "E:\720p YCbCr 8bpc"
Notes: If the path has spaces the path needs to be enclosed by double quotes
One should not include the "\"

seqDelete

Description: will delete a sequence from the library manager
Syntax: SeqDelete <clibrary> <cSequenceName>
Input: **cScrPathFile** Any Library Path
cSequenceName Any sequence name that conforms to Clear View sequence naming rules
Output: Received: Success
Received: Failure
Example: cv seqDelete "F:\1080i YCbCr" YCbCr 8-bit Football 8Mbps 1080 YCbCr 8-bit
Notes: Path must be from root
If the source path or sequence name has spaces they need to be enclosed by double quotes



View Mode Settings

aMinusBConfig

Description: AMinusBConfig allows a user to set A – B settings
Syntax: aMinusBConfig <bUseThreshold> <iThreshold> <bChroma> <bAddBack>
Input:

bUseThreshold	0 = off
	1 = on
iThreshold	Numerical value
bChroma	0 = off
	1 = on
bAddBack	0 = off
	1 = on

Output: Received: Success
Received: Failure
Example: cv AMinusBConfig 14 0 1
Notes: One must be in A – B mode before setting this configuration

viewmode

Description: Viewmode will change the current viewmode that Clear View is in
Syntax: viewmode <cMode> <bHorizontal> <iSplitPoint>
Input:

cMode	A	A Only
	B	B only
	Side	Side-by-Side
	Seamless	Seamless-Split
	Mirror	Split-Mirror
	AMinusB	A-B
bHorizontal	0	Off (Optional)
	1	On
iSplitPoint		Set Spatial Split point

Output: Received: Success
Received: Failure
Example: cv viewmode Mirror
cv viewmode mirror 0 10
Notes: bHorizontal and iSplitPoint are only available in Side/Seamless and Mirror modes



Playback

autoalign

Description: Autoalign aligns the sequence mapped to Viewport A to Viewport B, auto-align uses the currently selected frame in Viewport B for alignment

Syntax: autoalign bMaximizeAlignedLength bIntelligent

Input: **bMaximizeAlignedLength** OPTIONAL - **0** – Off
1 – On
bIntelligent OPTIONAL - **0** – Off
1 – On

Output: Received: Success
Received: Failure

Example: cv autoalign

Notes: bMaximizeAlignedLength is an option that will allow the sequenced to become the longest possible. If the you had the same sequence in viewport A that was in B and set the first frame in B to 10 and you aligned without this option you would get two sequences that ran from frame 10 to the end. If you check this option Clearview would first align them to 10 and then determine that there can be 10 more frames added to the beginning of each sequence and they would be frame 0 to the end.
bIntelligentAlignment enables intelligent alignment

configIntelligentAlign

Description: configIntelligentAlign configures intelligent alignment

Syntax: configIntelligentAlign bSpatial iSpatialX iSpatialY

Input: **bSpatial** **0** – Off
1 – On
iSpatialX Set spatial X offset range
iSpatialY Set spatial Y offset range

Output: Received: Success
Received: Failure

Example: cv configIntelligentAlign 4 4

Notes: NONE

first

Description: First moves the current frame of Clear View output to the first frame of both sequences

Syntax: First

Input: **NONE**

Output: Received: Success
Received: Failure

Example: cv first

Notes: none



fieldmode

Description: Fieldmode allows the user to change the field mode of either viewports

Syntax: Fieldmode <bViewport> <cFieldmode>

Input: **bViewport** **0** – Viewport A
 1 – Viewport B
cFieldmode **Frame** – Play entire frame
 F1 – Field 1 Only
 F2 – Field 2 Only
 F1F2 – F1 / F2

Output: Received: Success

 Received: Failure

Example: cv fieldmode 1 F2

Notes: none

goto

Description: GoTo will change the current position of the frame shown

Syntax: GoTo <eViewport> <iFrame>

Input: **eViewport** **0** – Viewport A
 1 – Viewport B
iFrame Any number that is within the total number of frames in the sequence

Output: Received: Success

 Received: Failure

Example: cv goto 1 10

Notes: Goto is much like moving the slider bar. That is if there is one file in Viewport A that goes from frame 0 – 100 and one in Viewport B that goes from frame 22 – 122 and one changes Viewport A to frame 20 (cv goto 20), Viewport B will change the current frame to 42, so that they both line up.

jogFwd

Description: JogFwd will jog forward a single frame on Clear View output

Syntax: JogFwd

Input: **NONE**

Output: Received: Success

 Received: Failure

Example: cv jogFwd

Notes: none

jogRev

Description: JogFwd will jog backwards a single frame on Clear View output

Syntax: JogFwd

Input: **NONE**

Output: Received: Success

 Received: Failure

Example: cv jogRev

Notes: none



last

Description: Last moves the current frame of Clear View to the last frame of both sequences
Syntax: Last
Input: **NONE**
Output: Received: Success
Received: Failure
Example: *cv last*
Notes: none

mapA

Description: MapA places a sequence into Viewport A
Syntax: mapA <cSequence> <iFirst> <iLast> <bForceFormatChange>
Input: **cSequence** Any sequence name that is the currently selected library
iFirst Set First frame
iLast Set Last Frame
bForceFormatChange **0** – Return error if video standard and image format do not match
1 – force ClearView to change to match video standard and image format
Output: Received: Success
Received: Failure
Example: *cv mapA "Impairments 1080 YCbCr 8-bit" 0 22 0*
Notes: If one omits the first AND last frame, the previous set first/last frame will be used
-1 in place of the first frame will set the first frame to 0
-1 in place of the last frame will set the last frame to the last frame in the sequence
If the sequence name has spaces the sequence name needs to be enclosed by double quotes

mapB

Description: MapB places a sequence into Viewport B
Syntax: mapB <cSequence> <iFirst> <iLast> <bForceFormatChange>
Input: **cSequence** Any sequence name that is the currently selected library
iFirst Set First frame
iLast Set Last Frame
bForceFormatChange **0** – Return error if video standard and image format do not match
1 – force ClearView to change to match video standard and image format
Output: Received: Success
Received: Failure
Example: *cv mapB "Impairments 1080 YCbCr 8-bit" -1 -1 1*
Notes: If one omits the first AND last frame, the previous set first/last frame will be used
-1 in place of the first frame will set the first frame to 0
-1 in place of the last frame will set the last frame to the last frame in the sequence
If the sequence name has spaces the sequence name needs to be enclosed by double quotes



pause

Description: Pause will pause Clear View output
Syntax: Pause
Input: **NONE**
Output: Received: Success
Received: Failure
Example: *cv pause*
Notes: none

play

Description: Play will begin playing Clear View output
Syntax: Play
Input: **NONE**
Output: Received: Success
Received: Failure
Example: *cv play*
Notes: none

playmode

Description: Playmode allows the user to change the play modes
Syntax: Playmode <cMode>
Input: **cMode** **Once** – Play Once
 Repeat – Repeat (Loop)
 Ping – Ping (fwd/bkwd)
 Alternate – Alternate A/B
Output: Received: Success
Received: Failure
Example: *cv playmode ping*
Notes: none

speed

Description: Speed changes the speed of the sequence on either of the viewports
Syntax: Speed <eViewport> <dSpeed>
Input: **eViewport** **0** – Viewport A
 1 – Viewport B
 dSpeed 0.00 to 2.00
Output: Received: Success
Received: Failure
Example: *cv speed 2 1.50*
Notes: none



stop

Description: Stop will stop Clear View output
Syntax: Stop
Input: **NONE**
Output: Received: Success
Received: Failure
Example: `cv stop`
Notes: none

unmapA

Description: unMapA removes a sequence from Viewport A
Syntax: mapA
Input: **NONE**
Output: Received: Success
Received: Failure
Example: `cv unmapA`
Notes: NONE

unmapB

Description: unMapB removes a sequence from Viewport B
Syntax: mapB
Input: **NONE**
Output: Received: Success
Received: Failure
Example: `cv unmapB`
Notes: NONE

Output

customVideoFormat

Description: CustomVideoFormat sets custom formats for DVI input
Syntax: CustomVideoFormat <iWidth> <iHeight> <iRate>
Input: **iWidth** Any width of a resolution already created
iHeight Any height of a resolution already created
iRate Any rate of a resolution already create
Output: Received: Success
Received: Failure
Example: `cv customVideoFormat 720 480 60`
Notes: **none**



dualOutput

Description: dualOutput will enable dualoutput
Syntax: DualOutput
Input: **NONE**
Output: Received: Success
Received: Failure
Example: cv DualOutput
Notes: none

outputHeight

Description: Returns the output height
Syntax: outputHeight
Input: **NONE**
Output: Received: Success
Received: Failure
Example: cv outputHeight
Notes: none

OutputRefresh

Description: Returns the output refresh rate
Syntax: outputRefresh
Input: **NONE**
Output: Received: Success
Received: Failure
Example: cv outputRefresh
Notes: none

outputWidth

Description: Returns the output width
Syntax: outputWidth
Input: **NONE**
Output: Received: Success
Received: Failure
Example: cv outputWidth
Notes: none



videoOutput

Description: VideoOutput will change the output device to the device specified

Syntax: VideoOutput <cOutputDevice>

Input: **cOutputDevice** **DVI** – DVI Output Module
broadcast – Broadcast Output Module
broadcast2 – Broadcast Output Module 2
none – No Video Output Module

Output: Received: Success
Received: Failure

Example: cv VideoOutput broadcast

Notes: none

Capture

inputHeight

Description: Returns the input height

Syntax: inputHeight

Input: **NONE**

Output: Received: Success
Received: Failure

Example: cv inputHeight

Notes: none

inputRefresh

Description: Returns the input refresh rate

Syntax: inputRefresh

Input: **NONE**

Output: Received: Success
Received: Failure

Example: cv inputRefresh

Notes: none

inputWidth

Description: Returns the input width

Syntax: inputWidth

Input: **NONE**

Output: Received: Success
Received: Failure

Example: cv inputWidth

Notes: none



videoInput

Description: VideoInput sets the input device to record
Syntax: videoInput <cInputDevice> <cRecordMode> <eInput> <eInputBoard> <cSourceFormat>

Input:

cInputDevice	broadcast – Broadcast Input module clearView – ClearView Output dvi – DVI Input Mode
cRecordMode	single – record a single input stream dual – record two input streams inOut – output a stream and input a stream
eInput	0 – logical input 0 1 – logical input 1
eInputBoard	0 – input board 0 1 – input board 1
cSourceFormat	<i>SDI Input Options</i> SDI – SDI Input 1 SDI2 – SDI Input 2 HDMI – HDMI (For single link configuration only) <i>Analog Input Options (Only for LH Configuration)</i> 525ComponentBetaUS – 525 Component Beta US 525ComponentSMPTEUS – 525 Component SMPTE US 525S-VideoUS – 525 S-Video US 525CompositeUS – 525 Composite US 525ComponentBetaJapan – 525 Component Beta Japan 525S-VideoJapan – 525 S-Video Japan 525CompositeJapan – 525 Composite Japan 625ComponentBeta – 625 Component Beta 625ComponentSMPTE – 625 Component SMPTE 625S-Video – 625 S-Video 625Composite
cAudioInput	SDI – SDI HDMI – HDMI (For single link configuration only) AES – AES Analog – Analog None – None
cSyncSource	OPTIONAL Ext – External SDI – SDI Input1 SDI2 – SDI Input 2 HDMI – HDMI (For single link configuration only) FREE – Free Run

Output: Received: Success
Received: Failure

Example: cv videoInput clearview
cv videoInput broadcast single 0 0 SDI None
cv videoInput broadcast dual 0 0 SDI2 None
cv videoInput DVI

Notes: Analog Options can only be used with the LH board



When doing a dual record you must run `videoInput` twice, first setting logical input 0 then logical input 1

The logical input should always be 0, except when doing a dual input. It is important to note that the logical input is not the same as the SDI input. Go to the broadcast tab, if you are in input mode there is only Input 1 shown (logical input 0). If you go to dual input you have Input 1 (logical input 0) as well as Input 2 (logical input 1). When doing a dual input you will need to run `cv videoinput broadcast` twice, once configuring logical input 0 and a second time configuring logical input 1.

record

Description: Record will begin to record for the current input source. (There are two options for this command)

Syntax: `Record <cLibrary> <cSeqName> <iNumFrames> <bAbortOnDrop> <btoMemory>`
`Record <cLibrary> <cSeqName> <cLibrary> <cSeqName> <iNumFrames>`
`<bAbortOnDrop>`

Input:

- cLibrary** Library path to record to
- cSeqName** A sequence name
- iNumFrames** The number of frames you want to record
- bAbortOnDrop** **0** – Off, no notification of a dropped frame
1 – On, notification of a dropped frame
- btoMemory** **1** – Off, save to disk
0 – On, save to memory

Output: Received: Success
Received: Failure

Example: `cv record "F:\Gold" "Football Gold" 100 1 0`
`cv record "F:\Encoder1" BasketballHD1 "F:\Encoder2" BasketballHD2 100 0`

Notes: If the sequence name has spaces the sequence name needs to be enclosed by double quotes
There are two record commands the first one listed as well as the first example is an example if doing a single input. The second one listed and second example is if doing a dual input.



Metrics

audioMetricFreq

Description: audioMetricFreq will run aFreq

Syntax: audioMetricFreq <LogFileName> <channelsA> <channelsB> <bAlign> <bNormalize>
<bThreshold>

Input:

LogFileName	Path to place the log file and name
channelsA	1,2,..8, one, two four, eight – Audio channels from Viewport A
channelsB	1,2,..8, one, two four, eight – Audio channels from Viewport B
bAlign	OPTIONAL - aligns the two channels before running the metric 0 – Off 1 – On
bNormalize	OPTIONAL 0 – Off, normalize information not used 1 – On, normalize information used
bThreshold	OPTIONAL – A threshold in which if the audio metric result is below this number the frame is considered bad and added to return number

Output: Received: Success: Failures = 0, Sequence Metric Value = 100.00, Audio Alignment offset = 0 frames = 0 samples = 0.00 ms

Received: Failure

Example: cv audioMetricFreq C:\Log 1 1 1 – this will compare channel 1 from Viewport A to channel 1 from Viewport B
cv audioMetricFreq C:\Log two two 1 – this will compare channel 1&2 from Viewport A to channel 1&2 from Viewport B

Notes: One can only run a single channel at a time when using number characters
If you want to run multiple channels at a time you will use words (four will check 1-4)
You cannot do more than one option at a time one character or one word
If you choose more than a single channel of audio the audio metric value is the average of all channels.

audioMetricPEAQ

Description: audioMetricPEAQ will run PEAQ

Syntax: audioMetricPEAQ <LogFileName> <channelsA> <channelsB> <bAlign> <bNormalize>
<bThreshold>

Input:

LogFileName	Path to place the log file and name
channelsA	1,2,..8, one, two four, eight – Audio channels from Viewport A
channelsB	1,2,..8, one, two four, eight – Audio channels from Viewport B
bAlign	OPTIONAL - aligns the two channels before running the metric 0 – Off 1 – On
bNormalize	OPTIONAL 0 – Off, normalize information not used 1 – On, normalize information used
bThreshold	OPTIONAL – A threshold in which if the audio metric result is below this number the frame is considered bad and added to return number

Output: Received: Success: Failures = 0, Sequence Metric Value = 100.00, Audio Alignment offset = 0 frames = 0 samples = 0.00 ms



Example: Received: Failure
cv audioMetricFreq C:\Log 1 1 1 – this will compare channel 1 from Viewport A to channel 1 from Viewport B
cv audioMetricFreq C:\Log two two 1 – this will compare channel 1&2 from Viewport A to channel 1&2 from Viewport B

Notes: One can only run a single channel at a time when using number characters
If you want to run multiple channels at a time you will use words (four will check 1-4)
You cannot do more than one option at a time one character or one word
If you choose more than a single channel of audio the audio metric value is the average of all channels.

configaFreq

Description: configaFreq will configure the aFreq metric
Syntax: configaFreq <iBatchMilliseconds> <iLowPassThreshold> <fSilenceThreshold>
Input: **iBatchMilliseconds** Value of number of Milliseconds aFreq uses to calculate
iLowPassThreshold Threshold to ignore low frequencies
fSilenceThreshold Threshold to ignore silence

Output: Received: Success
Received: Failure

Example: cv configaFreq 334 0 .0020

Notes: iBatchMilliseconds default is 334
iLowPassThreshold default is 0
fSilenceThreshold is 0.0020

configPEAQ

Description: configPEAQ will configure the PEAQ metric
Syntax: configPEAQ <cScale> <fSilenceThreshold> <bRefOnB>
Input: **cScale** **PEAQ** – PEAQ BS. 1387 (-4 to 0) Scale
BS1116 – BS. 1116 (1 to 5) Scale
fSilenceThreshold Threshold to ignore low frequencies
bRefOnB **0** – Off, reference on A
1 – On, reference on B

Output: Received: Success
Received: Failure

Example: cv configPEAQ PEAQ .0020 0

Notes: fSilenceThreshold is 0.0020



configDMOS

Description: configDMOS will configure the DMOS metric
Syntax: configDMOS <cFieldMode> <eMaxValue>
Input: **cFieldMode** **Field** – field mode not set
Frame – frame mode set
Auto – frame mode will run for progressive formats, field for interlace formats
eMaxValue **2** – 2
7 – 7
10 – 10
Output: Received: Success
Received: Failure
Example: cv configDMOS Auto
Notes: none

configPSNR

Description: ConfigPSNR will configure the PSNR metric
Syntax: configPSNR <bLimitNumerator>
Input: **bLimitNumerator** **0** – Off, do not limit the numerator
1 – On, limit the numerator
Output: Received: Success
Received: Failure
Example: cv configPSNR 0
Notes: bLimitNumerator limits the numerator in the PSNR metric. When off the numerator is from 0 -255 for all three components. If turned on, the numerator is limited to 16 – 235 for Luma and 16 – 240 for Chroma. These are 8 bit numbers, for 10 bit the ranges are a multiple of 4.

configJND

Description: ConfigJND will configure the JND metric
Syntax: configJND <eViewDist> <cFieldMode> <cDeinterlace> <cColorModeling>
<iMaxDisplayLuminance> <bOutputJndMaps> <cMapLocation>
Input: **eViewDist** **2** – 2
3 – 3
4 – 4
5 – 5
6 – 6
cFieldMode **Frame** – Frame
Field – Field
Auto – Auto
cDeinterlace **Average** – Average
Duplicate – Duplicate
Interpolate – Interpolate
Median – Media
cColorModeling **SMPTE274M** – SMPTE 274M
SMPTE240M – SMPTE 240M
EBU625 – EBU-625
iMaxDisplayLuminance Value 1 - 1000



bOutputJndMaps **0** – Off, do not output JND maps
 1 – On, output JND maps
 OPTIONAL – Path to place the JND Maps

cMapLocation
Received: Success
Received: Failure

Output: cv configJND 5 Auto Average SMPTE274M 70 0

Example: The above setting is the default setting for JND.

Notes: eViewDistan – is how far the human tester was standing away from the display
 cFieldMode – this should be set to Auto which allows the system to do the right thing. For interlaced video sequences, the field mode should be field. For progressive video sequences, the field mode should be frame. You can override this.
 cDeinterlace Flag - If you are calculating interlaced data in frame mode, then you need to choose a de-interlace method. In general, you should calculate fields in JND field mode.
 cColorModelingFlag - This is automatically set based on whether we are in HD or SD. This is an override.
 iMaxDisplayLuminance - This is the luminance of the viewed display.
 cMapLocation – This is where the map files are stored on the hard disk array. The files are viewed using a command line program called vpseqw32.exe.

dmos

Description: DMOS will perform a DMOS on the currently loaded sequences

Syntax: dmos <LogFileName> <bChroma> <dThresholdY> <bSpatialize > <bNormalize>

Input: **LogFileName** Path to place the log file and name
 bChroma OPTIONAL – A threshold in which if the mos result is above this number the frame is considered bad and added to return number
 dThresholdY OPTIONAL – A threshold in which if the mos result is above this number the frame is considered bad and added to return number
 bSpatialize OPTIONAL
 0 – Off, spatial information not used
 1 – On, spatial information used
 bNormalize OPTIONAL
 0 – Off, normalize information not used
 1 – On, normalize information used

Output: Received: Success: Threshold Failures = <Number of Failed Frames>
 Received: Failure

Example: cv dmos "C:\Football Impairments.dmos"

Notes: One should start any metric from Stop mode in Clear View
 If one omits the threshold value, the number of failed frames will always be 0
 If the log filename has spaces the log file needs to be enclosed by double quotes
 If the log file does not contain a path, the Clearview.exe path is used
 To turn on Chroma a value must be set for dThresholdChroma (even if it is 10)



jnd

Description: JND will perform a JND on the currently loaded sequences
Syntax: `jnd <LogFileName> <dThresholdY> <dThresholdChroma> <bSpatialize> <bNormalize>`

Input: **LogFileName** Path to place the log file and name
dThresholdY OPTIONAL – A threshold in which if the jnd result is above this number the frame is considered bad and added to return number
dThresholdChroma OPTIONAL – A threshold in which if the jnd result is above this number the frame is considered bad and added to return number
bSpatialize OPTIONAL
0 – Off, spatial information not used
1 – On, spatial information used
bNormalize OPTIONAL
0 – Off, normalize information not used
1 – On, normalize information used

Output: Received: Success: Threshold Failures = <Number of Failed Frames>
Received: Failure

Example: `cv jnd "C:\Football Impairments.jnd"`

Notes: One should start any metric from Stop mode in Clear View
If one omits the threshold value, the number of failed frames will always be 0
If the log filename has spaces the log file needs to be enclosed by double quotes
If the log file does not contain a path, the Clearview.exe path is used
To turn on Chroma a value must be set for dThresholdChroma (even if it is 100)

normalize

Description: normalizes viewport A to B

Syntax: `normalize`

Input: **NONE**

Output: Received: Success

Received: Failure

Example: `cv normalize`

Notes: none

normalizeOffsets

Description: sets the normalize offset

Syntax: `normalizeOffsets <iY> <iCb> <iCr>`

Input: **iY** y offset

iCb cb offset

iCr cr offset

Output: Received: Success

Received: Failure

Example: `cv normalizeOffset 3 3 52`

Notes: none



spatialAlign

Description: aligns viewport A and B spatially
Syntax: spatialAlign
Input: **NONE**
Output: Received: Success
Received: Failure
Example: *cv spatialAlign*
Notes: none

spatialOffsets

Description: sets the spatial offset
Syntax: spatialOffsets <iX> <iY>
Input: **iX** x offset
iY y offset
Output: Received: Success
Received: Failure
Example: *cv spatialOffsets 3 0*
Notes: none

metricWindow

Description: sets the window in which to perform a metric
Syntax: metricWindow <iX> <iY> <iW> <iH>
Input: **iX** x value for the left point
iY y value for the left point
iW total width of the window
iH total height of the window
Output: Received: Success
Received: Failure
Example: *cv metricWindow 3 3 1920 1080*
Notes: none

psnr

Description: PSNR will perform a PSNR on the currently loaded sequences
Syntax: PSNR <cLogName> <bThresholdY> <bThresholdCb> <bThresholdCr> <bNoRef>
<bSpatialize> <bNormalize>
Input: **cLogName** Path to place the log file and name
bThresholdY OPTIONAL – A threshold in which if the PSNR result for this component is below this number the frame is considered bad and added to return number
bThresholdCb OPTIONAL – A threshold in which if the PSNR result for this component is below this number the frame is considered bad and added to return number
bThresholdCr OPTIONAL – A threshold in which if the PSNR result for this component is below this number the frame is considered bad and added to return number
bNoRef **0** – Off, referenced used
1 – On, no referenced used



bSpatialize 0 – Off, spatial information not used
1 – On, spatial information used
bNormalize 0 – Off, normalize information not used
1 – On, normalize information used

Output: Received: Success: Threshold Failures = <Number of Failed Frames>
Received: Failure

Example: cv psnr "C:\Log.psnr" 95 95 95 1 0 0

Notes: One should start any metric from Stop mode in Clear View
If one omits the threshold value, the number of failed frames will always be 0
If the log filename has spaces the log file needs to be enclosed by double quotes
Threshold values must be used if you would like to use NoRef, Spatalize or Normalize
Values
If the log file does not contain a path, the Clearview exe path is used

spatial

Description: spatial will perform a spatial metric on the currently loaded sequences
Syntax: Spatial <cLogName> <bThresholdY> <bThresholdCb> <bThresholdCr> <bNoRef>
<bSpatalize> <bNormalize>

Input: **cLogName** Path to place the log file and name
bThresholdY OPTIONAL – A threshold in which if the spatial result for this component is above this number the frame is considered bad and added to return number
bThresholdCb OPTIONAL – A threshold in which if the spatial result for this component is above this number the frame is considered bad and added to return number
bThresholdCr OPTIONAL – A threshold in which if the spatial result for this component is above this number the frame is considered bad and added to return number

Output: Received: Success: Threshold Failures = <Number of Failed Frames>
Received: Failure

Example: cv spaital C:\Log.spatial

Notes: One should start any metric from Stop mode in Clear View
If one omits the threshold value, the number of failed frames will always be 0
If the log filename has spaces the log file needs to be enclosed by double quotes
If the log file does not contain a path, the Clearview exe path is used



temporal

Description: temporal will perform a temporal metric on the currently loaded sequences

Syntax: temporal <LogName> <bThresholdY> <bThresholdCb> <bThresholdCr>

- Input:**
- LogName** Path to place the log file and name
 - bThresholdY** OPTIONAL – A threshold in which if the temporal result for this component is above this number the frame is considered bad and added to return number
 - bThresholdCb** OPTIONAL – A threshold in which if the temporal result for this component is above this number the frame is considered bad and added to return number
 - bThresholdCr** OPTIONAL – A threshold in which if the temporal result for this component is above this number the frame is considered bad and added to return number

Output: Received: Success: Threshold Failures = <Number of Failed Frames>
Received: Failure

Example: cv temporal "C:\Football Impairments.temporal"

Notes: One should start any metric from Stop mode in Clear View
If one omits the threshold value, the number of failed frames will always be 0
If the log filename has spaces the log file needs to be enclosed by double quotes
If the log file does not contain a path, the Clearview exe path is used

6. Errors

Not recognized command

Error: 'cv' is not recognized as an internal or external command, operable program or batch file
Reason: This occurs as cv is not located in the system's path
Fix:
1. Add Clear View to the path (see Setup)
2. Run the cv command from the Clear View folder

connect() failed

Error: connect() failed: 10061
Reason: This occurs when CVServer is not running
Fix: Open a second dos window and run "CVServer 7", this will open a connection between CVServer and Clear View witch passes command through port 7 of the machine. Assure that you run CVServer, while it is your systems path or from C:\Program Files\Video Clarity\ClearView\.

Error opening config file

Error: error opening config file for read access.connect() failed: 10060
Reason: This occurs when the config file is missing or not correct
Fix: Assure there is a config file located at C:\Program Files\Video Clarity\ClearView\. This is a text file without an extension. The conents of this file should be your ip address, 192.168.1.1.