Cyviz Easy Controller

API commands

Version 2.19

Table of Contents

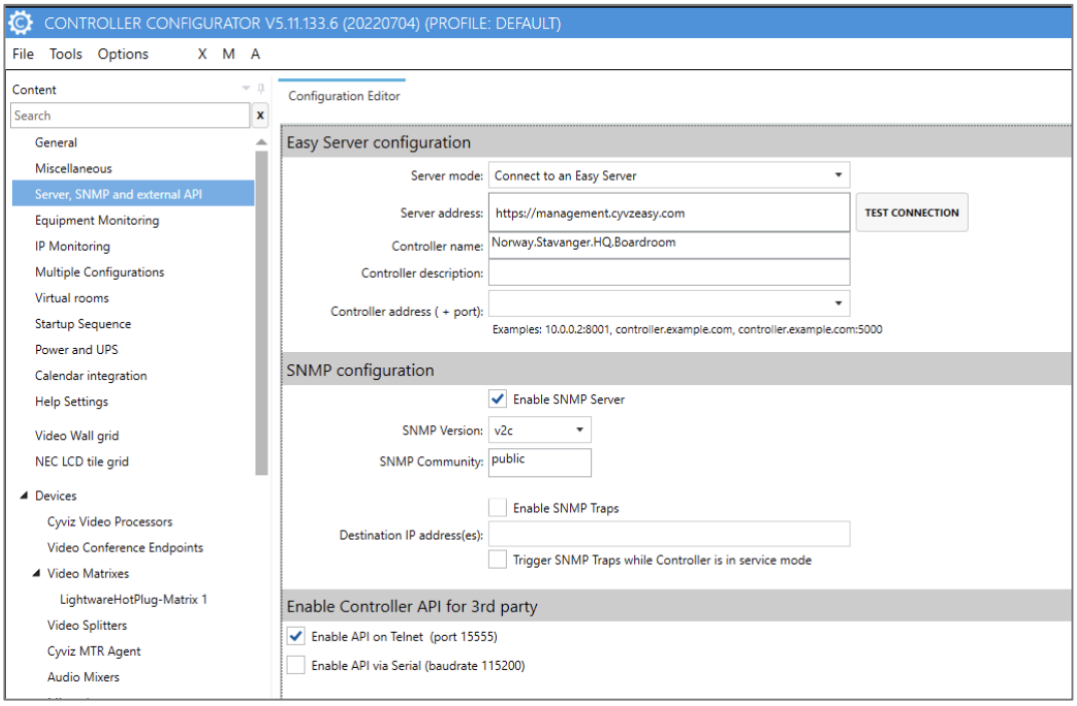
1 Cyviz Easy Controller – 3rd party API

1.1 Introduction

The Cyviz Easy Controller offers an ASCII API over IP for interaction with other control system.

1.2 Enabling the telnet API on a Controller

In the Controller Configuration Tool, under section “Server, SNMP and external API”, check the option named “Enable API on Telnet” under section “Enable Controller API for 3rd party”.



The API is available on port 15555 when enabled.

To open port 15555 in the Firewall on the Controller:

- Exit the Controller and enter the AdminTool.
- Start a Windows terminal as Administrator and right click on cmd.exe in windows\system32 and choose “Run as administrator”.

Enter the following command:

```
netsh advfirewall firewall add rule name=CDCExternalAPI dir=in action=allow
protocol=TCP localport=15555
```

1.3 Easy Controller Version

The API described in this document is available from version 5.11.142

1.4 General guidelines

Avoid calling the API with more that 2 calls per second.

2 Command overview

2.1 System

2.1.1 Get power status

Returns power status as ON or OFF.

Syntax

```
power status
```

Response

```
Power is off
```

2.1.2 Turn power on

Turns the display wall ON.

Syntax

```
power on
```

Response

```
Power is on
```

2.1.3 Turn power off

Turns the display wall OFF.

Syntax

```
power off
```

Response

```
Power is off
```

2.1.4 Clean system

Cleans (resets) the system for the next session. This will clear the display wall, clear targets, hang up VC calls and return to the Controller start page.

Syntax

```
system clean
```

Response

```
System is cleaned.
```

2.2 Display wall

2.2.1 Set primary source

Activate primary source. Id 0 is used for no source.

Syntax

```
primary source <Name or id of primary source>
```

Response

```
Primary source <Name of primary source> loaded.
```

or

```
ERROR:Could not find primary source named <Name of primary source>
```

Example

```
primary source Windows10
```

Response

```
Primary source Windows10 loaded.
```

2.2.2 Get stereo status

Returns status of stereo mode as ON or OFF.

Syntax

```
stereo status
```

Response

```
stereo is off|on
```

2.2.3 Turn stereo on

Turns stereo mode ON.

Syntax

```
stereo on
```

Response

```
stereo is on
```

2.2.4 Turn stereo off

Turns stereo mode OFF.

Syntax

```
stereo off
```

Response

```
stereo is off
```

2.2.5 Get list of PiPs on display wall

Will return a list of PiPs shown on the display wall.

The PiP Id will reflect the sequence of when the PiP was added to the video wall. If a PiP is changed by the intelligent PiP handler, the sequence (and the Id) will change.

Syntax

```
getpiplist
```

Response

```
*r start - getpiplist
*r amount of pips: <quantity>
*r pip <id>, <pos X>, <pos Y>, <width>, <height>, <visible|hidden>, <source name>
*r end
```

Example

```
getpiplist
```

Response

```
*r start - getpiplist
*r amount of pips: 2
*r pip 1, 0, 0, 1920, 1080, visible, ClickShare
*r pip 2, 1920, 0, 1280, 768, hidden, Workstation
*r end
```

2.2.6 Get list of PiPs on display wall v2

Will return a list of PiPs shown on the display wall, where the source is identified by its external name.

```
getpiplistv2
```

Response

```
*r start – getpiplistv2
*r amount of pips: <quantity>
*r pip <id>, <pos X>, <pos Y>, <width>, <height>, <visible|hidden>, <external
source name>
*r end
```

Example

```
getpiplistv2
```

Response

```
*r start – getpiplistv2
*r amount of pips: 2
*r pip 1, 0, 0, 1920, 1080, visible, cs1
*r pip 2, 1920, 0, 1280, 768, hidden, ws2
*r end
```

2.2.7 Replace PiP source and change size and position

Will change content, size and position of a PiP identified by PiP Id returned from **`getpiplist`**.

Syntax

`replacepip <id>, <name of source>, [<pos X>, < pos Y>, <width>, <height>]`

- **`<id>`** `Id of PiP, which is returned in getpiplist`
- **`<name of source>`** `Name of Source to replace current source`
- **`<pos X>`** `Position of PiP in X coordinate`
- **`<pos Y>`** `Position of PiP in Y coordinate`
- **`<width>`** `PiP Width`
- **`<height>`** `PiP Height`

Size and position are optional, when not specified, the PiP will inherit size & position from existing PiP. Position 0,0 is upper left corner. Aspect of PiP will not adjust to aspect of video source and content will be stretch to fit PiP size.

Response

```
*r start - replacepip
*r replacepip success
*r end
*e <additional information>
```

or

```
*r start - replacepip
*r replacepip error (<error message>)
*r end
```

Example

```
replacepip 1, workstation, 0, 0, 1280, 1024
```

or

```
replacepip 1, workstation
```

2.2.8 Replace PiP source and change size and position v2

Will change content, size and position of a PiP identified by PiP Id returned from **`getpiplist,`** and the source is identified by its external name.

Syntax

```
replacepipv2 <id>, <external source name>, [<pos X>, < pos Y>, <width>, <height>]
```

Response

```
*r start – replacepipv2
*r replacepipv2 success
*r end
*e <additional information>
```

or

```
*r start – replacepipv2
*r replacepipv2 error (<error message>)
*r end
```

Example

```
Replacepipv2 1, ws2, 0, 0, 1280, 1024
```

or

```
replacepipv2 1, ws2
```

2.2.9 Set visibility of a PiPs

Will set the visibility of a PiP identified by the Id.

Syntax

```
setpipvisibile <id>, <1|0>
```

```
1 for visible
0 for hidden
```

Response

```
*r start - setpipvisibile
Sat pip <id>visibility to <True|False>
*r end
```

Example

```
setpipvisibile 2, 0
```

Response

```
*r start - setpipvisibile
Sat PiP 1visibility to False
*r end
```

2.3 Sources

2.3.1 Get list of sources

Will return a list over secondary sources, with a GUID, name and confidentially level.

- The source name can be used in combination with the "**replacepip**" function.
    - The source GUID can be used in combination with the "**source set**" function.
    -

Syntax

```
getsourcelist
```

Response

```
*r start - getsourcelist
*r Name: "NUC#1 - Left" GUID: "10d817e1-ded6-41a9-b645-cfa254878114"
Confidentiality: 0
*r Name: "NUC#1 - Right" GUID: "ed5619cc-8580-4056-9242-0e67a3c82bbf"
Confidentiality: 0
*r end
```

2.3.2 Get list of sources v2

Will return a list over secondary sources, with the GUID, name, external name and confidentially level.

- The source name and external name can be used in combination with the "**replacepip**" function.
- The source GUID can be used in combination with the "**source set**" function.

```
getsourcelistv2
```

Response

```
*r start – getsourcelistv2
*r Name: "NUC#1 - Left" Confidentiality: 0 GUID: "10d817e1-ded6-41a9-b645-
cfa254878114" Confidentiality: 0 ExtName: "OpStatOneLeft"

*r Name: "NUC#1 - Right" Confidentiality: 0 GUID: "10d817e1-ded6-41a9-b645-
cfa254837614" Confidentiality: 1 ExtlName: "OpStatOneRight"

*r end
```

2.3.3 Change source

Will change the name and/or the confidentially level of the secondary source identified by the GUID.

Syntax

```
source set GUID: "<GUID>" Name:"<New name>"
```

or

```
source set GUID: "<GUID>" ConfidentialityLevel:<New level>
```

or

```
source set GUID: "<GUID>" Name:"<New name>" ConfidentialityLevel:<New level>
```

Response

```
*r start - source set name
*r source set name success
*r end
```

and/or

```
*r start - source set name
*r source set name error (No videosource with "GUID")
*r end
```

Example

```
source set GUID: "5d96cf53-7691-4cf2-9e70-b4c334f16cda" Name:"MapManager PC"
```

Response

```
*r start - source set name
*r source set name success
*r end
```

Example

```
source set GUID: "5d96cf53-7691-4cf2-9e70-b4c334f16cda" ConfidentialityLevel:2
```

Response

```
*r start - source set confidentiality
*r source set confidentiality success
*r end
```

Example

```
source set GUID: "5d96cf53-7691-4cf2-9e70-b4c334f16cda" Name:"Secret PC"
ConfidentialityLevel:2
```

Response

```
*r start - source set confidentiality
*r source set confidentiality success
*r end
*r start - source set name
*r source set name success
*r end
```

2.4 Targets

2.4.1 Get list of targets

Returns a list of targets identified with the GUID.

Syntax

```
gettargetlist
```

Response

```
*r start - gettargetlist
*r Room: "<Room name>" Name: "<Target name>" GUID: "<Target GUID>"
…
*r end
```

2.4.2 Get list of targets v2

Returns a list of targets with a set of target identifiers and information about the source routed to the target.

The target external name and the source external name are defined in the Easy Controller configuration.

Syntax

```
gettargetlistv2
```

Response

```
*r start – gettargetlistv2
*r Room: "<Room name>" Name: "<Target name>" GUID: "<Target GUID>" ExtName:
"<External Target Name>" SourceName: "<Source Name>" SourceGUID: "<Source GUID>"
SourceExtName: "<External Source Name>"
…
*r end
```

2.4.3 Clear target

Returns a list of targets.

Syntax

```
target clear "<TARGET GUID>"
or
target clear ExtName: "<External Target Name>"
```

Response

```
*r start - target clear
*r target clear success
*r end
```

or

```
*r start - target clear
*r target clear error (No target with GUID <specified target guid>)
*r end
```

Example

```
target clear "10d817e1-ded6-41a9-b645-cfa254878114"
```

Response

```
*r start - target clear
*r target clear success
*r end
```

2.4.4 Route a source to a target

Route a source to a target, both identified by their GUIDs.

Syntax

```
target route source: "<Source GUID>" target: "<Target GUID>"
or
target route sourceExtName: "<External Source Name>" target: "<TARGET GUID>"
or
target route sourceExtName: "<External Source Name>" targetExtName: "<External
Target Name>"
```

Response

```
*r start - target route
*r target route success
*r end
```

or

```
*r start - target route
*r target route error (No source with GUID "<SOURCE GUID<" and/or no target with
GUID "<TARGET GUID>")
*r end
```

Example

```
target route source: "3a8ac903-144e-4d11-be83-ba77bfbf62d5" target: "10d817e1-ded6-
41a9-b645-cfa254878114"
```

Response

```
*r start - target route
*r target route success
*r end
```

2.5 Audio

2.5.1 Get master audio level

Get the master audio level, values 0 – 100.

Syntax

```
audio master
```

Response

```
*r start - Audio Master Level
*r Level: <current level>
*r end
```

Example

```
audio master
```

Response

```
*r start - Audio Master Level
*r Level: 50
*r end
```

2.5.2 Set master audio level

Set the master audio level, values 0 – 100.

Syntax

```
audio master level:<level>
```

Response

```
*r start – Set Audio Master Level
*r Level: <current level>
*r end
```

Example

```
Audio master level: 30
```

Response

```
*r start – Set Audio Master Level
*r Level: 30
*r end
```

2.5.3 Mute master audio

Mutes the master volume

Syntax

```
mute master on
```

Response

```
Master volume muted.
*e AudioPort ID: "7473614d-7265-6262-4253-53426c752031" Muted: True
```

2.5.4 Unmute master audio

Unmutes the master volume

Syntax

```
mute master off
```

Response

```
Master volume unmuted.
*e AudioPort ID: "7473614d-7265-6262-4253-53426c752031" Muted: False
```

2.5.5 Get master audio mute state

Get the master volume mute state.

Syntax

```
getmastermute
```

Response

```
Master volume unmuted|muted
*e AudioPort ID: "7473614d-7265-6262-4253-53426c752031" Muted: False|True
```

2.5.6 Get PiP audio status

Will return the audio status of the PiPs shown on the display wall.

Syntax

```
getpipaudio
```

Response

```
*r start - getpipaudio
*r amount of pips: <quantity>
*r pip <id>, <muted | unmuted>
*r end
```

Example

```
getpipaudio
```

Response

```
*r start - getpipaudio
*r amount of pips: 2
*r pip 1, muted
*r pip 2, unmuted
*r end
```

2.5.7 Mute PiP audio

Mutes the audio on a PiP.

Syntax

```
mute pip <pip Id> on
```

Response

```
pip <pip Id> audio muted.
*e AudioPort ID: "<audio port GUID>" Muted: True
```

Example

```
Mute pip 2 on
```

Response

```
pip 2 audio muted.
*e AudioPort ID: "7473614d-7265-6262-8732-ad426c7520ef" Muted: True
```

2.5.8 Unmute PiP audio

Unmutes the audio on a PiP.

Syntax

```
mute pip <pip Id> off
```

Response

```
pip <pip Id> audio unmuted.
*e AudioPort ID: "<audio port GUID>" Muted: False
```

Example

```
Mute pip 2 off
```

Response

```
pip 2 volume unmuted.
*e AudioPort ID: "7473614d-7265-6262-8732-ad426c7520ef" Muted: False
```

2.5.9 Mute the microphone

Mute the microphone.

Syntax

```
mute mic on
```

Response

```
mute mic is on
```

2.5.10 Unmute the microphone

Unmutes the microphone.

Syntax

```
mute mic off
```

Response

```
mute mic is off
```

2.5.11 Get list of audio inputs

Will return list of all audio inputs

Syntax

```
getlistofaudioinputs
```

Response

```
*r start - getlistofaudioinputs
*r Audio port - id: <input source id 1> - name: <input source name 1>
*r Audio port - id: <input source id 2> - name: <input source name 2>
*r end
```

Example

```
getlistofaudioinputs
```

Response

```
*r start - getlistofaudioinputs
*r Audio port - id: 08a7418c-dbe7-af3d-6d17-622648b56816 - name: Main Source
*r Audio port - id: 55ebdgd6-fa11-fbbd-8e60-e7fb0a548cdb - name: Room 1
*r Audio port - id: 36ecfhd6- 1w25-fbbd-8e60-e7fb0a54848b - name: Room 2
*r end
```

2.5.12 Get list of audio outputs

Will return list of all audio outputs

Syntax
```
getlistofaudiooutputs
```

Response
```
*r start - getlistofaudiooutputs
*r Audio port - id: <input source id 1> - name: <input source name 1>
*r Audio port - id: <input source id 2> - name: <input source name 2>
*r end
```

Example
```
getlistofaudioinputs
```

Response
```
*r start – getlistofaudiooutputs
*r Audio port - id: 7453644c-7365-6262-4253-52326c752021 - name: Master
*r end
```

2.5.13 Set audio level

Set the input/output audio level, values 0 – 100.

Syntax
```
Audioport Name:"<audio input/output name>|ID:<audio input/output id>" Level:<level>
```
Response
```
*r AudioPort ID:"<audio name>|ID:<audio id>" Level:<level>
```

Example
```
Audioport ID:"05b5418b-tcq3-af6d-4w19-312648b52816" Level:25
```

Response
```
*r AudioPort ID: "05b5418b-tcq3-af6d-4w19-312648b52816" Level: 25
```

2.5.14 Get audio level

Get the input/output audio level.

Syntax
```
Audioport Name:"<audio input/output name>|ID:<audio input/output id>" Level
```
Response
```
*r AudioPort ID:"<audio name>|ID:<audio id>" Level:<level>
```

Example
```
Audioport ID:"05b5418b-tcq3-af6d-4w19-312648b52816" Level
```

Response
```
*r AudioPort ID: "05b5418b-tcq3-af6d-4w19-312648b52816" Level: 25
```

2.6 Light control

2.6.1 Get light preset list

Returns list of stored light presets

Syntax

```
light preset list
```

2.6.2 Load light preset by Name or ID

Load light preset by its preset name or by its id.

Syntax

```
light preset recall ID: "<ID>"
```

or

```
light preset recall Name: "<Name>"
```

2.6.3 Load the default light preset

Loads “default” preset as defined in light events in the Controller preferences

Syntax

```
light preset default
```

2.6.4 Load the light preset used when display wall is on

Loads the “displayon” preset as defined in light events in the Controller preferences.

Syntax

```
light preset displayon
```

2.7 Video conference

2.7.1 Set the outgoing video source

Change the outgoing main video channel from codec

Syntax

```
outgoing main <name of source>
```

Response

```
Source <name of source> set as outgoing main.
```

or

```
Valid source names: <list of valid sources>
```

Example

```
outgoing main front camera
```

Response

```
Source front camera set as outgoing main.
```

2.7.2 Load a camera preset

Load camera a preset defined in the Cyviz Easy Controller.

Syntax

```
load camera preset <cameranumber,presetname>
```

Response

```
Camera preset loaded
```

or

```
No such preset and/or camera
```

Example

```
load camera preset 1,front
```

Response

```
Camera preset loaded
```

2.7.3 Get video conference status

Get information about video conference.

Syntax
`getvcstatus`

Response
`Undefined`
or
`NotActive`
or
`Active`
or
`ActiveAndSharing`

Example
`getvcstatus`

Response
`Active`

2.7.4 Recall camera preset and change outgoing camera source

Change outgoing camera source and recall a saved camera preset with single command

Syntax
Outgoing camera with preset <camera number>,<preset name>

Response
Camera preset loaded

Source <camera name> set as outgoing main.

Example
Outgoing camera with preset 1,mypreset

Response
Camera preset loaded

Source MyCamera set as outgoing main.

2.8 Presets

2.8.1 Get list of Presets

Will return a list over available presets, these preset names can be used by the **`load preset`** functions.

Syntax

```
getpresetlist
```

Response

```
*r start - getpresetlist
*r Morning meeting
*r Overview Sea
*r end
```

2.8.2 Get list of Presets v2

Will return a list over available presets, with GUID and external name.

The external name can be defined in the Controller Preferences page.

Syntax

```
getpresetlistv2
```

Response

```
*r start - getpresetlistv2
*r Name: "Morning meeting" GUID: "10d817e1-ded6-41a9-b645-gfc254878114"
ExternalName: "MornMeeting1"
*r Name: "Overview Sea" GUID: "10d818e1-ced6-41a9-b647-gfc254878184" ExtName:
"OverviewSea"
*r end
```

2.8.3 Load a preset

Will load first preset which contains a name with closest match to the name given, or exact match of GUID or external preset name.

There are two implementations: **`load preset`** and **`cleanandloadpreset`**

`cleanandloadpreset` will remove all PiPs before the preset is loaded which will ensure that the PiP Ids returned by the **`getpiplist`** reflects the sequence of when the Preset PiPs where added.

Syntax

```
cleanandloadpreset <name of preset>
or
cleanandloadpreset GUID: "<preset GUID>"
or
cleanandloadpreset ExtName: "<external preset name>"


load preset <name of preset>
or
load preset GUID: "<preset GUID>"
or
load preset ExtName: "<external preset name>"
```

Response

```
*r start - load preset
*r load preset success (Preset <Name of preset> loaded.)
*r end
```

or

```
*r start - load preset
*r load preset error (<Could not find preset named <Preset specified>)
*r end
```

Example

```
load preset Overview Sea
```

Response

```
*r start - load preset
*r load preset success (Preset Overview Sea loaded.)
*r end
*e AudioPort ID: "6f656469-4d20-5854-4253-53426c752031" Muted: False
*e Preset Current Active: True
*e Preset Current Name: "Overview Sea"
*e Preset Current Number: 1
```

2.8.4 Get active preset

Will return the name of the active preset.

Syntax

```
preset active
```

Response

```
*r start - preset active
*r Preset Current Active: False
*r Preset Current Number: -1
*r Preset Current Name: ""
*r end
```

or

```
*r start - preset active
*r Preset Current Active: True
*r Preset Current Number: 1
*r Preset Current Name: "Overview Sea"
*r end
```

2.8.5 Get active preset v2

Will return the name, GUID and external name of the active preset.

Syntax

```
preset activev2
```

Response

```
*r start - preset activev2
*r Preset Current Active: True
*r Preset Current Number: 1
*r Preset Current Name: "Overview Sea"
*r Preset Current GUID: "e3078b2e-2fdd-4412-84b9-27e4cc6e3c3f"
*r Preset Current ExternalName: "OverviewSea"
*r end
```

2.9 Configuration profiles

2.9.1 Get the current configuration profile name

Returns the name of the current controller configuration profile.

Syntax

```
Room profile get
```

Response

```
Current profile is [Default]
```

2.9.2 Change the configuration profile

Change configuration profile by specifying the name of a configuration Profile. Password is optional. Changing the configuration profile will result in a Controller soft restart.

Syntax

```
room profile set Name:"<Profile name>" Password:"<Optional password>"
```

Response

```
*r start - room profile set
*r room profile set success
*r end
```

or

```
*r start - room profile set
*r room profile set error (No such profile, valid profiles are [Default, [Suspended mode]])
*r end
```

Example

```
room profile set Name:"Combined room" Password:"CyvizPW"
```

Response

```
*r start - room profile set
*r room profile set success
*r end
```

2.10 Alarms

2.10.1 Get list of alarms

Will return a list over Alarms defined on the Controller.

Syntax

```
getalarmlist
```

Response

```
*r start – getalarmlist
*r Name: "Fire alarm" State: "On"
*r Name: "Man with gun alarm" State: "Off"
*r end
```