Remote Control of VIOSO Software

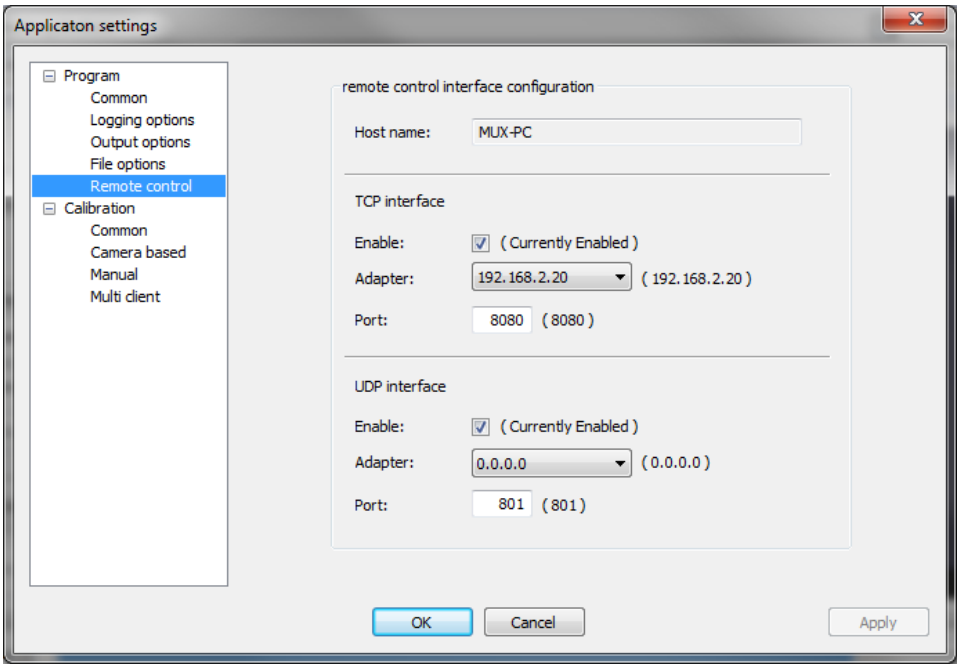
For: VIOSO Calibrator (part of VIOSO Anyblend, Anyblend VR&SIM) and for VIOSO Player

Last edited: 2022-10-06

Table of Contents

1 Preparation

The network control interface must be enabled.
You can find the network related options in the application menu:
“Options → Settings → Remote Control”.



You must enable the desired interface type first. Then you should configure the network adapter and port to use. The special adapter ip 0.0.0.0 represents all available network adapter of used pc. After an restart of the application, the configured interface is available.

2 Usage

You can send commands in two ways.

1. You can use each browser application to send a command request.
   Please type the command string with leading ip and port address in the address bar and press the enter button.
   e.g. "http://192.168.2.20:8080/player.htm?state=play"
   The server will answer with a HTTP response header and attached HTTP side or attached response string.

2. You can use your own network tool, media control solution or with the installation distributed test tools, to send command to the server.
   There a two ways to send the command string:

   1. You can send a simple command string, terminated with "\r\n" or "\n" character(s)
      (0xD 0xA hexadecimal, <CR><LF> or 0xA hexadecimal, <LF>).
      e.g. "player.htm?state=play\r\n"
      If you use the TCP interface, the server will answer with a simple response string.
      If you use the UDP interface, the server will send no answer (fire and forget).

   2. If you use the TCP interface, you can send a command request encapsulated in a minimum HTTP header, using the GET method.
      e.g. "GET /player.htm?state=play HTTP/1.1\r\n\r\n"
      The server will answer with a HTTP response header and attached HTTP side or attached response string.

3 Test Tools

If you have chosen install „Shared Components“ during Installation, there are several tools to equip you to test:

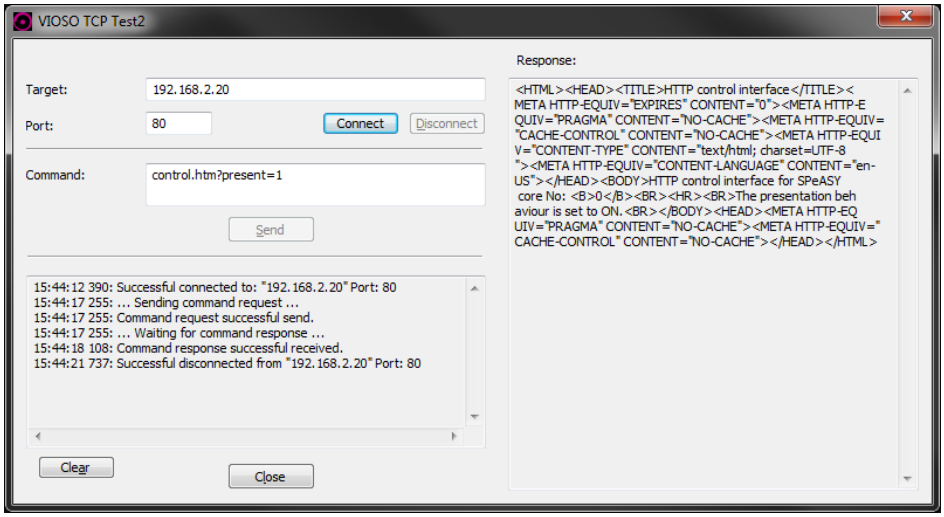
- TCP-/ UDP-interface
- command behavior
- spy/ record network communication with third party software

You can execute/ find the test tools over the application startmenu entry; sub-entry “Tools”.

Except TCP Test Tool 2, each of the described test tools below, have a command line interface. With the command line interface you are able to use the test tools in scripts, links and so on. You can find a documentation of available command line options inside sub-entry “Manuals”.

3.1 TCP Test Tool 2

With the **TCP Test Tool 2** you can test the TCP interface using a permanent (long live) connection to **one** target. This tool also display a log of all network operations.



You can specify one target by adding the IP of the remote pc to the target list.
By pressing the “Send” button, the tool:

- creates a connection to the current treated target
- sends the command string as simple string (see above)
- waits for response
- depicts the response string inside the response area on right side of the application
- and closes the connection to the current treated target.

3.1.1 Where to find it

Software location after installation of VIOSO Anyblend, Anyblend VR&SIM or VIOSO Player:

- Via Windows Start Menu:All Programs / VIOSO [Program Name] / Tools / TCP Test Tool
- Exe file location: C:\[Path of installation]\Shared Tools\ VIOSO_TCP_Test.exe

3.1.2 Command line usage

Available parameters:

/D:"ip"	ip of the destination if **/D** is not specified, the app will be configured for broadcasting
/P:#	# => port number
/C:"..."	"..." => command string to send (plain)
/A	if set, the application tries to send the command and closes app automatically

Example:

```
VIOSO_TCP_Test2.exe/D:"192.168.2.2" /P:801 /C:"control.htm?present=1" /A
```

Sends the command "**control.htm?present=1**" to **192.168.2.2,** using port **801** automatically.

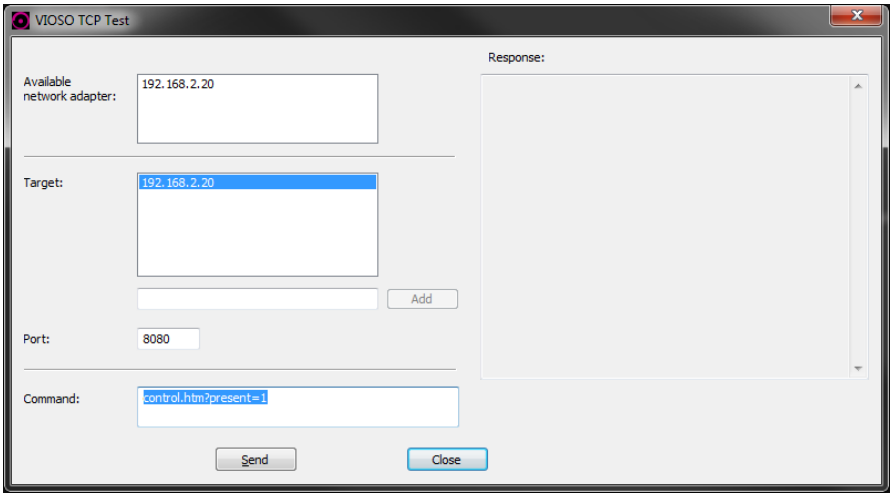
Exit code:

VIOSO_UDP_Test.exe returns an exit code on exit:

– 0 : successful
– 1 : an error occurred (Command could not be sent to all destinations.)

3.2 TCP Test Tool

The **TCP Test Tool** is the predecessor to **TCP Test Tool 2**.



This tool does the same actions as it successor TCP Test Tool 2, but does not use a permanent connection and does not monitor all network traffic. It just outputs the response of the target.

We recommend to use TCP Test Tool 2 whenever possible.

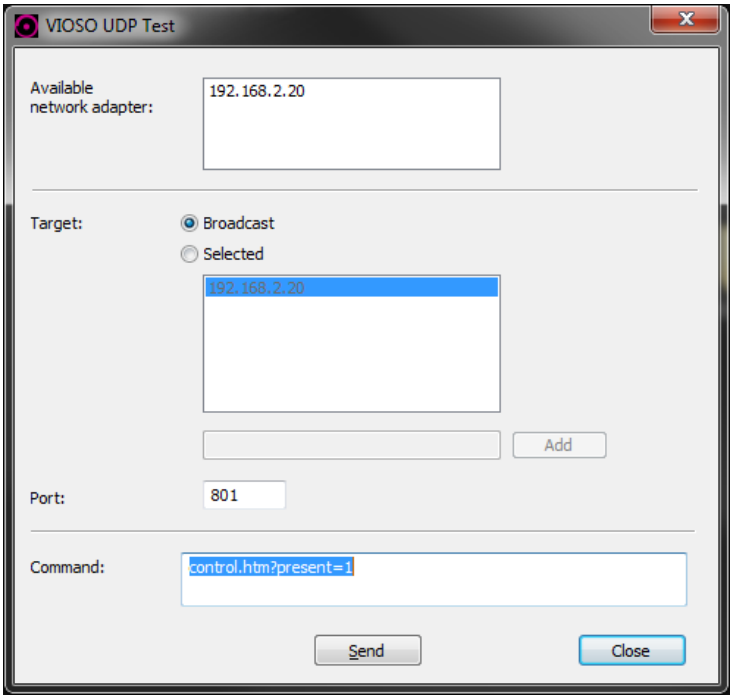
Software location after installation of VIOSO Anyblend, Anyblend VR&SIM or VIOSO Player:

- Via Windows Start Menu:All Programs / VIOSO [Program Name] / Tools / TCP Test Tool 2
- Exe file location: C:\[Path of installation]\Shared Tools\ VIOSO_TCP_Test2.exe

See previous chapter for command line parameters.

3.3 UDP Test Tool

With the **UDP Test Tool** you are able to test the UDP interface.



You can specify one ore more targets by adding the IP of a desired PC to the target list. Beside this option, the tool is able to send the command string as broadcast to all computers in network.

By pressing the “Send” button, the tool sends a datagram, containing the command string, to all selected targets successively. If you use the broadcast option, the tool sends a broadcast datagram.

3.3.1 Where to find it

Software location after installation of VIOSO Anyblend, Anyblend VR&SIM or VIOSO Player:

- Via Windows Start Menu:All Programs / VIOSO [Program Name] / Tools / UDP Test Tool
- Exe file location: C:\[Path of installation]\Shared Tools\ VIOSO_UDP_Test.exe

3.3.2 Command line usage

Available parameters:

`/D:"ip"`	ip of the destination if **/D** is not specified, the app will be configured for broadcasting
`/P:#`	# => port number
`/C:"..."`	"..." => command string to send (plain)
`/A`	if set, the application tries to send the command and closes app automatically

Example:

```
VIOSO_UDP_Test.exe /D:"192.168.2.2" /P:801 /C:"control.htm?present=1" /A
```

Sends the command "**control.htm?present=1**" to **192.168.2.2,** using port **801**, automatically.

Exit code:

VIOSO_UDP_Test.exe returns an exit code on exit:

- 0 : successful
- 1 : an error occurred (Command could not be sent to all destinations.)

4 Command Reference

4.1 Media Playback related commands

TCP/IP and UDP	description	from
player.htm/ playcontrol.htm	**Player related actions (playcontrol.htm with human readable response)**	1.0
player.htm?state=play	Sets the playback state, which starts the playback at the currently selected playlist item	1.0
player.htm?state=pause	Pauses the current playback	1.0
player.htm?state=stop	Stops the current playback, the seeker is set to the beginning of the currently selected playlist item	1.0
player.htm?select=next	Selects the next item of the playlist. If playback state is set, the playback of the selected item starts	1.0
player.htm?select=prev	Selects the previous item of the playlist. If playback state is set, the playback of the selected item starts	1.0
player.htm?select=1	Selects the first item of the playlist. If playback state is set, the playback of the selected item starts	1.0
player.htm?select=99999	Selects the last item of the playlist. If playback state is set, the playback of the selected item starts	1.0
player.htm?mute=1	Mutes the player audio output temporary	1.2
player.htm?mute=0	Re-Enables the player's audio output	1.2
player.htm?audio=0..100	Sets the global audio level to [0..100]%	1.2
player.htm?autoplay=1	Sets the autoplay state active	1.2
player.htm?autoplay=0	Sets the autoplay state inactive	1.2
player.htm?load=[filename]	Loads a playlist XML file from disk, specified by [filename] E.g.: player.htm?load=D:\My_Playlists\list1.xml !! not all characters are allowed, e.g. #, & !!	1.2
player.htm?save=[filename]	Saves the current playlist to a file specified by [filename] E.g.: player.htm?save=D:\My_Playlists\list1_backup.xml !! not all characters are allowed, e.g. #, & !!	1.2
player.htm?add=[filename, pause, stop, jump]&pos=[p]	Adds a new media or control item to the playlist. If the additional parameter pos=[p] will be used, the new item will be inserted at list position [p]. e.g.: player.htm?add=D:\My_Content\Image.jpg adds the image Image.jpg to the playlist player.htm?add=Image.jpg adds the image Image.jpg to to playlist. The file must located inside the default content directory. player.htm?add=pause adds a pause control item to the playlist !! not all characters are allowed, e.g. #, & !!	1.2
player.htm?move=[up, down, top, end, [t]]&select=[p]	Moves the playlist entry [p] to the specified position. [t] can be used to define an desired absolute position. e.g.: player.htm?move=up&select=5 moves playlist item no. 5 one position up player.htm?move=2&select=7 moves playlist item no 7 to the position 2.	1.2
player.htm?activate=[p]	Activates the playlist item [p]	1.2

TCP/IP and UDP	description	from
`player.htm?deactivate=[p]`	Deactivates the playlist item [p]	1.2
`player.htm?loop=[p]`	Activates the loop behaviour of item [p], if available	1.7
`player.htm?noloop=[p]`	Deactivates the loop behaviour of item [p], if available	1.7
`player.htm?delete=[[p], all]`	Deletes entry [p] from the playlist, [all] to clear the whole playlist	1.2
`player.htm?refresh=[[p], all]`	Refreshes the meta-data (size, date, etc.) of the item [p] in the playlist e.g. player.htm?refresh=2 refreshs the meta data of item 2 in the playlist	1.2
`Player.htm?tmPerImg=[s]&select=[p]`	In case of an image sequence item this command changes the show time of each image; [s] seconds as float value, [p] position of the item to change e.g. player.htm?tmPerImg=4.5&select=3 sets the time per image parameter of the image sequence at position 3 to 4.5 seconds	1.2
`player.htm?status`	Retrieves a comma separated list with status informations – current index of selected/ played item in playlist – -1 => empty playlist – n => 1 based index of the current selected/ played item – quantum of items in playlist – state of selected/played item – -1 => undefined state – 0 => item is in stop state – 1 => item is in pause state – 2 => item is in play state	1.7
`player.htm?status=index`	Retrieves the index of currently selected/ played item in playlist – [-1] => empty playlist – [n] => 1 based index of the current selected/ played item	1.7
`player.htm?status=count`	Retrieves the quantum of items in playlist	1.7
`player.htm?status=state`	Retrieves the state of the current selected/ played item – [-1] => undefined state – [0] => item is in stop state – [1] => item is in pause state – [2] => item is in play state	1.7
`player.htm?list=state`	Retrieves a survey of useful information as comma separated list. Format: a,b,c,d,e,f,g, a => showmode • [1]: on • [0]: off b => selected playlist item position c => player state • [-2]: unknown • [-1]: off • [0]: stop • [1]: pause • [2]: play d => stream position in percent, integer [0 .. 100] e => audio balance in percent, integer [-100 .. 100] f => audio volume in percent, integer [0 .. 100] g => duration in milliseconds [0 .. n]	1.7
`player.htm?list=volume`	Retrieves the current audio volume pitch setting. – [-2]: unknown – [-1]: offline – [0 .. 100] pitch in percent (integer)	1.7
`player.htm?list=balance`	Retrieves the current audio balance setting. – [-2]: unknown – [-1]: offline – [-100 .. 100] setting in percent (integer)	1.7

TCP/IP and UDP	description	from
`player.htm?list=playing`	Retrieves informations about the play state of the playlist. – [-2]: unknown – [-1]: off – [0]: stop – [1]: pause – [2]: play	1.7
`player.htm?list=showmode`	Retrieves the current show mode state. – [1]: on – [0]: off	1.7
`player.htm?list=playItem`	Retrieves the current selected playlist item position.	1.7
`player.htm?list=position`	Retrieves the current stream position in percent. – [-2]: unknown – [-1]: offline – [0 .. 100]: integer position in percent	1.7
`player.htm?list=seeker`	Retrieves the current stream position in more detailed mode. Format: a,b, a => seeker position • [-2]: unknown • [-1]: offline • [0.0 .. 1.0]: relative position inside the stream as double value b => duration of the stream in milliseconds • [0 .. n]: duration in milliseconds, 0 for infinite items, images and so on	1.7
`player.htm?list`	Retrieves a list of all items in playlist. One item per line. Per line contained information: a,b,c,d,e,f, a => item position in playlist b => type of the item • [ctrl]: control item • [unknown]: unknown item • [file]: media file • [seq]: image sequence • [mrd]: model rendering definition • [error]: error :) c => subtype of the item • [image]: image file • [video]: video file • [unknown]: unknown • [goto]: jump to control item • [pause]: pause control item • [stop]: stop control item d => description • file name in case of media file or [-unknown-] • empty for pause, stop control item • jump to item position for jump control item • path and search mask for image sequence or [-unknown-] e => duration • duration of item in milliseconds • [infinite] or empty otherwise f => additional description flag, one or more of follow bit flags • 0x1: current selected item • 0x2: item is disabled • 0x4: infinite duration • 0x8: loop ability is set	1.7
`player.htm?item=mediainfo&select=[p]` `player.htm?item=mediainfo`	Retrieves media informations about the current selected playlist item if no select parameter is specified, or about playlist item [p]. The media information comes as CSV: a,b,c,d,e,f,g,h,i,j, a => item type • [ctrl]: control item	1.8

TCP/IP and UDP	description	from
	• [file]: media file • [seq]: image sequence • [mrd]: model rendering definition • [-]: not specified b => subtype of the item • [image]: image file • [video]: video file • [goto]: jump to control item • [pause]: pause control item • [stop]: stop control item • [-]: not specified c => file name or [-] if not specified d => file size or [-] if not specified e => duration in milliseconds or [-] if not specified f => image size [width x height] or [-] if not specified g => bitrate or [-] if not specified h => video format list or [-] if not specified i => audio format list or [-] if not specified j => image format list or [-] if not specified	
`player.htm?seekerto=[f]`	Sets the stream position of an active playing stream based playlist item to a specific position. The presenter pipeline must be in showmode on state and the current played playlist item must be in play or pause state. [f] can be an integer [0 .. 100] defining the relative stream position in percent, or a double [0.0 .. 1.0] defining the relative stream position more precisely	1.7
`player.htm?seekerby=[f]`	Used to change the stream position of an active playing stream based playlist item by a specified portion. The presenter pipeline must be in showmode on state and the current played playlist item must be in play or pause state. [f] can be an positive or negativ integer, defining the portion in milliseconds the stream position should be changed, or a double [-1.0 .. 1.0] defining the portion in relative way Example: player.htm?seekerby=-10000 => goes 10 seconds backward from current stream position player.htm?seekerby=0.1 => goes 10% forward from current stream position	1.7
`player.htm?smoothborder=[1,0]`	Enables, disables the smooth border presenter pipeline feature.	1.7
`player.htm?fullscreen=[1,0]`	Enables, disables the fullscreen presenter pipeline feature.	1.7
`player.htm?blank=[1,0]`	Enables, disables the blank screen(s) presenter pipeline feature.	1.8

4.2 Receiving meta information about files

TCP/IP and UDP	description	from
mediafile.htm	**Interface to receive informations about available media files**	1.7
mediafile.htm?list mediafile.htm?list&fmt	Retrieves a list of all available media files onto the server (comma separated list). fmt: The response is given as HTTP side (one file per line).	1.7
playlistfile.htm?list playlistfile.htm?list&fmt	Retrieves a list of all available playlist files onto the server (comma separated list). fmt: The response is given as HTTP side (one file per line).	1.8
calibfile.htm?list calibfile.htm?list&fmt	Retrieves a list of all available calibration files onto the server (comma separated list). fmt: The response is given as HTTP side (one file per line).	1.8

4.3 Receiving preview images

TCP/IP and UDP	description	from
preview.htm	**Interface to receive preview images of playlist items**	1.7
preview.htm?select=[p]	Retrieves the preview image of playlist entry [p] . e.g.: preview.htm?select=5 retrieves the preview image of playlist entry 5.	1.7

4.4 Controlling render output behaviour

TCP/IP and UDP	description	from
presenter.htm	**Get/set presenter pipeline specific parameters**	1.7
presenter.htm?get=globalcolor presenter.htm?get=globalcolor&fmt	Retrieves the current global color settings (comma separated list). fmt: The response is given as HTTP side. Format: a,b,c,d, a => showmode state • [0]: off • [1]: on b => red color channel value, [0 .. 1000], empty if showmode off c => green color channel value, [0 .. 1000], empty if showmode off d => blue color channel value, [0 .. 1000], empty if showmode off	1.7
presenter.htm?get=fullscreen presenter.htm?get=fullscreen&fmt	Retrieves the current fullscreen presenter feature setting state. fmt: The response is given as HTTP side. – [0]: not set currently – [1]: current set – [-1]: current not available	1.7
presenter.htm?get=smoothborder presenter.htm?get=smoothborder&fmt	Retrieves the current smooth border presenter feature setting state. fmt: The response is given as HTTP side. – [0]: not set currently – [1]: current set – [-1]: current not available	1.7
presenter.htm?get=blank presenter.htm?get=blank&fmt	Retrieves the current blank screen(s) presenter feature setting state. fmt: The response is given as HTTP side. – [0]: not set currently – [1]: current set [-1]: current not available	1.8
presenter.htm?globalcolor=[r,g,b] presenter.htm?globalcolor=[r,g,b]&fmt	Sets new global color values. The presenter pipeline has to be in showmode on state. fmt: The response is given as HTTP side. r,g,b in the range of [0 .. 1000], empty color place for unchanged	1.7
presenter.htm?smoothborder=[1,0] presenter.htm?smoothborder=[1,0]&fmt	Enables, disables the smooth border presenter pipeline feature. fmt: The response is given as HTTP side.	1.7
presenter.htm?fullscreen=[1,0] presenter.htm?fullscreen=[1,0]&fmt	Enables, disables the fullscreen presenter pipeline feature. fmt: The response is given as HTTP side.	1.7
presenter.htm?blank=[1,0] presenter.htm?blank=[1,0]&fmt	Enables, disables the blank screen(s) presenter pipeline feature. fmt: The response is given as HTTP side.	1.8

4.5 General program control

TCP/IP and UDP	description	from
control.htm	**General program control**	1.0
control.htm?present=1	Enables show mode while Player is running	1.0
control.htm?present=0	Disable show mode while Player is running	1.0
control.htm?shutdown=1	Shuts down the operating system	1.0
control.htm?shutdown=2	Shuts down the operating system and reboots	1.0
control.htm?exit	Close the currently addressed running instance of the player	1.0
control.htm?calibrate=M0	Starts a camera based recalibration without any user interaction (1-click recalibration)	1.0
control.htm?calibrate=MC	Stops a running calibration without changing the currently used calibration.	1.0
control.htm?calibrate=N0	Starts a camera based network recalibration without any user interaction (1-click recalibration)	1.0
control.htm?calibrate=NC	Stops a running network calibration without changing the currently used calibration.	1.0
control.htm?calibrate=status	Returns 1, in case a calibration is running, 0 otherwise	1.8
control.htm?execute=[filename] [Param] control.htm?execute=[”filename”] [Param]	Executes the specified file. The executable has to be located inside the subdirectory “Execute” inside the data exchange directory of the server. [Param]: optional command line parameter !! not all characters are allowed, e.g. #, & !!	1.0
control.htm?preset=[x]	Loads the calibration file “Preset[x].sps” that must located inside the default calibration folder. [x] defines an integer. example: control.htm?preset=12 loads the file “Preset12.sps”	1.0
control.htm?settings=[x]	Loads the calibration file “Settings[x].sps” that must located inside the default calibration folder. [x] defines an integer. The difference towards “preset” is, that the calibration file will be loaded by the GUI and not intern. example: control.htm?settings=1 loads the file “Settings1.sps”	1.0
control.htm?clear=[xxx]	Clears configurations and data. [xxx] defines which data should be cleared, each letter defines a special kind of data: - c => calibration data - r => renderpipe configuration - p => custom resource parameter (e.g. custom display names) - l => file lists - s => display split informations example: control.htm?clear=cp clears all calibration data and the custom resource parameter	1.7
control.htm?save=[filename] control.htm?save=[filename]&wait=[x]	Saves the current calibration state to specified “filename” wait: optional, 1: wait until core has been saved | 0: set request only !! not all characters are allowed, e.g. #, & !!	2.0

TCP/IP and UDP	description	from
`control.htm?ShowOutput=0`	Makes all current output windows invisible (on master)	2.3
`control.htm?ShowOutput=1`	Makes all current invisible output windows visible again (on master)	2.3

4.6 3D model specific commands

TCP/IP and UDP	description	from
model3d.htm	**3D-model depended computation control**	1.6
model3d.htm?status model3d.htm?status&fmt	Retrieves the current status of the 3D-model engine, as text string. fmt: The response is given as HTTP side. Available response strings (defined in c conform string format definition): - "Status: \"Unitialized\"\r\n" - "Status: \"Waiting for base method settings.\"\r\n" - "Status: \"Preparing base method settings.\"\r\n" "Progress: \"%1.0f %%\"\r\n" (percental progress) - "Status: \"Ready for static model computation.\"\r\n" - "Status: \"Performing static model computation.\"\r\n" "Progress: \"%1.0f %%\"\r\n" (percental progress) - "Status: \"Undefined entity state.\"\r\n" - "Could not create 3D model treatment status report.\r\n"	1.6
model3d.htm?process	Starts the 3D-model computation, based on current settings. Available response strings: - "Computation request was set successful." - "Could not send process signal." - "Could not configure 3D model treatment behaviour." - "3D model treatment behaviour is not in right state."	1.6

4.7 Script Engine commands

TCP/IP and UDP	description	from
script.htm	**Interface to control the script engine**	1.6
script.htm?status script.htm?status&fmt	Retrieves the current status of the script engine, as text string. fmt: The response is given as HTTP side. Available response strings (defined in c conform string format definition): - "Script Host ready for use." - "Script: \"%s\"\r\n" (name of the current treated script) "Result: \"loaded\" (%s)\r\n" (error string) - "Script: \"%s\"\r\n" (name of the current treated script) "Result: \"could not load\" (%s)\r\n" (error string) - "Script: \"%s\"\r\n" (name of the current treated script) "Step: \"%u/%u\" (%s)\r\n" (example: Step: "2/21" (in progress)) "Result: \"%s\"\r\n" (error string) "Desc: \"%s\"\r\n" (short summary of the current treated script line) - "Script: \"%s\"\r\n" (name of the current treated script) "Result: \"successful processed\"\r\n" - "Could not create script host report.\r\n"	1.6
script.htm?execute=filename script.htm?execute=filename&param=[...]	Executes the specified script file. Available response strings (defined in c conform string format definition): - "Script host was initialized successful." - "Could not send activation signal." - "Script host executes another script currently or is in wrong entity state." - "Script file not found or an unknown error occurs." - "Could not convert file name." – "Could not create intern synchronize interface." In additional to the script file name to execute, a parameter string can be defined to define variables. A variable can defined by: [name]=[value] or [name]=[value],[format] where: – [name] => name of the variable – [value] => value of the variable – [format] => optional format specification • [f]: [value] is a float value • [d]: [value] is a double value • [i]: [value] is an integer value No format specificaton implies a string value. More than one variable can defined by a semicolon separated list. Example: script.htm?execute=start.ini&param=val1=front;val2=2.5,f The script “start.ini” should be processed and two variables should be defined: “val1” as string variable with value “front” and “val2” as float variable with value “2.5”. This variables can be used by “start.ini” script.	1.6
script.htm?abort	Aborts the execution of a currently processed script.	

4.8 Common Information commands

TCP/IP and UDP	description	from
info.htm	**Interface to retrieve common information**	1.7
info.htm?displays info.htm?displays&fmt	Retrieves a list of all available displays fmt: The response is given as HTTP page. Text based version defines one display per line as comma separated parameter list: adapter id, name, type of display, width, height, pos-X, pos-Y, id-string, Handle	1.7
info.htm?desktopdisplays info.htm?desktopdisplays&fmt	Retrieves a list of all available desktop displays fmt: The response is given as HTTP page. Text based version defines one display per line as comma separated parameter list: adapter id, name, type of display, width, height, pos-X, pos-Y, split-Rows, split-Cols, id-string, Handle	2.5
info.htm?deskpartident&possz=[p]&id=[i]&name=[n]&fmt	Shows an identification window on the specified place fmt: optional, the response is given as HTTP page. [p] : specifies the position and size of the identification window format: x,y,w,h (e.g. 0,0,640,480 defines a window at position (0,0) with size (640x480)) [i] : optional, can be used to define an identification string [n] : optional, can be used to define an additional name	2.4

4.9 Project Manager commands

TCP/IP and UDP	description	from
pm.htm	**Interface to control the project manager**	6.0.2
pm.htm?status[&fmt]	Retrieves the current status of the project manager. fmt: The response is given as HTTP. Layout for plain text response. Line separated CSV. first Line: a,"b",c, - a => [0,1] specifies if a new project can be loaded or a specific project step can be started currently (0: no, 1: yes) - b => path string of the current project - c => count of project steps of current project Other lines: a,"b","c" This lines contain informations about available project steps. - a => index of the project step - b => project step description string - c => project step state string	6.0.2
pm.htm?load=project[&fmt]	Tries to load the specified project. A [project] can be defines by it name (located in default project folder), or as complete path. fmt: The response is given as HTTP. Layout for plain text response. Line separated CSV: a,"b" - a => [0] : an error occurs, [1] : success - b => error string	6.0.2
pm.htm?start=step[&queue][&fmt]	Tries to start the specified project step [step]. Currently implemented are: "Design", "Calibrate", "Edit", "Edit MRD", "Export", "Recalibrate", "Activate", "Deactivate Not all steps are available in in all project types. Possible steps can be retrieved by pm.htm?status (project step description). fmt: The response is given as HTTP. queue: If project step can't be started immediately, the project step is queued to start after previous steps Layout for plain text response. Line separated CSV: a,"b" - a => [0] : an error occurs, [1] : success - b => error string	6.0.2