ELIUM Device Application Notes
RS232/Network Remote Control Description

1. General

Date: **15.01.2024**
Revision: **0.14**
Scope: The goal of this document is to describe how ELIUM Device can be controlled through RS232 connector (RS232-RC mode) or via network TCP connection (NET-RC mode).

2. The RS232/Network attachment

One of the many features implemented in ELIUM Device application is the possibility of bidirectional controlling the device through RS232 connection or via network TCP connection according to this Remote Control description.

3. Example application

ELIUM Device can be controlled from your PC. Be aware that only two wires of nine are used (RX and TX) in the case of RS232-RC mode. The TCP port 26 is the default communication port in the case of NET-RC mode.

4. Working conditions

The communication can work correctly only if the following conditions are fulfilled.

For the RS232-RC mode (via RS232 connection):

- Baud: 115.200 (default)
- Parity: none
- Data Bits: 8
- Stop Bits: 1
- Flow Control: none

For the NET-RC mode (via network TCP connection):

The client (PC) connects to the server (ELIUM Device) with its network address and port via TCP and fulfils the commands described below to control the device.

5. Attention:

Please mention that after switching on the unit by pushing the Power Switch, the unit is starting and during this procedure should not be disturbed. If you send anything during the starting procedure, the unit can go to Firmware update procedure. So it is recommended waiting until receive text information from application part - "#READY".

6. Note:

In certain moments ELIUM Device sent other "#" lines too.
The syntax is: #?/text/?#

They give information about: Boot, Application Version etc. These lines should not be taken into account.

7. Commands without additional return value

Each command starts with "<" char and ends with ">". Immediately after ">" sign is received, command will be performed.

If command is not recognized (for example, if <ABC> command is sent), the following text should appear on your terminal window:

```
#COMMAND: <ABC>
#ERROR: Command not supported
```

If command is supported and was received correctly you should get something like:

```
#COMMAND : <ON>
#OK
```

The line "#COMMAND:" is sent before command is performed. It only indicates that certain string of chars was received by the device. After that, command is performed and, if this action is finished, the line "#OK" should be sent.

In order to simplify (from programmer point of view) the reception of responses (so called confirmations) the first sign sent from the device is always "#". So, host should wait for "#", the next letter should indicate whether everything was all right or not (#C, #E or #C, #O).

Command	**Description**
<REB>	Reboot device
<SSS n>	Set RS232 Baud Rate n = RS232 baud rate value Supported values are: 9600, 19200, 38400, 115200. Baud Rate will be changed immediately. After that, application will always start with the new Baud Rate.
<IPC DHCP ON> <ETH1 IPC DHCP ON>	Enable DHCP for a network configuration for ETH / ETH-1 interface The previously set static network configuration parameters are saved and can be still used after disabling DHCP.
<IPC DHCP OFF> <ETH1 IPC DHCP OFF>	Disable DHCP for a network configuration for ETH / ETH-1 interface The previously set static network configuration parameters are used for networking. Please refer <SIP STATIC s>, <SIP s> commands description.
<SIP STATIC s> <ETH1 SIP STATIC s>	Change a static network configuration for ETH / ETH-1 interface s = list of string attributes separated with “;” delimiter in format: *ipaddr;netmask;gateway;dns1;dns2*

	where *ipaddr* = device IP-address or "-" if no change; *netmask* = device network mask or "-" if no change; *gateway* = network gateway IP-address or "-" if no change; *dns1* = Primary DNS server IP-address or "-" if no change; *dns2* = Secondary DNS server IP-address or "-" if no change; When DHCP is enabled the static network configuration parameters are only saved without any impact to the actual networking configuration. In the case when DHCP is disabled the command acts the same as the <SIP s> command. Example: #COMMAND: <SIP STATIC 10.1.1.55;255.255.255.0;10.1.1.1;10.1.1.1;8.8.8.8> #OK
<SIP s> <ETH1 SIP s>	Change a static network configuration for ETH / ETH-1 interface s = list of string attributes separated with ";" delimiter in format: *ipaddr;netmask;gateway;dns1;dns2* where *ipaddr* = device IP-address or "-" if no change; *netmask* = device network mask or "-" if no change; *gateway* = network gateway IP-address or "-" if no change; *dns1* = Primary DNS server IP-address or "-" if no change; *dns2* = Secondary DNS server IP-address or "-" if no change; Command should be used only in the case when DHCP is disabled. Otherwise the respective error message is sent. Example: #COMMAND: <SIP 10.1.1.54;-;10.1.1.1;10.1.1.1;-> #OK
<ETH2 SIP s>	Change a static network configuration for ETH-2 (Control) interface s = list of string attributes separated with ";" delimiter in format: *ipaddr;netmask* where *ipaddr* = device ETH-2 (Control) interface IP-address or "-" if no change; *netmask* = device ETH-2 (Control) interface network mask or

	“-” if no change; ETH-2 (Control) interface IP-address cannot be the same as ETH-1 interface IP-address for the both DHCP (obtained IP) and the static network configuration cases. Otherwise the respective error message is sent. Please refer <SIP STATIC s>, <SIP s> and <IPC> commands description. Note: Available only for devices equipped with dual ethernet interfaces (ETH-1 / ETH-2). Otherwise the respective error message is sent. Example: #COMMAND: <ETH2 SIP 192.168.198.210;255.255.255.0> #OK
<SDM n>	Set video display mode n = decimal code for video display mode Possible decimal codes for video display mode are: 0 PAL (analog video output via CVBS) 1 NTSC (analog video output via CVBS) 2 480i 60Hz 3 576i 50Hz 4 480p 60Hz 5 576p 50Hz 6 720p 50Hz 7 720p 60Hz 8 1080i 50Hz 9 1080i 60Hz 10 1080p 24Hz 11 1080p 50Hz 12 1080p 60Hz 13 2160p 50Hz 420 (UHD 4:2:0 - additional license required) 14 2160p 60Hz 420 (UHD 4:2:0 - additional license required) 15 2160p 24Hz 422 (UHD 4:2:2 - additional license required) 16 2160p 25Hz 422 (UHD 4:2:2 - additional license required) 17 2160p 30Hz 422 (UHD 4:2:2 - additional license required) Example: #COMMAND: <SDM 12> #OK
<SARM n>	Set video aspect ratio adjustment mode n = decimal code for video aspect ratio adjustment mode Possible decimal codes for video aspect ratio adjustment mode are: 0 Auto 1 Full Stretch 2 4:3 3 16:9 4 Non-Linear 5 Original Size 6 4:3 Ignore 7 4:3 Letterbox 8 4:3 Pan&Scan 9 4:3 Combined

	10 16:9 Ignore 11 16:9 Letterbox 12 16:9 Pan&Scan 13 16:9 Combined Example: #COMMAND: <SARM 11> #OK
<ON>	Turn on device Note: Command should be used only in Standby mode. Otherwise the following error message is sent: #ERROR: Not in standby
<OFF>	Turn off device Note: Command should not be used in Standby mode. Otherwise the following error message is sent: #ERROR: Already in standby
<RMCC n>	Simulates an input via remote control n = remote control key code for the given key Possible key codes (for device remote control) are: 22 Key ON/OFF 2 Key '1' 5 Key '2' 6 Key '3' 9 Key '4' 10 Key '5' 13 Key '6' 14 Key '7' 17 Key '8' 18 Key '9' 21 Key '0' 34 Key MODE 37 Key RADIO/TV 38 Key MUTE 41 Key LAST 25 Key UP 26 Key DOWN 29 Key LEFT 33 Key RIGHT 30 Key OK 42 Key MENU 45 Key EXIT 58 Key Red (PVR / DVR) 61 Key Green (MOVIE / DVD) 62 Key Yellow (MUSIC / MP3/JPEG) 65 Key Blue (MEDIA / GAME) 46 Key << (rew. back) 49 Key PLAY/PAUSE 50 Key >> (rew. forward 53 Key |<< (go prev.) 54 Key REC/STOP

	57 Key >>| (go next) 66 Key INFO 69 Key EPG 70 Key TIMER 73 Key TXT 74 Key PIP 77 Key SEARCH / FREEZE 78 Key TECH.INFO / ZOOM 81 Key AUDIO VIDEO
<RMC c>	Simulates an input via remote control Possible 'c' chars are: 0 Key '0' 1 Key '1' 2 Key '2' 3 Key '3' 4 Key '4' 5 Key '5' 6 Key '6' 7 Key '7' 8 Key '8' 9 Key '9' i Key OK m Key EXIT b Key MENU u Key UP j Key DOWN k Key RIGHT h Key LEFT e Key TXT d Key EPG f Key REC/STOP g Key INFO x Key Yellow (MUSIC / MP3/JPEG) c Key Red (PVR / DVR) v Key RADIO/TV n Key LAST t Key ON/OFF a Key MUTE J Key PIP K Key SEARCH / FREEZE L Key TECH.INFO / ZOOM M Key AUDIO VIDEO A Key MODE B Key Green (MOVIE / DVD) C Key Blue (MEDIA / GAME) D Key << (rew. back) E Key PLAY/PAUSE F Key >> (rew. forward) G Key |<< (go prev.) H Key >>| (go next) I Key TIMER
<DIG n>	Simulates a digit input from the remote control (n = 0..9)

<MNU>	GUI Menu request
<EXT>	GUI Exit and Leave Menu request
<CNF>	GUI Confirmation and Selection request
<NAV U>	GUI Move Up request Note: The main purpose of the <NAV U> command is the GUI navigation. Command is not assumed to be used for channels switch. <PRT U> and <PRR U> commands should be used instead.
<NAV D>	GUI Move Down request Note: The main purpose of the <NAV D> command is the GUI navigation. Command is not assumed to be used for channels switch. <PRT D> and <PRR D> commands should be used instead.
<NAV L>	GUI Move Left request
<NAV R>	GUI Move Right request
<INF>	GUI Informations menu request
<TVL>	GUI Channellist request (activates the list of available programs)
<AVM 1> <AVM ON>	GUI Audio and Multifeed menu On request
<AVM 0> <AVM OFF>	GUI Audio and Multifeed menu Off request
<AVM U>	GUI Audio and Multifeed menu Move Up request
<AVM D>	GUI Audio and Multifeed menu Move Down request
<AVM L>	GUI Audio and Multifeed menu Move Left request
<AVM R>	GUI Audio and Multifeed menu Move Right request
<EPG 1> <EPG ON>	GUI EPG Electronic Program Guide menu On request
<EPG 0> <EPG OFF>	GUI EPG Electronic Program Guide menu Off request

<EPG U>	GUI EPG Electronic Program Guide menu Move Up request
<EPG D>	GUI EPG Electronic Program Guide menu Move Down request
<EPG L>	GUI EPG Electronic Program Guide menu Move Left request
<EPG R>	GUI EPG Electronic Program Guide menu Move Right request
<RCL>	Return to the last channel request
<SAC n>	Set audio stream for the current channel n = audio stream ID (received with <GAC> command) Available audio streams can retrieved by <GAC> command. Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) or in Timeshifting mode (when the current mode is Time Shift). Otherwise the respective error message is sent. In the case of Timeshifting mode the command should be used only to change audio stream within the current recording and will not change the current playback. Example: #COMMAND: <SAC 5406> #OK
<REC 1> <REC ON>	Start recording of the current program Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). The recording storage should be also available (please refer <HDD INFO>, <NAS GET CONFIG n> commands description). Otherwise the respective error message is sent.
<REC 0> <REC OFF>	Stop recording of the current program The recording should be previously started (please refer <REC 1> command description). Otherwise the respective error message is sent. In the case of Timeshifting mode the command will also stop timeshifting.
<REC D>	Stop recording of the current program and remove recording files The recording should be previously started (please refer <REC 1> command description). Otherwise the respective error message is sent. In the case of Timeshifting mode the command will also stop timeshifting. Command is assumed to be used for Timeshifting mode for the case when the recordings are no more needed after timeshifting.
<TSHFT 1> <TSHFT ON>	Start timeshifting with the current recording Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). The recording should be previously started (please refer <REC 1> command description). Otherwise the respective error message is sent.

<TSHFT 0> <TSHFT OFF>	Stop timeshifting with the current recording Command should be used only in Timeshifting mode (when the current mode is Time Shift). Otherwise the respective error message is sent. Command will not stop the the current recording.
<MPLAY 1> <MPLAY ON>	GUI Recordings Browser menu On request
<MPLAY 0> <MPLAY OFF>	GUI Recordings Browser menu Off request (stops media player) Stop media player file playback Stop timeshifting with the current recording
<MPLAY =s>	Start media player for the given media file s = media file path in format of <GRL>/<GML>/<GAL>/<GFL> command reply (please refer <GRL>,<GML>,<GAL>,<GFL> commands description) or custom media file path in the following format: *<container>;<filename>* where The *<container>* is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive USB = USB drive The *<filename>* is the full file path (including subfolders) to the media file (using '/' char as subfolder delimiter). Examples: #COMMAND: <MPLAY =PVR;HDD1;N24__20171202_123001__Re.ts> #OK or #COMMAND: <MPLAY =PVR;NAS2;3sat__20190117_081500.ts> #OK or #COMMAND: <MPLAY =MOVIE;HDD1;movie1.avi> #OK or #COMMAND: <MPLAY =HDD1;myaudio/mysong1.mp3> #OK or #COMMAND: <MPLAY =HDD1;nonexistent_file.mp4> #ERROR: File not available or #COMMAND: <MPLAY =FILE;USB;subfolder/movie2.mkv> #OK
<MPLAYLIST +=s>	Add the given media file to the media player play queue s = media file path in format of <GRL>/<GML>/<GAL>/<GFL> command reply (please refer <GRL>,<GML>,<GAL>,<GFL> commands description) or custom media file path in the following format: *<container>;<filename>* where The *<container>* is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive

	USB = USB drive The <*filename*> is the full file path (including subfolders) to the media file (using '/' char as subfolder delimiter). Examples: #COMMAND: <MPLAYLIST +=PVR;HDD1;M1__20171203_162430.ts> #OK or #COMMAND: <MPLAYLIST +=MOVIE;NAS1;movie1.avi> #OK or #COMMAND: <MPLAYLIST +=HDD1;myaudio/mysong1.mp3> #OK or #COMMAND: < MPLAYLIST +=HDD1;nonexistent_file.mp4> #ERROR: File not available or #COMMAND: <MPLAYLIST +=USB;subfolder/movie2.mkv> #OK
<MPLAYLIST clear>	Clear the media player play queue Example: #COMMAND: <MPLAYLIST CLEAR> #OK
<MPLAY list>	Start the media player play queue The media player play queue should be previously filled up with the media files (please refer <MPLAYLIST +=s> command description). Examples: #COMMAND: <MPLAY LIST> #OK or #COMMAND: <MPLAY LIST> #ERROR: Empty playlist
<MPLAY LOOP n>	Set media player play queue loop setting n = decimal code for the media player play queue loop setting Possible decimal codes for the media player play queue loop setting: 0 playback loop is disabled 1 playback loop is enabled for the current file 2 playback loop is enabled for the play queue
<MPSTOP>	Stop media player file playback (returns back to Recordings Browser GUI) Stop timeshifting with the current recording
<MPPAUSE> <PAUSE>	Pause file playback in the media player Pause timeshifting
<MPPLAY> <RESUME>	Resume file playback in the media player Resume timeshifting

<MPFF>	Jump 5% (minimum 30 seconds) forward in the current file playback Jump 5% (minimum 30 seconds) forward in timeshifting
<MPFF n>	Jump n minutes forward in the current media player file playback Jump n minutes forward in timeshifting
<MPRW>	Jump 5% (minimum 30 seconds) backward in the current file playback Jump 5% (minimum 30 seconds) backward in timeshifting
<MPRW n>	Jump n minutes backward in the current media player file playback Jump n minutes backward in timeshifting
<MPSTA>	Jump to the start of the current media player file playback Jump to the start of the current timeshifting file Jump to the previous timeshifting file (within 30 seconds from the current timeshifting file start)
<MPMID>	Jump to the middle of the current media player file playback Jump to the middle of the current timeshifting file
<MPEND>	Jump to end of the current media player file playback Jump to the next file in the media player play queue Jump to the end of the current timeshifting file Jump to the next timeshifting file if available
<MPTIME>	GUI media player Timeline request (timeline for the current file playback)
<PLAY s>	Start playback for the stream URL link defined in 's' parameter s = stream URL to play (see the following examples) If the device was already playing some stream, file (including Timeshifting mode) or TV/Radio program it automatically stops the previous playback and starts playing the stream from the new URL. The command also turns the device on in the case of Standby mode. In the case when the IPTV stream with the given URL exists inside the TV or Radio channels list (please refer <CLT ADD URL s>, <CLR ADD URL s> commands), the device will switch to the respective TV/Radio channel. The device will start Media Player mode playback otherwise. Examples: For multicast/broadcast UDP/RTP streams #COMMAND: <PLAY udp://239.35.10.231:60231> #OK #COMMAND: <PLAY rtp://255.255.255.255:1234> #OK For HTTP/RTSP (and other unicast) streams #COMMAND: <PLAY http://10.1.1.55:31339> #OK #COMMAND: <PLAY rtsp://192.168.1.5:5454/some_stream> #OK For file playback from previously connected NAS network drive

	#COMMAND: <PLAY file:///storage/nas1/movies/movie01.mp4> #OK For file playback from custom Samba network file server #COMMAND: <PLAY smb://10.1.1.200/Public/music/song01.mp3> #OK For file playback from USB drive (refer e.g. <GFLFP> to get available files) #COMMAND: <PLAY /storage/usb/subfolder/movie01.mp4> #OK For files playback from previously prepared playlist (refer <PLS ADD s>) #COMMAND: <PLAY PLS> #OK
<MPTS n;s>	Start playback for the service id (SID) defined in 'n' parameter from the MPEG MPTS stream URL link defined in 's' parameter n = service id (SID) in decimal form s = MPEG MPTS stream URL to play (see the following examples) Command should be used for the case of Multi Program MPEG Transport Stream (MPTS) playback source. It will not start playback successfully if the playback source does not provide MPEG TS. If the device was already playing some stream, file (including Timeshifting mode) or TV/Radio program it automatically stops the previous playback and starts playing the stream with the given SID from the new URL. The command also turns the device on in the case of Standby mode. In the case when the IPTV stream with the given URL exists inside the TV or Radio channels list (please refer <CLT ADD URL s>, <CLR ADD URL s> commands), the device will switch to the respective TV/Radio channel. The device will start Media Player mode playback otherwise. Examples: For multicast/broadcast UDP/RTP streams #COMMAND: <MPTS 28721;udp://239.35.10.231:60231> #OK #COMMAND: <MPTS 28721;rtp://255.255.255.255:1234> #OK For HTTP (and other unicast) streams #COMMAND: <MPTS 28721;http://10.1.1.55:31339> #OK
<STOP>	Stop the current playback (and go into Idle mode)
<PLS ADD s>	Add media file URL to the permanent PLS playlist s = media file URL to add (see the following examples) Unlike the media player play queue (refer <MPLAYLIST ...>, <MPLAY list> and <MPLAY LOOP n> commands) the permanent PLS playlist and the appropriate settings are saved across reboots. It's also possible to add URL links to Samba network file server shares and the URLs of non existing media files (there is no control for the file availability) aforehand. <PLAY PLS> command should be used to play the previously prepared permanent PLS playlist (prease refer <PLAY s> command).

	<PLS VIEW> command can be used to view the permanent PLS playlist. Examples: For file from NAS1 network drive (inside movies folder) #COMMAND: <PLS ADD file:///storage/nas1/movies/movie01.mp4> #OK For file from HDD1 drive connected to eSATA (inside movies folder) #COMMAND: <PLS ADD /storage/hdd1/movies/movie02.mp4> #OK For file from custom Samba network file server #COMMAND: <PLS ADD smb://10.1.1.200/Public/music/song01.mp3> #OK For file from USB drive (in filesystem root) #COMMAND: <PLS ADD /storage/usb/song01.mp3> #OK
<PLS CLEAR>	Clear the permanent PLS playlist Example: #COMMAND: <PLS CLEAR> #OK
<PLS LOOPS n>	Set FILE/PLS playback retries/loops setting (for <PLAY s> command) n = [0] | [1 .. 100] - retries/loops setting value n = 0 - infinite playback loop n = 1 .. 100 - playback retries/loops number Example: #COMMAND: <PLS LOOPS 1> #OK
<PLS EOF n>	Enable/Disable 'keep EOF' setting for FILE/PLS playback (via <PLAY s>) n = ON | OFF - keep EOF setting value n = ON – enable keep EOF setting (black screen on EOF) n = OFF – disable keep EOF setting (switch to the last stream/program) When the 'keep EOF' setting is enabled the media player holds the black screen waiting for the other commands after the playback completion. The device is still in Media Player mode and indicates EOF via playback state (please refer <PLAYSTATE> command). When the 'keep EOF' setting is disabled the device starts the last stream/program playback (if available or goes into Idle mode) after the FILE/PLS playback completion.
<PLS RESTORE n>	Set 'restore playback' setting for FILE/PLS playback (via <PLAY s>) n = decimal code of the 'restore playback' for FILE/PLS playback setting Possible decimal codes of the are: 0 restore of the last FILE/PLS playback is disabled 1 restore of the last FILE/PLS playback is always enabled 2 restore of the last FILE/PLS playback is enabled only when there is no stream playback available (i.e. channels list is empty)

<REC RMFILE s>	Revome PVR recording file s = recording file path in format of <GRL> command reply (please refer <GRL> command description) or the file path in the following format: *<container>;<filename>* where the *<container>* is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive In the case when the given file is used within the current recording the following error message is sent: #ERROR: File is used by recording The recording storage should be available (please refer <HDD INFO>, <NAS GET CONFIG n> commands description). Otherwise the respective error message is sent. Examples: #COMMAND: <REC RMFILE PVR;HDD1;R1__20181107_123001__Re.ts> #OK or #COMMAND: <REC RMFILE HDD1;ZDF__20181107_170001__heute.ts> #OK
<REC RMDATE s>	Remove PVR recording files by date (created earlier than the given date) s = date string as a row of the attributes separated with ";" char in the following format: *yyyy;mm;dd;HH;MM* where *yyyy* full year, e.g. 2018 *mm* month as a decimal number (01-12) *dd* day of the month (01-31) *HH* hour in 24h format (00-23) *MM* minute (00-59) The recording storage should be available (please refer <HDD INFO>, <NAS GET CONFIG n> command description). Otherwise the respective error message is sent. Example: #COMMAND: <REC RMDATE 2018;11;07;12;30> #OK
<REC RMDAYS n>	Remove PVR recording files older than the given number of days ago n = number of days (how long ago the recording files should be stored) Note: Command is used to remove recordings manually. Please refer <RECCFG RMDAYS n> command to configure the automatic recordings removal. Example: #COMMAND: <REC RMDAYS 7> #OK

<SETPLAYPOS n>	Set position / Seek inside the current file playback (in seconds) for media player / timeshifting n = [+/-] seconds n without a leading sign – set the absolute position from start. n with a leading sign – relative seek from the current position. Example: #COMMAND: <SETPLAYPOS -120> #OK
<SETPLAYTIME s>	Set position inside the current file playback (as preformatted string) for media player / timeshifting s = absolute position as string in format HH:MM:SS Example: #COMMAND: < SETPLAYTIME 00:02:30> #OK
<TSHFT SETPLAYPOS n>	Set position / Seek inside playback (in seconds) for timeshifting n = [+/-] seconds n without a leading sign – set the absolute position from start. n with a leading sign – relative seek from the current position. The command (unlike <SETPLAYPOS n> command which operates with the currently played file) seeks inside the whole timeshifting with the respects to the possible recordings split. In the case when there was no recordings split the command operates the same like <SETPLAYPOS n> command.
<TSHFT SETPLAYTIME s>	Set position inside playback (as preformatted string) for timeshifting s = absolute position as string in format HH:MM:SS The command (unlike < SETPLAYTIME s> command which operates with the currently played file) sets position inside the whole timeshifting with the respects to the possible recordings split. In the case when there was no recordings split the command operates the same like < SETPLAYTIME s> command.
<TSHFT GOPREV>	Jump to the previous timeshifting file if available Jump to the start of the timeshifting otherwise
<TSHFT GONEXT>	Jump to the next timeshifting file if available Jump to the end of the timeshifting otherwise

<CLT RM n>	Remove TV channel(s) from TV channels list
	
	n= [ALL] | [SCRAMBLED] | [1 .. *channellist-size*]
	n = ALL – remove all TV channels (empty TV channels list)

	n = SCRAMBLED – remove all scrambled TV channels n = 1 .. *channellist-size* – channel number in TV list to remove Examples: #COMMAND: <CLT RM ALL> #OK or #COMMAND: <CLT RM SCRAMBLED> #OK or #COMMAND: <CLT RM 19> #OK
<CLR RM n>	Remove Radio channel(s) from Radio channels list n = [ALL] | [SCRAMBLED] | [1 .. *channellist-size*] n = ALL – remove all Radio channels (empty Radio channels list) n = SCRAMBLED – remove all scrambled Radio channels n = 1 .. *channellist-size* – channel number in Radio list to remove Example: #COMMAND: <CLR RM 81> #OK
<SCAN STOP>	Stop currently running channels search Command should be used only within the search of type 'Auto' or 'Network' (please refer <SCAN STATUS> command description). The search should be currently running. Otherwise the respective error message is sent. Example: #COMMAND: <SCAN STOP> #OK
<SCAN EXIT>	GUI Channel Search menu Off request Stop currently running channels search Command can be used within search of any type. The search should be previously launched (can be already complete). Otherwise the respective error message is sent. Example: #COMMAND: <SCAN EXIT> #OK
<SCAN DVB-S AUTO s>	Request DVB-S automatic channels search s = row of the attributes separated with “;” char in the following format: *sat_pos;scan_type;fta_only;fix_obsolete* where *sat_pos* satellite position, e.g. '19.2E' or '26.0W' *scan_type* channels scan type 0 = TV & Radio 1 = TV Only

	2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs *fix_obsolete* fix (replace) obsolete programs 0 = Off (disabled) 1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-S AUTO 19.2E;0;off;0> #OK
<SCAN DVB-S MANUAL s>	Request DVB-S manual channels search s = row of the attributes separated with ";" char in the following format: *sat_pos;freq;pol;sr;scan_type;fta_only;fix_obsolete* where *sat_pos* satellite position, e.g. '19.2E' or '26.0W' *freq* transponder frequency in MHz, e.g. 10744 *pol* transponder polarization ('H' or 'V') *sr* transponder symbol rate, e.g. 22000 *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs *fix_obsolete* fix (replace) obsolete programs 0 = Off (disabled) 1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-S MANUAL 19.2E;10744;H;22000;1;on;0> #OK
<SCAN DVB-C AUTO s>	Request DVB-C automatic channels search s = row of the attributes separated with ";" char in the following format: *mod;sr;scan_type;fta_only;fix_obsolete* where *mod* modulation ('auto', 'QAM64' or 'QAM256') *sr* symbol rate ('auto', '6111', '6875', '6900' or '6950') *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs

	fix_obsolete fix (replace) obsolete programs 0 = Off (disabled) 1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-C AUTO Auto;Auto;0;on;1> #OK
<SCAN DVB-C MANUAL s>	Request DVB-C manual channels search s = row of the attributes separated with “;” char in the following format: *freq;mod;sr;scan_type;fta_only;fix_obsolete* where *freq* transponder frequency in MHz, e.g. 322 *mod* transponder modulation ('auto', 'QAM16', 'QAM32', 'QAM64', 'QAM128' or 'QAM256') *sr* transponder symbol rate, e.g. 6900 or 'auto' *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs *fix_obsolete* fix (replace) obsolete programs 0 = Off (disabled) 1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-C MANUAL 322;QAM259;6900;2;off;0> #OK
<SCAN DVB-C NETWORK s>	Request DVB-C network (LCN) channels search s = row of the attributes separated with “;” char in the following format: *nid;freq;mod;sr;scan_type;fta_only;fix_obsolete* where *nid* network ID, e.g. 1234 *freq* transponder frequency in MHz, e.g. 322 *mod* transponder modulation ('QAM16', 'QAM32', 'QAM64', 'QAM128' or 'QAM256') *sr* transponder symbol rate, e.g. 6900 *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs *fix_obsolete* fix (replace) obsolete programs 0 = Off (disabled)

	1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-C NETWORK 1234;322;QAM259;6900;0;off;0> #OK
<SCAN DVB-T AUTO s>	Request DVB-T automatic channels search s = row of the attributes separated with “;” char in the following format: *scan_type;fta_only;fix_obsolete* where *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs *fix_obsolete* fix (replace) obsolete programs 0 = Off (disabled) 1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-T 3;on;0> #OK
<SCAN DVB-T MANUAL s>	Request DVB-T manual channels search s = row of the attributes separated with “;” char in the following format: *freq;scan_type;fta_only;fix_obsolete* where *freq* transponder frequency in kHz, e.g. 360000 *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs *fix_obsolete* fix (replace) obsolete programs 0 = Off (disabled) 1 = By Name (replacement by program name) 2 = By TSiD (replacement by ONID:TSID:SID) 3 = By Name & TSiD (both name and DVB triplet) Example: #COMMAND: <SCAN DVB-T MANUAL 360000;0;off;0> #OK

<SCAN IPTV s>	Request IPTV (manual) channels search s = row of the attributes separated with “;” char in the following format: *iptv_proto;iptv_address;iptv_port;scan_type;fta_only* where *iptv_proto* IPTV stream proto ('UDP','RTP','HTTP') *iptv_address* IPTV stream multicast/broadcast/unicast address, e.g. 239.35.10.231 or 10.1.1.55 *iptv_port* IPTV stream port, e.g. 1234 or 8080 *scan_type* channels scan type 0 = TV & Radio 1 = TV Only 2 = Radio Only 3 = All Services *fta_only* scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs Example: #COMMAND: <SCAN IPTV UDP;239.35.10.231;1234;0;off> #OK
<SCAN DAB>	Request DAB band scan (automatic channels search) Example: #COMMAND: <SCAN DAB> #OK
<SCAN DAB OVERWRITE>	Request DAB band scan (automatic channels search) Overwrite (remove) existing DAB programs on search
<SCAN FM>	Request FM band scan (automatic channels search) Example: #COMMAND: <SCAN FM> #OK
<SCAN FM OVERWRITE>	Request FM band scan (automatic channels search) Overwrite (remove) existing FM programs on search
<CMXC n>	Set CPU cooling fan maximum temperature (n=50, 55, 60, 65, 70, 75)
<CMXL n>	Set left cooling fan maximum temperature (n=50, 55, 60, 65, 70, 75)
<CMXR n>	Set right cooling fan maximum temperature (n=50, 55, 60, 65, 70, 75)
<CMNL n>	Set left cooling fan minimum temperature (n=20, 25, 30, 35, 40, 45)
<CMNR n>	Set right cooling fan minimum temperature (n=20, 25, 30, 35, 40, 45)
<CFANL n>	Enable/Disable left cooling fan (n = ON, OFF)

<CFANR n>	Enable/Disable right cooling fan (n = ON, OFF)
<APK START n>	Start APK with the given package name n = APK package name (retrieved e.g. from <APK LIST> command) The respective error message is sent in the case when there is no APK with the given package name available. Example: #COMMAND: <APK START com.netflix.mediaclient> #OK
<APK STOP>	Stop APK / Exit App Mode Command should be used only in App Mode. Otherwise the respective error message is sent. Examples: #COMMAND: <APK STOP> #OK or #COMMAND: <APK STOP> #ERROR: Not running
<APK RC KEY n>	Simulate a keypad keypress for the currently running APK n = keypad key to simulate Possible keypad keys are: LEFT Move Left inside APK RIGHT Move Right inside APK UP Move Up inside APK DOWN Move Down inside APK OK | CENTER | SELECT | CNF Select / confirm inside APK EXIT | BACK Exit / Back key inside APK HOME Simulate Home key (stops APK and exits App Mode) Command should be used only in App Mode. Otherwise the respective error message is sent. Example: #COMMAND: <APK RC KEY LEFT> #OK
<APK RC NKEY s>	Simulate a sequence of keypad keypresses for the currently running APK s = list of string attributes separated with “;” delimiter in format: *key;repeat;delay* where *key* = keypad key to simulate Possible keypad keys are: LEFT Move Left inside APK RIGHT Move Right inside APK UP Move Up inside APK

	DOWN Move Down inside APK OK | CENTER | SELECT | CNF Select / confirm inside APK EXIT | BACK Exit / Back key inside APK *repeat* = number of repeats for the specified key (> 0) *delay* = delay in milliseconds between repeats (>= 0) Command should be used only in App Mode. Otherwise the respective error message is sent. Example: #COMMAND: <APK RC NKEY UP;8;50> #OK
<APK RC MOUSE X n>	Simulate a horizontal mouse move for the currently running APK n = number of units to move the mouse horizontally to the left (n < 0) or to the right (n > 0) The screen size for the mouse simulation is assumed to be 1920x1080 units. Real mouse position on the screen is automatically recalculated in the case when display is configured to some other mode than those with 1920x1080 pixel resolution. The mouse moves to the respective edge of the screen in the case when the given number of units to move n is greater than the screen size. Command should be used only in App Mode. Otherwise the respective error message is sent. Example: #COMMAND: <APK RC MOUSE X 660> #OK
<APK RC MOUSE Y n>	Simulate a vertical mouse move for the currently running APK n = number of units to move the mouse vertically up (n < 0) or down (n > 0) The screen size for the mouse simulation is assumed to be 1920x1080 units. Real mouse position on the screen is automatically recalculated in the case when display is configured to some other mode than those with 1920x1080 pixel resolution. The mouse moves to the respective edge of the screen in the case when the given number of units to move n is greater than the screen size. Command should be used only in App Mode. Otherwise the respective error message is sent. Example: #COMMAND: <APK RC MOUSE Y 680> #OK
<APK RC MOUSE W n>	Simulate a vertical scroll wheel for the currently running APK n = number of units to scroll vertically down (n < 0) or up (n > 0) Command should be used only in App Mode. Otherwise the respective error

	message is sent. Example: #COMMAND: <APK RC MOUSE W -3> #OK
<APK RC MOUSE WH n>	Simulate a horizontal scroll wheel for the currently running APK n = number of units to scroll horizontally to the right (n < 0) or to the left (n > 0) Command should be used only in App Mode. Otherwise the respective error message is sent. Example: #COMMAND: <APK RC MOUSE WH 8> #OK
<APK RC MOUSE BTN n>	Simulate a mouse button press for the currently running APK n = mouse button to simulate Possible mouse button values are: OK | LEFT Left mouse button (Select / confirm inside APK) EXIT | RIGHT Right mouse button (Exit / Back inside APK) Command should be used only in App Mode. Otherwise the respective error message is sent. Examples: #COMMAND: <APK RC MOUSE BTN OK> #OK or #COMMAND: <APK RC MOUSE BTN EXIT> #OK
<APK RC MOUSE HIDE>	Hide mouse cursor pointer Command assumed to be used after the mouse simulation commands sequence. Example: #COMMAND: <APK RC MOUSE HIDE> #OK
<APK RC DELAY n>	Delay during keypad/mouse simulation commands sequence n = delay value in milliseconds Example: #COMMAND: <APK RC DELAY 300> #OK
<WEBVIEW SHOW s>	Enable Ads WebView with the given page URL and parameters

	s = list of string attributes separated with “;” delimiter in format: *video_x;video_y;video_w;video_h;video_crop;page_url* where *video_x* video rect left point in 1920x1080 units *video_y* video rect top point in 1920x1080 units *video_w* video rect width in 1920x1080 units *video_h* video rect height in 1920x1080 units *video_crop* 0 = scale video in the given rect; 1 = crop video *page_url* web page URL The screen size for the given web page is assumed to be 1920x1080 units. The colors transparency is supported by the View to show content above the video (in the case of cropped video). Real size of the scaled video rect is automatically recalculated in the case when display is configured to some other mode than those with 1920x1080 pixel resolution. Ads WebView becomes visible after the page loading is complete. The page loading status can be received via <WEBVIEW ?> command. Example: #COMMAND: <WEBVIEW SHOW 0;0;320;240;0;http://10.1.1.2/ad.html> #OK
<WEBVIEW HIDE>	Disable (hide) Ads WebView
<RSTDEF>	Reset settings (Factory Defaults)
<LIC ADD n>	Add License Manager license n = 16-digit decimal license key Examples: #COMMAND: <LIC ADD 4752181890275954> #OK or #COMMAND: <LIC ADD 4752181890275955> #ERROR: License key is not valid for this device or #COMMAND: <LIC ADD 4752731679643762> #ERROR: License key is already used for this device
<LIC RM n>	Remove existing License Manager license n = 16-digit decimal license key Examples: #COMMAND: <LIC RM 4752181890275954> #OK or #COMMAND: <LIC RM 4752181890275955> #ERROR: License key is not used for this device
<LIC RESET>	Reset (clear) License Manager licenses Example: #COMMAND: <LIC RESET>

	#OK
<LOGLEVEL n>	Set log level (logging verbosity) n = 0 .. 4 - log level value (increasing value = more verbose logging) n = 0 - SILENT (no log messages) n = 1 - ERROR (log errors) n = 2 - WARNING (log errors, warnings and some important messages) n = 3 - INFO (log also info messages, e.g. some details) n = 4 - VERBOSE (verbose logging with very detailed information intended mostly for debug) Note: n = 3 (INFO) should be good enough for the most purposes Example: #COMMAND: <LOGLEVEL 3> #OK
<LOG DATETIME n>	Set log messages date/time info setting n = 0 .. 2 - log messages date/time setting value n = 0 - no date/time info in log messages n = 1 - time only shown in log messages (default setting value) n = 2 - date and time shown in log messages Example: #COMMAND: <LOG DATETIME 1> #OK
<LOG USB CLR>	Clear USB drive logs (remove the existing log files including rotated files)
<LOG MMC CLR>	Clear MMC (Internal Storage) logs (remove the existing log files including rotated files)
<LOG MMC CP>	Copy MMC (Internal Storage) logs into USB drive filesystem root
<LOG MMC MV>	Move MMC (Internal Storage) logs into USB drive filesystem root (last logs including rotated files are removed from MMC Internal Storage)
<WEBAPP REMOVE>	Remove previously installed Custom WebApp GUI Application The device will immediately launch System GUI Application in the case when the Custom WebApp GUI was running. Please refer <WEBAPP CUSTOM DENY n>, <WEBAPP SYNC n s> commands description to temporary deny or update/install Custom WebApp GUI.
<HTDATA REMOVE>	Remove previously installed Custom HTTP Views Please refer <HTDATA DENY n>, <HTDATA SYNC n s> commands description to temporary deny or update/install Custom HTTP Views.
<APPDATA REMOVE s>	Remove all or any of the previously written Custom Application Data s = [*] | [*FILEPATH*] s = * - clear internal application data storage (remove all data)

	s = *FILEPATH* - relative path inside custom application data storage to the data file or folder to be removed Please refer <APPDATA SYNC n s> command description to (re)write custom application data into the storage. Examples: To remove single file from internal application data storage #COMMAND: <APPDATA REMOVE bootvideo.mp4> #OK To remove some folder (e.g. logo) from internal application data storage #COMMAND: <APPDATA REMOVE logo> #OK To clear internal application data storage (remove all data) #COMMAND: <APPDATA REMOVE *> #OK
<RESET APKDATA>	Clear all data associated with the APK packages (including accounts data) Note: Command should not be used within the App Mode. The only allowed modes for the command execution are TV/Radio program playback mode, Timeshifting (when the current mode is Time Shift), Media Player, Idle mode or Standby. Otherwise the respective error message is sent. The command is followed by the device restart. Example: #COMMAND: <RESET APKDATA> #OK

8. Commands with additional return value

Each command starts with "<" char and ends with ">". Immediately after ">" sign is received, command will be performed.

If command is not recognized (for example, if <ABC> command is sent), the following text should appear on your terminal window:

```
#COMMAND: <ABC>
#ERROR: Command not supported
```

If command is supported and was received correctly you should get something like:

```
#COMMAND: <GCS>
#RET: on
#OK
```

Command	**With Return Value**
<VER>	Get device firmware and hardware information Example: #COMMAND: <VER> #Mainboard: Rev.01 #Firmware: Ver.00.01 #S/N: 01928374657410 #OK
<FWINFO>	Get extended device firmware information Example: #COMMAND: <FWINFO> #Firmware: Ver.00.01 build 01 (04.10.2018 11:42) #OK
<UPDATE n s>	Update device firmware n = [NET | USB] - update container (firmware image container) s = list of parameters depending of n (see below) Possible update containers are: 1. n = NET Update from network share (NAS, file server). In this case '*s*' is the list of string attributes separated with “;” delimiter in format: *image_file;server_ip;share_name;username;password* where *image_file* = relative path to firmware image file inside shared folder; *server_ip* = IP-address of the network file server; *share_name* = network shared resource name; *username* = network file server user name or “-” if empty for anonymous (guest) login; *password* = network file server user password or

	“-” if empty (no password); 2. n = USB Update from USD drive. In this case '*s*' is relative path to firmware image file inside the USB drive root filesystem. The operation progress can also be observed during the update process. Device sends progress information as: #RET: *progress_percent* Device sends error message if the update process fails for some reason: #ERROR: *error_description* Examples: To update from network share #COMMAND: <UPDATE NET elium_IRD2160_v00.01.img;10.1.1.5;Public;-;-> #RET: 20% #RET: 40% #RET: 60% #RET: 80% #RET: 100% #OK To update from USB drive #COMMAND: <UPDATE USB elium_IRD2160_v00.01.img> #RET: 20% … #RET: 80% #RET: 100% #OK
<UPDATE ELIUM>	Update device firmware with the latest release via Web The operation progress can be observed during the update process. Device sends progress information as: #RET: *progress_percent* Device sends error message if the update process fails for some reason: #ERROR: *error_description* Example: #COMMAND: <UPDATE ELIUM> #RET: 20% … #RET: 80% #RET: 100% #OK
<IPC> <ETH1 IPC>	Get the current (actual) network configuration for ETH / ETH-1 interface When DHCP is enabled the actual networking configuration parameters (the values automatically obtained via DHCP) are returned. <IPC STATIC> command can be used to check the previously saved static network configuration parameters in this case. Please refer <IPC STATIC> command description. Examples: In the case when DHCP is enabled

	#COMMAND: <IPC> #MACADDR: 00:E0:4C:01:0B:31 #DHCP: on #DHCP IP: 192.168.0.52 #DHCP MASK: 255.255.255.0 #DHCP GW: 192.168.0.1 #DHCP DNS1: 81.173.194.77 #DHCP DNS2: 194.8.194.60 #OK In the case when DHCP is disabled #COMMAND: <IPC> #MACADDR: 00:E0:4C:01:0B:31 #DHCP: off #IP: 10.1.1.55 #MASK: 255.255.255.0 #GW: 10.1.1.3 #DNS1: 10.1.1.3 #DNS2: 8.8.8.8 #OK
<IPC STATIC> <ETH1 IPC STATIC>	Get a static network configuration for ETH / ETH-1 interface When DHCP is enabled the previously saved static network configuration parameters are returned instead of the actual networking configuration. In the case when DHCP is disabled the command acts the same as the <IPC> command. Please refer <IPC> command description. Examples: In the case when DHCP is enabled #COMMAND: <IPC STATIC> #MACADDR: 00:E0:4C:01:0B:31 #DHCP: on #STATIC IP: 10.1.1.55 #STATIC MASK: 255.255.255.0 #STATIC GW: 10.1.1.3 #STATIC DNS1: 10.1.1.3 #STATIC DNS2: 8.8.8.8 #OK In the case when DHCP is disabled the same as for <IPC> command.
<IPC DHCP ?> <ETH1 IPC DHCP ?>	Get DHCP status for a network configuration for ETH / ETH-1 interface Example: #COMMAND: <IPC DHCP ?> #DHCP: on #OK or #COMMAND: <IPC DHCP ?> #DHCP: off #OK
<ETH2 IPC>	Get the current static network configuration for ETH-2 (Control) interface Note: Available only for devices equipped with dual ethernet interfaces (ETH-1 / ETH-2). Otherwise the respective error message is sent.

	Example: #COMMAND: <ETH2 IPC> #MACADDR: 00:E0:4C:02:0B:31 #IP: 192.168.198.210 #MASK: 255.255.255.0 #OK
<DISPLAY MODES>	Get all available video display modes Note: UHD 4:2:0 / UHD 4:2:2 additional licenses are required to activate the respective UHD video display modes. The respective video display modes are not available and omitted in the case when there is no valid license. Example: #COMMAND: <DISPLAY MODES> #RET: 0;PAL #RET: 1;NTSC #RET: 2;480i 60Hz #RET: 3;576i 50Hz #RET: 4;480p 60Hz #RET: 5;576p 50Hz #RET: 6;720p 50Hz #RET: 7;720p 60Hz #RET: 8;1080i 50Hz #RET: 9;1080i 60Hz #RET: 10;1080p 24Hz #RET: 11;1080p 50Hz #RET: 12;1080p 60Hz #RET: 13;2160p 50Hz 420 #RET: 14;2160p 60Hz 420 #RET: 15;2160p 24Hz 422 #RET: 16;2160p 25Hz 422 #RET: 17;2160p 30Hz 422 #OK
<DISPLAY MODE n>	Set video display mode (as string value) Get video display mode (as string value) n = [?] | [*mode_string*] n = ? - get current video display mode as string n = *mode_string* - set video mode with the string value Possible string values for video display mode are: PAL NTSC 480i 60Hz 576i 50Hz 480p 60Hz 576p 50Hz 720p 50Hz 720p 60Hz 1080i 50Hz 1080i 60Hz 1080p 24Hz 1080p 50Hz 1080p 60Hz

	2160p 50Hz 420 2160p 60Hz 420 2160p 24Hz 422 2160p 25Hz 422 2160p 30Hz 422 Note: UHD 4:2:0 / UHD 4:2:2 additional licenses are required to activate the respective UHD video display modes. The respective error message is sent in the case when there is no valid license. Examples: #COMMAND: <DISPLAY MODE ?> #RET: 1080p 60Hz #OK or #COMMAND: <DISPLAY MODE 2160p 60Hz 420> #RET: 2160p 60Hz 420 #OK
<GDM>	Get video display mode (decimal code and string value) Example: #COMMAND: <GDM> #RET: 12;1080p 60Hz #OK
<AR MODES>	Get all available video aspect ratio adjustment modes Example: #COMMAND: <AR MODES> #RET: 0;Auto #RET: 1;Full Stretch #RET: 2;4:3 #RET: 3;16:9 #RET: 4;Non-Linear #RET: 5;Original Size #RET: 6;4:3 Ignore #RET: 7;4:3 Letterbox #RET: 8;4:3 Pan&Scan #RET: 9;4:3 Combined #RET: 10;16:9 Ignore #RET: 11;16:9 Letterbox #RET: 12;16:9 Pan&Scan #RET: 13;16:9 Combined #OK
<AR MODE n>	Set video aspect ratio adjustment mode (as string value) Get video aspect ratio adjustment mode (as string value) n = [?] | [*mode_string*] n = ? - get current aspect ratio adjustment mode as string n = *mode_string* - set aspect ratio adjustment mode with the string value Possible string values for video aspect ratio adjustment mode are: Auto Full Stretch

	4:3 16:9 Non-Linear Original Size 4:3 Ignore 4:3 Letterbox 4:3 Pan&Scan 4:3 Combined 16:9 Ignore 16:9 Letterbox 16:9 Pan&Scan 16:9 Combined Examples: #COMMAND: <AR MODE ?> #RET: Auto #OK or #COMMAND: <AR MODE Full Stretch> #RET: Full Stretch #OK
<GARM>	Get video aspect ratio adjustment mode (decimal code and string value) Example: #COMMAND: <GARM> #RET: 12;16:9 Pan&Scan #OK
<GCV>	Get current volume (mute state and volume level) Example: #COMMAND: <GCV> #RET: on;100 #OK
<VOL n>	Set/change volume (mute state or volume level) Get current volume (mute state and volume level) n = [?] | [+/-][0 .. 100] | [ON,OFF] n = ? - get current volume setting (same as <GCV> command) n = ON or OFF - turn audio on or off (on for unmute, off for mute) n = [+/-][0 .. 100] - without a leading sign sets the volume absolute, with a leading sign sets the volume relative to the current value Examples: #COMMAND: <VOL ?> #RET: on;100 #OK or #COMMAND: <VOL -10> #RET: on;90 #OK or #COMMAND: <VOL 50> #RET: on;50

	#OK or #COMMAND: <VOL OFF> #RET: off;50 #OK
<VOLDB n>	Set audio volume level in dB n = [?] | [-25 .. +25] n = ? - get current audio volume level in dB n = [-25 .. +25] – set audio volume level in dB Examples: #COMMAND: <VOLDB ?> #RET: 0.0 dB #OK or #COMMAND: <VOLDB +10.5> #RET: 10.5 dB #OK or #COMMAND: <VOLDB -5> #RET: -5.0 dB #OK or #COMMAND: <VOLDB -26> #RET: off #OK
<VOLLCK n>	Lock/Unlock volume settings (both volume level and mute state) Get Lock status for volume settings (both volume level and mute state) n = [?] | [ON,OFF] n = ? - get Lock status (on = both volume level and mute are locked, off = at least one setting is unlocked) n = ON or OFF – lock or unlock volume (level and mute) settings Both level and mute settings are locking conjunctively. Each setting can be locked/unlocked separately with the below commands. Example: #COMMAND: < VOLLCK ON> #RET: on #OK
<VOLLEVELLCK n>	Lock/Unlock volume mute setting Get Lock status for volume mute setting n = [?] | [ON,OFF] n = ? - get Lock status (on = volume mute locked, off = unlocked) n = ON or OFF – lock or unlock volume mute setting Example: #COMMAND: < VOLLEVELLCK OFF> #RET: off #OK

<VOLMUTELCK n>	Lock/Unlock volume level setting change Get Lock status for volume level setting n = [?] | [ON,OFF] n = ? - get Lock status (on = volume level locked, off = unlocked) n = ON or OFF – lock or unlock volume level setting Example: #COMMAND: < VOLMUTELCK ?> #RET: on #OK
<LSD n>	Set audio lipsync delay adjustment value Get current audio lipsync delay adjustment value n = [?] | [+/-][1 .. 4500] | 0 n = ? - get current audio lipsync delay adjustment value n = + 1 .. 4500 - set audio lipsync delay (audio is later) n = - 1 .. 4500 - set audio lipsync delay (audio is earlier) n = 0 - disable audio lipsync delay adjustment Examples: #COMMAND: <LSD ?> #RET: +500 #OK or #COMMAND: <LSD -300> #RET: -300 #OK or #COMMAND: <LSD +300> #RET: +300 #OK or #COMMAND: <LSD 0> #RET: 0 #OK
<AUDIO PT AC3 n>	Enable/Disable AC3 audio passthrough Get AC3 audio passthrough status n = [?] | [ON,OFF] n = ? - get AC3 audio passthrough status (on = enabled, off = disabled) n = ON or OFF – enable or disable AC3 audio passthrough Note: AC3 encoded bitstream passthrough is available only via HDMI/SPDIF interfaces and requires additional receiver able to decode the data. Example: #COMMAND: <AUDIO PT AC3 ON> #RET: on #OK
<AUDIO PT DTS n>	Enable/Disable DTS audio passthrough Get DTS audio passthrough status

	n = [?] | [ON,OFF] n = ? - get DTS audio passthrough status (on = enabled, off = disabled) n = ON or OFF – enable or disable DTS audio passthrough Note: DTS encoded bitstream passthrough is available only via HDMI/SPDIF interfaces and requires additional receiver able to decode the data. Example: #COMMAND: <AUDIO PT DTS ?> #RET: off #OK
<SDI AC s n>	Enable/Disable SDI audio channels Get SDI audio channels status (enabled/disabled) s = [1+2] | [3+4] | [5+6] | [7+8] – selected SDI audio channels n = [?] | [ON,OFF] n = ? - get selected SDI audio channels status n = ON – enable (embed) selected SDI audio channels n = OFF – disable (do not embed) selected SDI audio channels Note: Available only for SDI Backend Boards. Otherwise the respective error message is sent. Examples: #COMMAND: <SDI AC 1+2 ?> #RET: on #OK or #COMMAND: <SDI AC 3+4 ON> #RET: on #OK or #COMMAND: <SDI AC 7+8 OFF> #RET: off #OK
<LCK n>	Lock/Unlock Frontpanel Keys Get Lock status for Frontpanel Keys n = [?] | [ON,OFF] | [1,0] n = ? - get Lock status (on = locked, off = unlocked) n = ON or 1 - lock Frontpanel Keys n = OFF or 0 - unlock Frontpanel Keys Example: #COMMAND: <LCK ON> #RET: on #OK

<LCI n>	Lock/Unlock IR Remote
	Get Lock status for IR Remote
	
	n = [?] | [ON,OFF] | [1,0]
	n = ? - get Lock status (on = locked, off = unlocked)
	n = ON or 1 - lock IR Remote

	n = OFF or 0 - unlock IR Remote Example: #COMMAND: <LCI ?> #RET: off #OK

<VRMC n>	Enable/Disable Virtual IR Remote Control (VRMC) mode
	Get VRMC mode status (enabled/disabled)
	
	n = [?] | [ON,OFF] | [1,0]
	n = ? - get VRMC mode status (on = enabled, off = disabled)
	n = ON or 1 - enable VRMC mode
	n = OFF or 0 - disable VRMC mode
	
	Enabling VRMC mode is suitable to simulate an input via remote control within one char. There is no need to send any command to simulate the input in such case. The input simulation will be performed immediately after the respective char is received.
	
	Please refer <RMC c> command for the possible VRMC chars.

<IR ADDR n>	Set IR Remote Control address Get current IR Remote Control address n = [?] | [OFF] | [0] | [1 .. 4] n = ? - get current IR Remote Control address (off or 1 .. 4) n = OFF or 0 - disable IR Remote address check (any IR Remote accepted) n = 1 .. 4 - only IR Remote with the same address accepted Examples: #COMMAND: <IR ADDR ?> #RET: off #OK or #COMMAND: <IR ADDR 4> #RET: 4 #OK
<GCS>	Get current device state (On or Standby) Possible return: #RET: on device is On (active) #RET: off device is in Standby
<GCM>	Get current device mode Possible return: #RET: Restart Device is booting up #RET: Standby Device is in standby #RET: Setup Menu Device is in setup menu #RET: Idle Device is idle (no playback) #RET: TV TV program playback #RET: Radio Radio program playback #RET: Time Shift Timeshifting mode with the current recording

	#RET: Media Player Media Player mode with the file or stream playback #RET: Search Searching for the channels #RET: Firmware Update Updating firmware #RET: HDD Format Formatting HDD #RET: App Mode Running APK
<TTT>	Turn to TV mode Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode when TV channels are available. Otherwise the respective error message is sent. Example: #COMMAND: <TTT> #RET: TV #OK
<TTR>	Turn to Radio mode Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode when Radio channels are available. Otherwise the respective error message is sent. Example: #COMMAND: <TTR> #RET: Radio #OK
<PRT U>	Switch TV channel Up (current channel -1) Command should be used only in TV mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRT U> #RET: TV;51;Sky Sport News #OK
<PRT D>	Switch TV channel Down (current channel +1) Command should be used only in TV mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRT D> #RET: TV;52;tagesschau24 HD #OK
<PRR U>	Switch Radio channel Up (current channel -1) Command should be used only in Radio mode. Otherwise the respective error message is sent.

	Example: #COMMAND: <PRR U> #RET: Radio;25;MDR KLASSIK #OK
<PRR D>	Switch Radio channel Down (current channel +1) Command should be used only in Radio mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRR D> #RET: Radio;26;NDR Kultur #OK
<PRT =pn>	Switch to TV channel whose name is equal with pn Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRT =tagesschau24 HD> #RET: TV;52;tagesschau24 HD #OK
<PRT *pn>	Switch to TV channel whose name contains pn Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRT *Sport News> #RET: TV;51;Sky Sport News #OK
<PRR =pn>	Switch to Radio channel whose name is equal with pn Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRR =NDR Kultur> #RET: Radio;26;NDR Kultur #OK
<PRR *pn>	Switch to Radio channel whose name contains pn Command should be used only in TV/Radio program playback (when the

	current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRR *Kultur> #RET: Radio;26;NDR Kultur #OK
<PRT n>	Switch to TV channel by number (n is a number) n = channel number in TV list (received e.g. from <GCL> command) Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRT 11> #RET: TV;11;Das Erste HD #OK
<PRR n>	Switch to Radio channel by number (n is a number) n = channel number in Radio list (received e.g. from <GCL> command) Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Example: #COMMAND: <PRR 32> #RET: Radio;32;NDR Plus #OK
<GNT>	Get number of TV channels Example: #COMMAND: <GNT> #RET: 1170 #OK
<GNR>	Get number of Radio channels Example: #COMMAND: <GNR> #RET: 162 #OK
<GCC>	Get current channel (channel number) Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) or in Timeshifting mode (when the current

	mode is Time Shift). Otherwise the respective error message is sent. Example: #COMMAND: <GCC> #RET: 19 #OK
<GCP>	Get current program (name of program) Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) or in Timeshifting mode (when the current mode is Time Shift). Otherwise the respective error message is sent. Example: #COMMAND: <GCP> #RET: SES UHD Demo Channel #OK
<GCL>	Get channel list (list of TV and Radio programs including numbering) Each table row is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent. Example: #COMMAND: <GCL> #RET: TV;1;tagesschau24 #RET: TV;2;ONE #RET: TV;3;arte #RET: TV;4;PHOENIX ... #RET: TV;1169;Sky Sport Bundesliga 7 HD #RET: TV;1170;BBC World News Europe HD #RET: Radio;1;DKULTUR #RET: Radio;2;DLF #RET: Radio;3;DRadio Wissen ... #RET: Radio;161;F. INTER #RET: Radio;162;CULTURE #END #OK
<GCLPT n m>	Get the part of the current TV channels list (including numbering) n = the program number to start from m = the total (maximum) number of programs to get from start Example: #COMMAND: <GCLPT 10 2> #RET: TV;10;ZDFinfo #RET: TV;11;Das Erste HD #END #OK
<GCLPR n m>	Get the part of the current Radio channels list (including numbering)

	n = the program number to start from m = the total (maximum) number of programs to get from start Example: #COMMAND: <GCLPR 161 1000> #RET: Radio;161;F. INTER #RET: Radio;162;CULTURE #END #OK
<GCPMUX>	Get current channel (including type, number, name, mux and TS info) Return format: *type;number;muxinfo;SID;ONID;TSID;name* where *type* channel type (either TV or Radio) *number* channel number *muxinfo* mux info string depending from delivery system (please refer examples below) *SID* service id / DAB service id / FM RDS PI (dec) *ONID* original network id / DAB component id *TSID* transport stream id / DAB ensemble id *name* name of program Examples: #COMMAND: <GCPMUX> #RET: TV;2;DVB-S 19.2E 10744 H 22000;28722;1;1051;ONE #OK or #COMMAND: <GCPMUX> #RET: TV;800;DVB-C 322 QAM256 6900;28721;1;1051;tagesschau24 #OK or #COMMAND: <GCPMUX> #RET: TV;900;DVB-T 514000 kHz;769;8468;27136;Das Erste HD #OK or #COMMAND: <GCPMUX> #RET: TV;950;IPTV TS udp://239.35.10.55:1234;762;0;0;Das Erste #OK or #COMMAND: <GCPMUX> #RET: Radio;5;IPTV URL http://panel.nadaje.com:1038;0;0;0;BIZZ.fm #OK or #COMMAND: <GCPMUX> #RET: Radio;180;DAB 178352 kHz;28368;6540;3525;Absolut relax #OK or #COMMAND: <GCPMUX> #RET: Radio;190;FM 102400 kHz;54161;0;0;1LIVE #OK
<GCLMUX>	Get channel list (including type, numbering, name, mux and TS info) Each table row is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent.

	The return row format is the same as for <GCPMUX> command above. Example: #COMMAND: <GCLMUX> #RET: TV;1;DVB-S 19.2E 10744 H 22000;28721;1;1051;tagesschau24 #RET: TV;2;DVB-S 19.2E 10744 H 22000;28722;1;1051;ONE #RET: TV;3;DVB-S 19.2E 10744 H 22000;28724;1;1051;arte #RET: TV;4;DVB-S 19.2E 10744 H 22000;28725;1;1051;PHOENIX … #RET: Radio;1;DVB-S 19.2E 11954 H 27500;28012;1;1079;DKULTUR #RET: Radio;2;DVB-S 19.2E 11954 H 27500;28013;1;1079;DLF … #RET: Radio;162;DVB-S 19.2E 12363 V 27500;9158;1;1098;CULTURE … #RET: Radio;180;DAB 178352 kHz;28368;6540;3525;Absolut Relax … #RET: Radio;190;FM 102400 kHz;54161;0;0;1LIVE #END #OK
<GCLMUXPT n m>	Get the part of the current TV channels list (including type, numbering, name, mux and TS info) n = the program number to start from m = the total (maximum) number of programs to get from start The return row format is the same as for <GCPMUX> command above. Example: #COMMAND: <GCLMUXPT 10 2> #RET: TV;10;DVB-S 19.2E 11954 H 27500;28011;1;1079;ZDFinfo #RET: TV;11;DVB-S 19.2E 11494 H 22000;10301;1;1019;Das Erste HD #END #OK
<GCLMUXPR n m>	Get the part of the current Radio channels list (including type, numbering, name, mux and TS info) n = the program number to start from m = the total (maximum) number of programs to get from start The return row format is the same as for <GCPMUX> command above. Example: #COMMAND: <GCLMUXPR 161 1000> #RET: Radio;161;DVB-S 19.2E 12363 V 27500;9157;1;1098;F. INTER #RET: Radio;162;IPTV URL http://panel.nadaje.com:1038;0;0;0;BIZZ.fm #RET: Radio;163;DAB 178352 kHz;28367;6540;3525;1LIVE #RET: Radio;164;FM 102400 kHz;54161;0;0;1LIVE #END #OK
<PRMUX s>	Switch to TV/Radio channel by mux info (including SID) s = row of the attributes (retrieved e.g. from <GCLMUX> command)

	separated with “;” char in the following format: *muxinfo;SID* where *muxinfo* mux info string depending from delivery system (please refer <GCPMUX>, <GCLMUX> commans) *SID* service id / DAB service id / FM don't care (0) Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio), in Timeshifting (when the current mode is Time Shift), Media Player or Idle mode. Otherwise the respective error message is sent. Examples: #COMMAND: <PRMUX DVB-S 19.2E 11954 H 27500;28011> #RET: TV;10;ZDFinfo #OK or #COMMAND: <PRMUX DVB-S 19.2E 12363 V 27500;9158> #RET: Radio;162;CULTURE #OK or #COMMAND: <PRMUX IPTV TS udp://239.35.10.55:1234;762> #RET: TV;950;Das Erste #OK or #COMMAND: <PRMUX DAB 178352 kHz;28367> #RET: Radio;163;1LIVE #OK or #COMMAND: <PRMUX FM 102400 kHz;0> #RET: Radio;164;1LIVE #OK
<PRTMUX s>	Switch to TV channel by mux info (including SID) s = row of the attributes (retrieved e.g. from <GCLMUX> command) separated with “;” char in the following format: *muxinfo;SID* where *muxinfo* mux info string depending from delivery system *SID* service id Example: #COMMAND: <PRTMUX DVB-S 19.2E 11954 H 27500;28011> #RET: TV;10;ZDFinfo #OK
<PRRMUX s>	Switch to Radio channel by mux info (including SID) s = row of the attributes (retrieved e.g. from <GCLMUX> command) separated with “;” char in the following format: *muxinfo;SID* where *muxinfo* mux info string depending from delivery system *SID* service id / DAB service id / FM don't care (0) Example:

	#COMMAND: <PRRMUX DVB-S 19.2E 12363 V 27500;9158> #RET: Radio;162;CULTURE #OK or #COMMAND: <PRRMUX DAB 178352 kHz;28367> #RET: Radio;163;1LIVE #OK or #COMMAND: <PRRMUX FM 102400 kHz;0> #RET: Radio;164;1LIVE #OK
<PRTS s>	Switch to TV/Radio channel by TS info (including delivery system) s = row of the attributes (retrieved e.g. from <GCLMUX> command) separated with “;” char in the following format: *delsys;SID;ONID;TSID* where *delsys* delivery system ('DVB-S','DVB-C', 'DVB-T', 'DAB' or 'FM') *SID* service id / DAB service id / FM RDS PI (dec) *ONID* original network id / DAB component id FM don't care (0) *TSID* transport stream id / DAB ensemble id FM don't care (0) Examples: #COMMAND: <PRTS DVB-S;28011;1;1079> #RET: TV;10;ZDFinfo #OK or #COMMAND: <PRTS DVB-S;9158;1;1098> #RET: Radio;162;CULTURE #OK or #COMMAND: <PRTS DAB;28367;6540;3525> #RET: Radio;163;1LIVE #OK or #COMMAND: <PRTS FM;54161;0;0> #RET: Radio;164;1LIVE #OK
<PRTTS s>	Switch to TV channel by TS info (including delivery system) s = row of the attributes (retrieved e.g. from <GCLMUX> command) separated with “;” char in the following format: *delsys;SID;ONID;TSID* where *delsys* delivery system ('DVB-S','DVB-C' or 'DVB-T') *SID* service id *ONID* original network id *TSID* transport stream id Example: #COMMAND: <PRTTS DVB-S;28011;1;1079> #RET: TV;10;ZDFinfo

	#OK
<PRRTS s>	Switch to Radio channel by TS info (including delivery system) s = row of the attributes (retrieved e.g. from <GCLMUX> command) separated with “;” char in the following format: *delsys;SID;ONID;TSID* where *delsys* delivery system ('DVB-S','DVB-C', 'DVB-T', 'DAB' or 'FM') *SID* service id / DAB service id / FM RDS PI (dec) *ONID* original network id / DAB component id FM don't care (0) *TSID* transport stream id / DAB ensemble id FM don't care (0) Example: #COMMAND: <PRRTS DVB-S;9158;1;1098> #RET: Radio;162;CULTURE #OK or #COMMAND: <PRRTS DAB;28367;6540;3525> #RET: Radio;163;1LIVE #OK or #COMMAND: <PRRTS FM;54161;0;0> #RET: Radio;164;1LIVE #OK
<GCPEXT>	Get extended info over current channel (including type, number, name, mux and TS info, scrambled status, volume and lipsync adjustment, provider or DAB/FM PTY) Return format: *type;number;muxinfo;SID;ONID;TSID;name;scram;vol;lsd;provider* where *type* channel type (either TV or Radio) *number* channel number *muxinfo* mux info string depending from delivery system (please refer examples below) *SID* service id / DAB service id / FM RDS PI (dec) *ONID* original network id / DAB component id *TSID* transport stream id / DAB ensemble id *name* name of program (or “-” if empty) *scram* FTA/Scrambled status (0 = free, 1 = scrambled) *vol* volume adjustment value (0 = no adjustment) *lsd* lipsync adjustment value (0 = no adjustment) *provider* channel provider name / DAB PTY string / FM PTY string (or “-” if empty) Examples: #COMMAND: <GCPEXT> #RET: TV;2;DVB-S 19.2E 10744 H 22000;28722;1;10;ONE;0;+10;0;ARD #OK or #COMMAND: <GCPEXT> #RET: TV;80;DVB-C 322 QAM256 6900;280;1;1;RTL HD;1;-20;+300;SES

	#OK or #COMMAND: <GCPEXT> #RET: TV;900;DVB-T 514000 kHz;769;84;272;-;1;0;-500;ZDFmobil #OK or #COMMAND: <GCPEXT> #RET: TV;950;IPTV TS udp://239.35.10.55:1234;76;0;0;SR 2;0;0;0;- #OK or #COMMAND: <GCPEXT> #RET: Radio;5;IPTV URL http://st.fm.com:1038;0;0;0;ST.fm;0;+20;0;- #OK or #COMMAND: <GCPEXT> #RET: Radio;180;DAB 222064 kHz;54164;14;4332;WDR 4;0;-24;0;Varied #OK or #COMMAND: <GCPEXT> #RET: Radio;190;FM 90700 kHz;54164;0;0;WDR 4;0;0;0;Oldies Music #OK
<GCLEXT>	Get extended info over channel list (including type, number, name, mux and TS info, scrambled status, volume and lipsync adjustment, provider or DAB/FM PTY) Each table row is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent. The return row format is the same as for <GCPEXT> command above.
<GCPPIDS>	Get current channel PIDs info (including type, number, name, mux and TS info, scrambled status, pmt/video/audio pids info) Return format: *type;number;muxinfo;SID;ONID;TSID;name;scram;PP;VP;VT;AP;AT;AL* where *type* channel type (either TV or Radio) *number* channel number *muxinfo* mux info string depending from delivery system *SID* service id / DAB service id / FM RDS PI (dec) *ONID* original network id / DAB component id *TSID* transport stream id / DAB ensemble id *name* name of program (or “-” if empty) *scram* FTA/Scrambled status (0 = free, 1 = scrambled) *PP* PMT PID as decimal (or “-” if not available) *VP* video stream PID as decimal (or “-” if not available) *VT* video stream type: MPEG-1/MPEG-2/H.264/HEVC etc (or “-” if not available) *AP* audio stream PID as decimal (or “-” if not available) *AT* audio stream type: MPGA/AC3/AAC/AAC+/MP3 etc (or “-” if not available) *AL* audio stream language code: deu, eng, fra, nor etc (or “-” if not available)

	Example: #COMMAND: <GCPPIDS> #RET: TV;1;DVB-T 514000 kHz;76;1;1;ONE;0;30;31;HEVC;32;AC3;deu #OK
<CLCFG LCN n>	Enable/Disable channellist LCN Numbering setting Get current status of channellist LCN Numbering setting n = [?] | [ON,OFF] n = ? – get LCN Numbering setting status (on = enabled, off = disabled) n = ON – enable LCN Numbering (the spaces inside the list are allowed) n = OFF – disable LCN Numbering (programs are enumerated in a row without any spaces) Disabling LCN Numbering will immediately renumber the TV/Radio channels to remove the spaces.
<CLT ADD URL s>	Add custom IPTV stream URL to TV channels list s = row of the attributes separated with “;” char in the following format: *name;stream_url* where *name* TV channel name *STRING (SIZE (1..32))* *stream_url* IPTV stream URL *STRING (SIZE (1..127))* Example: #COMMAND: <CLT ADD URL TLC;http://s-hls.iptv.net/hls/tlc.m3u8> #RET: TV;15;TLC #OK
<CLR ADD URL s>	Add custom Internet Radio stream URL to Radio channels list s = row of the attributes separated with “;” char in the following format: *name;stream_url* where *name* Radio channel name *STRING (SIZE (1..32))* *stream_url* Internet Radio stream URL *STRING (SIZE (1..127))* Example: #COMMAND: <CLR ADD URL Beats FM;http://send.blackbeats.fm:13000> #RET: Radio;34;Beats FM #OK
<CLT SETNAME s>	Set/Edit TV channel name s = row of the attributes separated with “;” char in the following format: *number;name* where *number* TV channel number *1 .. channellist-size*

	name TV channel name *STRING (SIZE (1..32))* Example: #COMMAND: <CLT SETNAME 15;TLC HD> #RET: TV;15;TLC HD #OK
<CLR SETNAME s>	Set/Edit Radio channel name s = row of the attributes separated with ";" char in the following format: *number;name* where *number* Radio channel number *1 .. channellist-size* *name* Radio channel name *STRING (SIZE (1..32))* Example: #COMMAND: <CLR SETNAME 34;Black Beats FM> #RET: Radio;34;Black Beats FM #OK
<CLT SETVOL s>	Set/Edit TV channel audio volume adjustment value s = row of the attributes separated with ";" char in the following format: *number;vol* where *number* TV channel number *1 .. channellist-size* *vol* volume adjustment value +/- 1 .. 99 | 0 (no adjustment) Current audio volume adjustment value for each channel can be observed with <GCLEXT> and <GCPEXT> commands. Please refer the respective commands description. Examples: #COMMAND: <CLT SETVOL 15;-10> #RET: TV;15;TLC HD;-10 #OK or #COMMAND: <CLT SETVOL 15;0> #RET: TV;15;TLC HD;0 #OK
<CLR SETVOL s>	Set/Edit Radio channel audio volume adjustment value s = row of the attributes separated with ";" char in the following format: *number;vol* where *number* Radio channel number *1 .. channellist-size* *vol* volume adjustment value +/- 1 .. 99 | 0 (no adjustment)

	Current audio volume adjustment value for each channel can be observed with <GCLEXT> and <GCPEXT> commands. Please refer the respective commands description. Example: #COMMAND: <CLR SETVOL 34;+23> #RET: Radio;34;Black Beats FM;+23 #OK
<CLT SETLIPSYNC s>	Set/Edit TV channel audio lipsync delay adjustment value s = row of the attributes separated with ";" char in the following format: *number;lsd* where *number* TV channel number *1 .. channellist-size* *lsd* lipsync delay adjustment value +/- 1 .. 4500 | 0 (no adjustment) Current audio lipsync delay adjustment value for each channel can be observed with <GCLEXT> and <GCPEXT> commands. Please refer the respective commands description. If set, the channel lipsync asjustment value takes priority over the global audio lipsync delay adjustment setting (please, refer <LSD n> command). Examples: #COMMAND: <CLT SETLIPSYNC 15;-1000> #RET: TV;15;TLC HD;-1000 #OK or #COMMAND: <CLT SETLIPSYNC 15;+300> #RET: TV;15;TLC HD;+300 #OK or #COMMAND: <CLT SETLIPSYNC 15;0> #RET: TV;15;TLC HD;0 #OK
<CLR SETLIPSYNC s>	Set/Edit Radio channel audio lipsync delay adjustment value The command syntax is the same as for <CLT SETLIPSYNC s> command. Note: Lipsync adjustment makes sense only if both audio and video streams present and has no effect otherwise.
<CLT MOVE s>	Set/Edit TV channel number (move TV channel in the list) s = row of the attributes separated with ";" char in the following format: *number;set_number* where *number* TV channel number *set_number* assigned TV channel number Example: #COMMAND: <CLT MOVE 15;1> #RET: TV;1;TLC HD

	#OK
<CLR MOVE s>	Set/Edit Radio channel number (move Radio channel in the list) s = row of the attributes separated with ";" char in the following format: *number;set_number* where *number* Radio channel number *set_number* assigned Radio channel number Example: #COMMAND: <CLR MOVE 34;30> #RET: Radio;30;Black Beats FM #OK
<SCAN STATUS>	Get current/last channels search status Return format: *state;delsys;scantype;progress;*TV *total/ins;*Radio *total/ins* where *state* Scan state on = search is enabled (currently running) off = search is disabled (complete/not running) *delsys* Scanned delivery system DVB-S, DVB-C, DVB-T, IPTV, DAB or FM *scantype* Scan type Auto = Automatic Search Manual = Manual Search Network = DVB-C Network/LCN Search *progress* Scan progress in percent TV *total/ins* Found TV channels *total* = total found channels number *ins* = number of the inserted (new) channels Radio *total/ins* Found Radio channels *total* = total found channels number *ins* = number of the inserted (new) channels Command shows the status of the currently running channels search or the last complete channels search. Examples: When the search is currently running #COMMAND: <SCAN STATUS> #RET: on;DVB-S;Auto;40%;TV 400/42;Radio 120/20 #OK or #COMMAND: <SCAN STATUS> #RET: on;DAB;Auto;94%;TV 0/0;Radio 26/13 #OK When the search is complete #COMMAND: <SCAN STATUS> #RET: off;DVB-C;Manual;100%;TV 6/0;Radio 4/4 #OK When there was no search before #COMMAND: <SCAN STATUS> #RET: off;-;-;0%;TV 0/0;Radio 0/0 #OK

<SCAN PRT n>	Get found TV channels within the current/last search n = the channel number to start from Each table row is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent. Example: #COMMAND: <SCAN PRT 1> #RET: 1;tagesschau24 #RET: 2;ONE #RET: 3;arte #RET: 4;PHOENIX ... #RET: 1169;Sky Sport Bundesliga 7 HD #RET: 1170;BBC World News Europe HD #END #OK
<SCAN PRR n>	Get found Radio channels within the current/last search n = the channel number to start from Each table row is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent. Example: #COMMAND: <SCAN PRR 14> #RET: 14;Absolute Relax #RET: 15;SCHLAGERPARADIES #END #OK
<GAC>	Get available audio streams for the current channel Each table row is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent. Each return row is send in the following format: *apid* (*aformat*) - *adesc* where *apid* audio stream ID *aformat* audio format (MPGA, AC3, AAC, AAC+, DTS etc) *adesc* audio language or description (optional) Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) or in Timeshifting mode (when the current mode is Time Shift). Otherwise the respective error message is sent. Example: #COMMAND: <GAC> #RET: 5102 (MPGA) - German #RET: 5103 (MPGA) - Miscellaneous #RET: 5106 (AC3) - German #END #OK

<GCA>	Get current audio stream for the current channel The return row format is the same as for <GAC> command above. Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) or in Timeshifting mode (when the current mode is Time Shift). Otherwise the respective error message is sent. Example: #COMMAND: <GCA> #RET: 5102 (MPGA) - German #OK
<EVT n>	Get available EPG events for TV channel by number (n is a number) The information for each event is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent. The event information for every event consists of: - channel type (TV) - channel number - name of program - Event ID - date and time of the event in format dd/mm/yy HH:MM - duration of the event (in minutes) - title of the event - FSK decimal code Note: The data for EPG events cache is taken from Transport Stream (for currently used transponder only). Therefore, host should wait some time and use channels from the same transponder before sending <EVT> command for the selected channel (typically few minutes for the next several days). Otherwise, not all EPG events may be stored in the cache. Example: #COMMAND: <EVT 3> #RET: TV;3;arte;295830206700322970;02/08/17 06:00;1440;ARTE;0 #RET: TV;3;arte;295830206700353332;02/08/17 17:40;45;Xenius;0 … #RET: TV;3;arte;295830206700353912;16/08/17 16:40;25;Journal;0 #RET: TV;3;arte;295830206700353790;16/08/17 17:05;30;360°;0 #END #OK
<EVR n>	Get available EPG events for Radio channel by number (n is a number) The answer format is the same like the above <EVT> command except that the event information for every event contains - channel type (Radio) instead of (TV). All restrictions are also the same as for <EVT> command.
<EVDESC n>	Get detailed description for the selected EPG event. n = Event ID of the selected EPG event Available events (Event IDs) can be retrieved by <EVT> or <EVR> command (please refer the description above). Example:

	#COMMAND: <EVDESC 286823006238237861> #RET: Sportschau #RET: Fußball - Audi Cup: SSC Neapel - FC Bayern München #RET: * Reporter: Marc Schlömer * Übertragung aus München * Moderation: Julia Scharf Produziert in HD #END #OK
<GSQ>	Get signal strength and quality (SSI and SQI) in percent Example: #COMMAND: <GSQ> #RET: 50;100 #OK
<GSCNR>	Get signal carrier-to-noise ratio (CNR) in dB Example: #COMMAND: <GSCNR> #RET: 12.2 dB #OK
<GSSNR>	Get signal signal-to-noise ratio (SNR) in dB #RET: 27.0 dB or if SNR is not available #RET: N/A
<GSRSSI>	Get signal RSSI in dBuV Example: #COMMAND: <GSRSSI> #RET: 39.7 dBuV #OK
<GSQEXT>	Get extended info over signal strength and quality Return format: *SSI;SQI;RSSI;CNR* where *SSI* SSI (strength) in precent *SQI* SQI (quality) in precent *RSSI* signal RSSI in dBuV *CNR* signal CNR in dB Example: #COMMAND: <GSQEXT> #RET: 50;100;39.7 dBuV;12.0 dB #OK
<GSBER>	Get signal bit-error-rate (BER)

	#RET: 1E-8 or if BER is not available #RET: N/A
<GSPER>	Get signal TS Packet Error Ratio (PER) #RET: 1E-8 or if PER is not available #RET: N/A
<GSFEC>	Get code rate (FEC) for DVB-S/S2 signal #RET: 2/3 or if FEC is not available #RET: N/A
<GSPSK>	Get demodulated constellation for DVB-S/S2 signal #RET: 8PSK or if DVB-S/S2 constellation is not available #RET: N/A
<GSSTD>	Get signal modulation (DVB-S/DVB-S2/DVB-C/DVB-C2/DVB-T/DVB-T2/DAB/FM) #RET: DVB-S2
<GSUCBLK>	Get signal uncorrectable TS packets counter (since signal lock) #RET: 0
<GSTSBR>	Get TS bit rate estimated by the demodulator #RET: 3308 kbps or if not available #RET: N/A
<GSTSCLK>	Get current TS output clock frequency from demodulator #RET: 45084 kHz or if not available #RET: N/A
<DAB METRICS>	Get current DAB metrics Return format: *SSI;SQI;RSSI;CNR;SNR;SIGVALID;ACQVALID;AMODE;ABR* where *SSI* SSI (strength) in precent *SQI* SQI (quality) in precent *RSSI* signal RSSI in dBuV *CNR* signal CNR in dB *SNR* signal SNR in dB *SIGVALID* signal validation (1=valid, 0=invalid) *ACQVALID* ensemble acquisition (1=valid, 0=invalid) *AMODE* audio mode ('Dual', 'Mono', 'Stereo, 'Joint Stereo' or '-' if not available) *ABR* audio bitrate in kbps or '-' if not available Example: #COMMAND: <DAB METRICS> #RET: 42;100;27.0 dBuV;16.0 dB;7.0 dB;1;1;Stereo;72 kbps #OK
<DAB DLS>	Get DLS text of the currently playing DAB channel Example: #COMMAND: <DAB DLS> #RET: MISSING YOU - JOHN WAITE

	#OK
<DAB DLSSTATUS>	Get DLS text and status of the currently playing DAB channel Return format: *status;timestamp;dls* where *status* DLS update status (1 = DLS received, 0 = DLS empty) *timestamp* DLS receiption timestamp in format dd.mm.yyyy HH:MM:SS *dls* DLS text Example: #COMMAND: <DAB DLSSTATUS> #RET: 1;06.03.2019 14:20:22;GET LUCKY - DAFT PUNK #OK
<FM METRICS>	Get current FM metrics Return format: *SSI;SQI;RSSI;SNR;MPATH;SIGVALID;RSQVALID;AMODE;ASPLT;ASBL* where *SSI* SSI (strength) in precent *SQI* SQI (quality) in precent *RSSI* signal RSSI in dBuV *SNR* signal SNR in dB *MPATH* signal multipath *SIGVALID* signal validation (1=valid, 0=invalid) *RSQVALID* quality validation (1=valid, 0=invalid) *AMODE* audio mode ('Mono', 'Stereo' or '-' if not available) *ASPLT* stereo pilot indicator (1=present, 0=not present) *ASBL* stereo blend (separation) in percent Example: #COMMAND: <FM METRICS> #RET: 100;100;67.0 dBuV;46.0 dB;2;1;1;Stereo;1;100 #OK
<FM RDS>	Get RDS info of the currently playing FM channel Return format: *pi;pty;ptystr;pst;rt* where *pi* RDS PI (program id) in hex, e.g. D391 *pty* RDS PTY (program type) code in dec, e.g. 10 *ptystr* RDS PTY (program type) string representation, e.g. 'Pop Music' or '-' if not available *pst* RDS PST (program station name) *rt* RDS RT (radio text) Example: #COMMAND: <FM RDS> #RET: D391;10;Pop Music;1LIVE;Bruno Mars - 24K Magic #OK

<FM RDSSTATUS>	Get RDS info and status of the currently playing FM channel Return format: *status;timestamp;pi;pty;ptystr;pst;rt* where *status* RDS update status in dec, bitmask of 0x01 – RDS PI received 0x02 – RDS PTY received 0x04 – RDS RT received 0x08 – RDS PST received *timestamp* RDS receiption/check timestamp in format dd.mm.yyyy HH:MM:SS *pi* RDS PI (program id) in hex, e.g. D391 *pty* RDS PTY (program type) code in dec, e.g. 10 *ptystr* RDS PTY (program type) string representation, e.g. 'Pop Music' or '-' if not available *pst* RDS PST (program station name) *rt* RDS RT (radio text) Example: #COMMAND: <FM RDSSTATUS> #RET: 5;06.03.2019 14:54:07;D79B;0;-;-;Meine Stadt. Mein Radio. #OK
<FM RDS PST>	Get RDS PST (program station) of the currently playing FM channel Example: #COMMAND: <FM RDS PST> #RET: BERG #OK
<FM RDS RT>	Get RDS RT (radio text) of the currently playing FM channel Example: #COMMAND: <FM RDS RT> #RET: 5 Seconds Of Summer - Valentine #OK
<IPTV META TITLE>	Get internet radio stream metadata: Song title Example: #COMMAND: <IPTV META TITLE> #RET: Just Dance #OK
<IPTV META GENRE>	Get internet radio stream metadata: Genre Example: #COMMAND: <IPTV META GENRE> #RET: Pop Music #OK
<REC ?>	Get current recording status

	Returns information for the currently running recording in the following format: #RET: on;*<container>*;*<filename>* where the *<container>* is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive If the recording is currently disabled the command reply will be: #RET: off Examples: #COMMAND: <REC ?> #RET: on;HDD1;ZDF_HD__20171205_170001__heute.ts #OK or #COMMAND: <REC ?> #RET: off #OK
<REC DURATION>	Get current recording duration (total time) in seconds The recording should be previously started (please refer <REC 1> command description). Otherwise the respective error message is sent: #ERROR: Not running Example: #COMMAND: <REC DURATION> #RET: 4802 #OK
<REC TIME>	Get current recording timeline information including start time and duration (total time) as preformatted strings Return format: #RET: *<start_time>;<duration>* where *<start_time>* recording start time as string in format dd/mm/yy HH:MM:SS *<duration>* recording duration as string in format HH:MM:SS The recording should be previously started (please refer <REC 1> command description). Otherwise the respective error message is sent: #ERROR: Not running Example: #COMMAND: <REC TIME> #RET: 07/11/18 10:20:38;01:20:02 #OK
<REC FILES>	Get current recording files list Each recording file in the list is sent in new row in the following format: #RET: *<container>*;*<filename>*;*<start_time>*;*<duration>* where *<container>* is the one of the following:

	HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive *<filename>* recording file name *<start_time>* recording file start time as string in format dd/mm/yy HH:MM:SS *<duration>* recording duration as string in format HH:MM:SS At the end of transmission '#OK' line is sent. The recording could be split into multiple files for the several reasons including EPG program change (split can be disabled: please refer <RECCFG EPGSPLIT n> command description), audio stream change within the current recording (please refer <SAC n> command description) or stream PSI tables updates (e.g. caused by local channel / dynamic program change). The file currently used by recording is always transmitted in the last row. In the case when there was no recording split the only one row with the file currently used by recording is returned. Example: #COMMAND: <REC FILES> #RET: HDD1;Test_R__20181107_162652__Dynamische_Programmumschaltung.ts;07/11/18 16:26:52;00:00:25 #RET: HDD1;Test_R__20181107_162717__Dynamische_Programmumschaltung.ts;07/11/18 16:27:17;00:01:59 #RET: HDD1;Test_R__20181107_162916__Dynamische_Programmumschaltung.ts;07/11/18 16:29:16;00:01:20 #OK
<REC FSIZE>	Get current recording file size Returns information for the file currently used by recording in the following format: #RET: *<file_size>;<container>;<filename>* where *<file_size>* recording file size, e.g. '750 MB' or '2.1 GB' *<container>* is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive *<filename>* recording file name The recording should be previously started (please refer <REC 1> command description). Otherwise the respective error message is sent: #ERROR: Not running Example: #COMMAND: <REC FSIZE> #RET: 1.2 GB;HDD1;ZDF_HD__20171205_170001__heute.ts #OK
<RECCFG EPGSPLIT n>	Enable/Disable split of recording into multiple files by EPG Get current status of the split of recording into multiple files by EPG n = [?] | [ON,OFF] n = ? – get recordings split status (on = enabled, off = disabled) n = ON or OFF – enable or disable split of recording by EPG

	The setting can be changed only when recording is currently disabled. Otherwise the respective error message is sent. Note: Not available for MP form factor devices. Available only for IRD form factor devices equipped with DVB frontends where EPG data is taken from TS and therefore split of recording into multiple files by EPG is possible.
<RECCFG RMDAYS n>	Enable/Set automatic PVR recordings removal (to automatically remove recordings older than the given number of days ago) Disable automatic PVR recordings removal Get current status of the automatic PVR recordings removal n = [?] | [OFF] | [0 .. 1460] n = ? - get current status of the automatic removal n = OFF or 0 - disable automatic removal n = 1 .. 1460 - enable automatic removal and set the number of days (how long ago the recording files should be stored) Note: Command is used to configure the automatic recordings removal. Please refer <REC RMDAYS n> command to remove recordings manually. Examples: #COMMAND: <RECCFG RMDAYS 7> #RET: 7 #OK or #COMMAND: <RECCFG RMDAYS ?> #RET: 7 #OK or #COMMAND: <RECCFG RMDAYS OFF> #RET: off #OK
<RECCFG PREFSTORAGE n>	Set PVR recordings preferred storage Get PVR recordings preferred storage setting n = [?] | [HDD,NAS] n = HDD – PVR default storage is HDD connected to eSATA n = NAS – PVR default storage is NAS network drive The command should be used only when both HDD and NAS network drives are available for PVR recording. Note: Not available for MP form factor devices (where HDD eSATA interface is not available and therefore NAS is the only storage for PVR recordings).

<RECCFG SHOWICON n>	Enable/Disable REC icon (permanent during recording) setting
	Get current status of REC icon (permanent during recording) settings
	
	n = [?] | [ON,OFF]
	n = ? – get REC icon setting status (on = enabled, off = disabled)
	n = ON or OFF – enable or disable REC icon setting
	
	When the setting is enabled the REC icon is shown permanently during active

	recording. When the setting is disabled the REC icon icon is not shown permanently and only appears within the respective OSD GUI elements.
<MPLAY LOOP ?>	Get media player play queue loop setting (decimal code and string value) Examples: #COMMAND: <MPLAY LOOP ?> #RET: 0;off #OK or #COMMAND: <MPLAY LOOP ?> #RET: 1;file #OK or #COMMAND: <MPLAY LOOP ?> #RET: 2;playlist #OK
<PLS VIEW>	View the permanent PLS playlist Each URL in playlist is sent in a new row. At the end of transmission '#OK' line is sent. Example: #COMMAND: <PLS VIEW> #RET: file:///storage/nas1/movies/movie01.mp4 #RET: /storage/hdd1/movies/movie02.mp4 #RET: smb://10.1.1.200/Public/music/song01.mp3 #OK
<PLS LOOPS ?>	Get FILE/PLS playback retries/loops setting (for <PLAY s> command) Return: 0 – infinite playback loop 1..100 – playback retries/loops number Example: #COMMAND: <PLS LOOPS ?> #RET: 0 #OK
<PLS EOF ?>	Get value of 'keep EOF' setting for FILE/PLS playback (via <PLAY s>) Return: on – keep EOF setting enabled (black screen on EOF) off – keep EOF setting disabled (switch to the last stream/program) Please refer <PLS EOF n> command description.
<PLS RESTORE ?>	Get 'restore playback' setting for FILE/PLS playback (via <PLAY s>) Please refer <PLS RESTORE n> command description.

<GETPLAY>	Get current playback state Return: on – playback is disabled (stopped) off – playback is enabled (playing something) Example: #COMMAND: <GETPLAY> #RET: on #OK
<PLAYINFO>	Get current playback information (for IPTV streams in TV/Radio or Media Player mode and for FILE/PLS playback) including stream type/URL/SID Return format: #RET: *<type>;<url>;<sid>* where *<type>* playback type NONE = playback is stopped For IPTV streams or FILE/PLS playback: URL = custom url playback started e.g. with <PLAY s> MPTS = MPTS playback started e.g. with <MPTS n;s> FILE = file playback started e.g. with <PLAY s> PLS = PLS (files) playback started with <PLAY PLS> For the other cases (neither IPTV nor FILE/PLS): DVB-S, DVB-C, DVB-T, DAB, FM or TIMESHIFT *<url>* For IPTV streams, FILE/PLS playback or TIMESHIFT: stream URL or the current file For the other cases (neither IPTV nor FILE/PLS): mux info and current channel name in format e.g. '330 QAM256 6900 | arte' or '514000 kHz | ONE' *<sid>* service id (SID) for MPTS or DVB-S/-C/-T playback or 0 when not available Examples: For IPTV streams #COMMAND: <PLAYINFO> #RET: MPTS;rtp://239.35.10.55:1234;28721 #OK For FILE/PLS playback #COMMAND: <PLAYINFO> #RET: FILE;file:///storage/nas1/movies/movie01.mp4;0 #OK #COMMAND: <PLAYINFO> #RET: PLS;smb://10.1.1.200/Public/music/song04.mp3;0 #OK For TIMESHIFT mode #COMMAND: <PLAYINFO> #RET: TIMESHIFT;/storage/hdd1/recordings/ONE__20200201_103000.ts #OK For the other cases (neither IPTV nor FILE/PLS) #COMMAND: <PLAYINFO> #RET: DVB-S;19.2E 11494 H 22000 | Das Erste HD;10301 #OK
<PLAYSTATE>	Get current playback play state

	Return format: #RET: *n;s* where n - play state decimal code s – play state title string Possible play state codes are: 0 : Playback stopped, s = Stopped 1 : Starting playback, s = Starting 2 : Probing play stream (stays here if stream is not ok), s = Probing 3 : Playing (stream is ok), s = Playing 4 : Playback is paused (only valid for file playback), s = Paused 5 : Stream EOF reached (only valid for file playback), s = Eof Example: #COMMAND: <PLAYSTATE> #RET: 2;Probing #OK
<PLAYRETRIES>	Get current (FILE/PLS media player) playback retries/loops counter Example: #COMMAND: <PLAYRETRIES> #RET: 1 #OK
<ISPAUSE>	Check whether the file playback in the media player or timeshifting is paused 0 : no pause 1 : playback is paused Example: #COMMAND: <ISPAUSE> #RET: 1 #OK
<GETDURATION>	Get current file playback duration (total time) in seconds for media player / timeshifting Example: #COMMAND: <GETDURATION> #RET: 3722 #OK
<GETDURTIME>	Get current file playback duration (total time) as preformatted string for media player / timeshifting Example: #COMMAND: <GETDURTIME> #RET: 01:20:02 #OK
<GETPLAYPOS>	Get current file playback position (elapsed time) in seconds for media player / timeshifting

	Example: #COMMAND: <GETPLAYPOS> #RET: 68 #OK
<GETPLAYTIME>	Get current file playback position (elapsed time) as preformatted string for media player / timeshifting Example: #COMMAND: <GETPLAYTIME> #RET: 00:01:08 #OK
<GETPLAYFILE>	Get current playback file for media player / timeshifting Examples: #COMMAND: <GETPLAYFILE> #RET: PVR;HDD1;arte__20171201_203000__News.ts #OK or #COMMAND: <GETPLAYFILE> #RET: MOVIE;NAS1;MyMovie1.avi #OK
<TSHFT GETDURATION>	Get total playback duration (total time) in seconds for timeshifting The command (unlike <GETDURATION> command which returns duration of the currently played file in the case of the possible recordings split) returns the total timeshifting playback duration (i.e. current recording duration) with the respects to the buffering inside timeshifting.
<TSHFT GETDURTIME>	Get total playback duration (total time) as preformatted string for timeshifting The command (unlike <GETDURTIME> command which returns duration of the currently played file in the case of the possible recordings split) returns the total timeshifting playback duration (i.e. current recording duration) with the respects to the buffering inside timeshifting.
<TSHFT GETPLAYPOS>	Get total playback position (elapsed time) in seconds for timeshifting The command (unlike <GETPLAYPOS> command which returns playback position of the currently played file) returns the total position inside the whole timeshifting with the respects to the possible recordings split. In the case when there was no recordings split the command reply is the same as for <GETPLAYPOS> command.
<TSHFT GETPLAYTIME>	Get total playback position (elapsed time) as preformatted string for timeshifting The command (unlike <GETPLAYTIME> command which returns playback position of the currently played file) returns the total position inside the whole

	timeshifting with the respects to the possible recordings split. In the case when there was no recordings split the command reply is the same as for <GETPLAYTIME> command.
<GRL>	Get available PVR recordings list Each available recording in the list is sent in new row in the following format: #RET: PVR;<container>;<filename> At the end of transmission '#OK' line is sent. The <container> is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive Example: #COMMAND: <GRL> #RET: PVR;HDD1;arte__20171201_203000__News.ts #RET: PVR;NAS1;ZDF__20190117_213005__Sport.ts #OK
<GML>	Get available media player movies list Each available movie in the list is sent in new row in the following format: #RET: MOVIE;<container>;<filename> At the end of transmission '#OK' line is sent. The <container> is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive USB = USB drive (video files in 'movies' folder inside filesystem root) Example: #COMMAND: <GML> #RET: MOVIE;HDD1;MyMovie1.avi #RET: MOVIE;NAS4;MyMovie2.m2ts #OK
<GAL>	Get available audio files list Each available audio file in the list is sent in new row in the following format: #RET: MUSIC;<container>;<filename> At the end of transmission '#OK' line is sent. The <container> is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive USB = USB drive (audio files in 'music' folder inside filesystem root) Example: #COMMAND: <GAL> #RET: MUSIC;HDD1;MySong1.mp3 #RET: MUSIC;NAS3;MySong2.MP3 #OK
<GFL>	Get available media (audio/video) files list

	Each available file in the list is sent in new row in the following format: #RET: FILE;<container>;<filename> At the end of transmission '#OK' line is sent. The <container> is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive USB = USB drive The <filename> is the relative file path inside the container (using '/' char as subfolder delimiter). Example: #COMMAND: <GFL> #RET: FILE;HDD1;movies/MyMovie1.avi #RET: FILE;HDD1;music/MySong1.mp3 #RET: FILE;NAS3;music/MySong2.MP3 #RET: FILE;NAS4;movies/MyMovie2.m2ts #RET: FILE;USB;subfolder/Audio01.mp3 #RET: FILE;USB;Video01.avi #RET: FILE;USB;Video02.mkv #OK
<GFL n>	Get available files list with filter option n = [MOVIE] | [MUSIC] | [ALL] n = MOVIE – filter only video files (by extension) n = MUSIC – filter only audio files (by extension) n = ALL – list all available files (inside HDD/NAS/USB containers) The return row format is the same as for <GFL> command above.
<GRLFP>	Get available PVR recording files (as URLs for <PLAY s> command) Each available file (absolute file path within container) is sent in new row. At the end of transmission '#OK' line is sent. Example: #COMMAND: <GRLFP> #RET: /storage/hdd1/recordings/arte__20171201_203000__News.ts #RET: /storage/nas1/recordings/ZDF__20190117_213005__Sport.ts #OK
<GMLFP>	Get available movie files (as URLs for <PLAY s> command) Each available file (absolute file path within container) is sent in new row. At the end of transmission '#OK' line is sent. Example: #COMMAND: <GMLFP> #RET: /storage/hdd1/movies/MyMovie1.avi #RET: /storage/nas4/movies/MyMovie2.m2ts #OK
<GALFP>	Get available audio files (as URLs for <PLAY s> command)

	Each available file (absolute file path within container) is sent in new row. At the end of transmission '#OK' line is sent. #COMMAND: <GALFP> #RET: /storage/hdd1/music/MySong1.mp3 #RET: /storage/nas3/music/MySong2.MP3 #OK
<GFLFP>	Get available media (audio/video) files (as URLs for <PLAY s> command) Each available file (absolute file path within container) is sent in new row. At the end of transmission '#OK' line is sent. Example: #COMMAND: <GFLFP> #RET: /storage/hdd1/movies/MyMovie1.avi #RET: /storage/hdd1/music/MySong1.mp3 #RET: /storage/nas3/music/MySong2.MP3 #RET: /storage/nas4/movies/MyMovie2.m2ts #RET: /storage/usb/subfolder/Audio01.mp3 #RET: /storage/usb/Video01.avi #RET: /storage/usb/Video02.mkv #OK
<GFLFP n>	Get available files with filter option (as URLs for <PLAY s> command) n = [MOVIE] | [MUSIC] | [ALL] n = MOVIE – filter only video files (by extension) n = MUSIC – filter only audio files (by extension) n = ALL – list all available files (inside HDD/NAS/USB containers) The reply format is the same as for <GFLFP> command above.
<HDD1 INFO> <HDD2 INFO>	Get eSATA HDD1/HDD2 status information Examples: When 500GB HDD1 is connected and partitioned The partition is formatted (NTFS), 430GB from total 500GB is available #COMMAND: <HDD1 INFO> #HDD1: 500 GB #PART1: NTFS;430 GB;500 GB #OK When 500GB HDD2 is connected and partitioned The partition is not formatted #COMMAND: <HDD2 INFO> #HDD2: 500 GB #PART2: Not formatted #OK When 500GB HDD1 is connected but not partitioned #COMMAND: <HDD1 INFO> #HDD1: 500 GB #PART1: Not available #OK

	When HDD1 is not connected #COMMAND: <HDD1 INFO> #HDD1: Not available #PART1: Not available #OK When HDD2 is not connected #COMMAND: <HDD2 INFO> #HDD2: Not available #PART2: Not available #OK
<HDD INFO>	Get eSATA HDDs status information (for both HDD1 and HDD2) Examples: HDD1: 500GB HDD is connected and partitioned HDD1 partition is formatted (NTFS), 430GB from total 500GB available HDD2: 2048GB HDD is connected and partitioned HDD2 partition is formatted (EXT4), read-only, total 2048GB #COMMAND: <HDD INFO> #HDD1: 500 GB #PART1: NTFS;430 GB;500 GB #HDD2: 2048 GB #PART2: EXT4 (read-only);0 GB;2048 GB #OK HDD1: 500GB HDD is connected and partitioned HDD1 partition is not formatted HDD2: 2048GB HDD is connected but not partitioned #COMMAND: <HDD INFO> #HDD1: 500 GB #PART1: Not formatted #HDD1: 2048 GB #PART1: Not available #OK When only HDD1 is connected (partitioned and formatted) #COMMAND: <HDD INFO> #HDD1: 500 GB #PART1: NTFS;430 GB;500 GB #HDD2: Not available #PART2: Not available #OK When both HDD1 and HDD2 are not connected #COMMAND: <HDD INFO> #HDD1: Not available #PART1: Not available #HDD2: Not available #PART2: Not available #OK
<HDD1 FORMAT> <HDD2 FORMAT>	Format eSATA HDD1/HDD2 (perform partitioning and formatting) The operation progress can be observed during the formatting process. Device sends progress information as:

	#RET: *progress_percent* Device sends error message if the formatting process fails for some reason: #ERROR: *error_description* Examples: #COMMAND: <HDD1 FORMAT> #RET: 20% #RET: 40% #RET: 60% #RET: 80% #RET: 100% #OK or #COMMAND: <HDD2 FORMAT> #RET: 20% … #RET: 80% #RET: 100% #OK
<HDD FORMAT>	Format available eSATA HDD (perform partitioning and formatting) Format HDD1 (default) when both HDD1 and HDD2 are available. Format available HDD (from HDD1/HDD2) otherwise. The operation progress can be observed during the formatting process. Device sends progress information as: #RET: *progress_percent* Device sends error message if the formatting process fails for some reason: #ERROR: *error_description* Example: #COMMAND: <HDD FORMAT> #RET: 20% #RET: 40% #RET: 60% #RET: 80% #RET: 100% #OK
<NAS GET CONFIG n>	Get NAS network drive configuration n = 1..4 – NAS network drive number Returns NAS network drive settings in the following format: #RET: *<type>*;*<ip>*;*<share>*;*<enabled>*;*<user>*;*<pwd>*;*<status>* where *<type>* is the one of the following: CIFS = Samba (Windows) network file server NFS = NFS (Linux/Unix) network file server *<ip>* IP-address of the network file server *<share>* network shared resource name *<enabled>* is the one of the following: on = NAS drive is enabled (used by automounter) off = NAS drive is disabled (not used) *<user>* network file server user name (only for CIFS) empty for anonymous/guest login *<pwd>* network file server user password (only for CIFS)

	empty for anonymous/guest login *<status>* is the one of the following: off = network drive is disabled (not used) on = network drive is mounted and writeable readonly = network drive is mounted but without write permissions connect = network drive is currently processed by automounter (applying changes) fail = network drive is currently unavailable Example: #COMMAND: <NAS GET CONFIG 1> #RET: CIFS;10.1.1.200;Public;on;;;on #OK
<NAS SET CONFIG n;s>	Set NAS network drive configuration n = 1..4 – NAS network drive number s = NAS network drive settings as the list of string attributes separated with ";" delimiter in format: *<type>*;*<ip>*;*<share>*;*<enabled>*;*<user>*;*<pwd>* where *<type>* is the one of the following: CIFS = Samba (Windows) network file server NFS = NFS (Linux/Unix) network file server or "-" to disable network drive and remove settings *<ip>* IP-address of the network file server or "-" to disable network drive and remove settings *<share>* network shared resource name or "-" to disable network drive and remove settings *<enabled>* is the one of the following: on = NAS drive is enabled (used by automounter) off = NAS drive is disabled (not used) or "-" to disable network drive and remove settings *<user>* network file server user name (only for CIFS) "-" if empty for anonymous/guest login *<pwd>* network file server user password (only for CIFS) "-" if empty for anonymous/guest login The command reply is the same as for <NAS GET CONFIG n> command. NAS network drive configuration can be changed only when recording is currently disabled or the network drive is not available for recording (no write permissions, unavailable or disabled). Otherwise the respective error message is sent.

<NAS c n>	Enable/Disable NAS network drive
	Get NAS network drive status
	
	c = [?] | [ON,OFF]
	c = ? – get NAS network drive status
	c = ON or OFF – enable or disable NAS network drive
	n = 1..4 – NAS network drive number
	
	NAS network drive can be enabled only when it is properly configured (please refer <NAS SET CONFIG n;s> command description).
	NAS network drive can be disabled only when recording is currently disabled

	or the network drive is not available for recording (no write permissions, unavailable). Otherwise the respective error messages are sent. The return value can be the one of the following: #RET: off = network drive is disabled (not used) #RET: on = network drive is mounted and writeable #RET: readonly = network drive is mounted but without write permissions #RET: connect = network drive is currently processed by automounter (applying changes) #RET: fail = network drive is currently unavailable Example: #COMMAND: <NAS ON 1> #RET: readonly #OK
<NAS DEF STORAGE c>	Enable/Disable NAS as PVR preferred storage Get current status of NAS as PVR preferred storage setting c = [?] | [ON,OFF] c = ? – get the setting status, which is the one of the following: ON – PVR default storage is NAS network drive OFF – PVR default storage is HDD connected to eSATA c = ON/OFF – enable/disable NAS storage as PVR default The command should be used only when both HDD and NAS network drives are available for PVR recording. Alternatively the PVR preferred storage setting can be set with <RECCFG PREFSTORAGE n> command (please refer the command description).
<SNMP COMMUNITY n>	Set SNMP community string Get current SNMP community string n = [?] | [*STRING (SIZE (1..32))*] n = ? – get current SNMP community string n = *STRING (SIZE (1..32))* – set SNMP community string The valid characters for community string are: 'A..Z' | 'a..z' | '0..9' | '_' | '-' (string should not start with '_' and '-'). Examples: #COMMAND: <SNMP COMMUNITY ?> #RET: public #OK or #COMMAND: <SNMP COMMUNITY elium-private> #RET: elium-private #OK

<SNMP n>	Enable/Disable SNMP agent
	Get SNMP agent status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP agent status (on = enabled, off = disabled)
	n = ON or OFF – enable or disable SNMP agent

	Note: SNMP TRAP notifications still can be used when SNMP agent is disabled (please refer <SNMPTRAP n> command description).
<SNMPTRAP n>	Enable/Disable SNMP TRAP notifications Get SNMP TRAP notifications status n = [?] | [ON,OFF] n = ? – get SNMP TRAP notifications status (on/off = enabled/disabled) n = ON or OFF – turn SNMP TRAP notifications enabled or disabled Note: SNMP TRAP host IP address should be set before the SNMP TRAP notifications are turned ON (please refer <SNMPTRAP HOST n>command description).
<SNMPTRAP HOST n>	Set SNMP TRAP host IP address Get SNMP TRAP host IP address n = [?] | [*IP-ADDRESS*] n = ? – get SNMP TRAP host IP address n = *IP-ADDRESS* – set SNMP TRAP host IP address Example: #COMMAND: <SNMPTRAP HOST 10.1.1.220> #RET: 10.1.1.220 #OK
<SNMPTRAP DELAY n>	Set SNMP TRAP notifications delay value Get current SNMP TRAP notifications delay value value n = [?] | [1 .. 60] n = ? – get current SNMP TRAP notifications delay value (in minutes) n = [1 .. 60] – set SNMP TRAP notifications delay value (in minutes) SNMP TRAP notifications delay value represents the delay in minutes between two consecutive alarm notifications of the same type. Example: #COMMAND: <SNMPTRAP DELAY 5> #RET: 5 #OK
<SNMPTRAP alarmNotifications n>	Enable/Disable SNMP TRAP alarm notifications Get SNMP TRAP alarm notifications status n = [?] | [ON,OFF] n = ? – get SNMP TRAP alarm notifications status n = ON/OFF – turn SNMP TRAP alarm notifications enabled/disabled Device alarm notifications are generated continuously with respective delay (please refer <SNMPTRAP DELAY n>command description) between two consecutive notifications until alarm disappears. Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled. To set each notification apart please use respective commands below.

<SNMPTRAP noSignal n>	Enable/Disable SNMP noSignal TRAP alarm notification Get SNMP noSignal TRAP alarm notification status n = [?] | [ON,OFF] n = ? – get SNMP noSignal TRAP status n = ON/OFF – turn SNMP noSignal TRAP enabled/disabled The notification is generated in the case when the tuner unit receives weak signal not suitable for the program playback. It indicates that the tuner cannot lock to the signal and is followed by the retune attempts. The notifications are generated continuously until the signal lock will be reached.
<SNMPTRAP noStream n>	Enable/Disable SNMP noStream TRAP alarm notification Get SNMP noStream TRAP alarm notification status n = [?] | [ON,OFF] n = ? – get SNMP noStream TRAP status n = ON/OFF – turn SNMP noStream TRAP enabled/disabled The notification is generated in the case when there is neither audio nor video streams found in the incoming TS. The notifications are generated continuously until the streams suitable for playback will be found.

<SNMPTRAP scrambledStream n>	Enable/Disable SNMP scrambledStream TRAP alarm notification
	Get SNMP scrambledStream TRAP alarm notification status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP scrambledStream TRAP status
	n = ON/OFF – turn SNMP scrambledStream TRAP enabled/disabled
	
	The notification is generated in the case when the incoming stream is scrambled or corrupted. The notifications are generated continuously until the incoming stream will be good enough for playback.

<SNMPTRAP firmwareEvents n>	Enable/Disable SNMP TRAP firmware events Get SNMP TRAP firmware events status n = [?] | [ON,OFF] n = ? – get SNMP TRAP firmware events status n = ON/OFF – turn SNMP TRAP firmware events enabled/disabled Firmware event notifications are generated once pro each event. Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled. To set each notification apart please use respective commands below.

<SNMPTRAP firmwareReady n>	Enable/Disable SNMP firmwareReady TRAP firmware event
	Get SNMP firmwareReady TRAP firmware event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP firmwareReady TRAP status
	n = ON/OFF – turn SNMP firmwareReady TRAP enabled/disabled

	The notification is generated after the device software startup. It notifies that the device is initialized and ready for use.

<SNMPTRAP firmwareFault n>	Enable/Disable SNMP firmwareFault TRAP firmware event
	Get SNMP firmwareFault TRAP firmware event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP firmwareFault TRAP status
	n = ON/OFF – turn SNMP firmwareFault TRAP enabled/disabled
	
	The notification is generated in the case when the device software reaches hard fault and is normally followed by the software restart.

<SNMPTRAP firmwareUpdate n>	Enable/Disable SNMP firmwareUpdate TRAP firmware event
	Get SNMP firmwareUpdate TRAP firmware event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP firmwareUpdate TRAP status
	n = ON/OFF – turn SNMP firmwareUpdate TRAP enabled/disabled
	
	The notification is sent when the device firmware update starts.

<SNMPTRAP fwUpdateFAIL n>	Enable/Disable SNMP fwUpdateFAIL TRAP firmware event
	Get SNMP fwUpdateFAIL TRAP firmware event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP fwUpdateFAIL TRAP status
	n = ON/OFF – turn SNMP fwUpdateFAIL TRAP enabled/disabled
	
	The notification is sent in the case when the device firmware update process has failed for some reason.

<SNMPTRAP standbyEvents n>	Enable/Disable SNMP TRAP standby events Get SNMP TRAP standby events status n = [?] | [ON,OFF] n = ? – get SNMP TRAP standby events status n = ON/OFF – turn SNMP TRAP standby events enabled/disabled Standby event notifications are generated once pro each event. Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled. To set each notification apart please use respective commands below.

<SNMPTRAP enterStandby n>	Enable/Disable SNMP enterStandby TRAP standby event
	Get SNMP enterStandby TRAP standby event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP enterStandby TRAP status
	n = ON/OFF – turn SNMP enterStandby TRAP enabled/disabled
	
	The notification is sent when the device enters Standby.

<SNMPTRAP leaveStandby n>	Enable/Disable SNMP leaveStandby TRAP standby event
	Get SNMP leaveStandby TRAP standby event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP leaveStandby TRAP status
	n = ON/OFF – turn SNMP leaveStandby TRAP enabled/disabled
	
	The notification is sent when the device leaves Standby.

<SNMPTRAP playbackEvents n>	Enable/Disable SNMP TRAP playback events Get SNMP TRAP playback events status n = [?] | [ON,OFF] n = ? – get SNMP TRAP playback events status n = ON/OFF – turn SNMP TRAP playback events enabled/disabled Playback event notifications are generated once pro each event. Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled. To set each notification apart please use respective commands below.

<SNMPTRAP playbackStart n>	Enable/Disable SNMP playbackStart TRAP playback event
	Get SNMP playbackStart TRAP playback event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP playbackStart TRAP status
	n = ON/OFF – turn SNMP playbackStart TRAP enabled/disabled
	
	The notification is sent when the device starts playback. It also signifies that the current playback status was changed: e.g. on playback mode change or on channel switching.

<SNMPTRAP playbackStop n>	Enable/Disable SNMP playbackStop TRAP playback event
	Get SNMP playbackStop TRAP playback event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP playbackStop TRAP status
	n = ON/OFF – turn SNMP playbackStop TRAP enabled/disabled
	
	The notification is sent when the device stops playback. It can signify that the current device mode is changed: e.g. device goes into setup menu or into Standby.

<SNMPTRAP recordingEvents n>	Enable/Disable SNMP TRAP recording events Get SNMP TRAP recording events status n = [?] | [ON,OFF] n = ? – get SNMP TRAP recording events status n = ON/OFF – turn SNMP TRAP recording events enabled/disabled Recording event notifications are generated once pro each event. Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled.

	To set each notification apart please use respective commands below.

<SNMPTRAP recordingStart n>	Enable/Disable SNMP recordingStart TRAP recording event
	Get SNMP recordingStart TRAP recording event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP recordingStart TRAP status
	n = ON/OFF – turn SNMP recordingStart TRAP enabled/disabled
	
	The notification is sent when the device starts DVB recording.

<SNMPTRAP recordingStop n>	Enable/Disable SNMP recordingStop TRAP recording event
	Get SNMP recordingStop TRAP recording event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP recordingStop TRAP status
	n = ON/OFF – turn SNMP recordingStop TRAP enabled/disabled
	
	The notification is sent when the device stops DVB recording.

<SNMPTRAP streamingEvents n>	Enable/Disable SNMP TRAP streaming events Get SNMP TRAP streaming events status n = [?] | [ON,OFF] n = ? – get SNMP TRAP streaming events status n = ON/OFF – turn SNMP TRAP streaming events enabled/disabled Streaming event notifications are generated once pro each event. Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled. To set each notification apart please use respective commands below.

<SNMPTRAP streamingStart n>	Enable/Disable SNMP streamingStart TRAP streaming event
	Get SNMP streamingStart TRAP streaming event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP streamingStart TRAP status
	n = ON/OFF – turn SNMP streamingStart TRAP enabled/disabled
	
	The notification is sent when the device starts IPTV streaming.

<SNMPTRAP streamingStop n>	Enable/Disable SNMP streamingStop TRAP streaming event
	Get SNMP streamingStop TRAP streaming event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP streamingStop TRAP status
	n = ON/OFF – turn SNMP streamingStop TRAP enabled/disabled
	
	The notification is sent when the device stops IPTV streaming.

<SNMPTRAP appModeEvents n>	Enable/Disable SNMP TRAP App Mode events Get SNMP TRAP App Mode events status

	n = [?] | [ON,OFF]
	n = ? – get SNMP TRAP App Mode events status
	n = ON/OFF – turn SNMP TRAP App Mode events enabled/disabled
	
	App Mode event notifications are generated once pro each event.
	
	Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled.
	To set each notification apart please use respective commands below.

<SNMPTRAP enterAppMode n>	Enable/Disable SNMP enterAppMode TRAP App Mode event
	Get SNMP enterAppMode TRAP App Mode event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP enterStandby TRAP status
	n = ON/OFF – turn SNMP enterStandby TRAP enabled/disabled
	
	The notification is sent when the device enters App Mode.

<SNMPTRAP leaveAppMode n>	Enable/Disable SNMP leaveAppMode TRAP App Mode event
	Get SNMP leaveAppMode TRAP App Mode event status
	
	n = [?] | [ON,OFF]
	n = ? – get SNMP leaveStandby TRAP status
	n = ON/OFF – turn SNMP leaveStandby TRAP enabled/disabled
	
	The notification is sent when the device leaves App Mode.

<CEC CONTROL n>	Enable/Disable HDMI-CEC remote control
	Get HDMI-CEC remote control status
	
	n = [?] | [ON,OFF]
	n = ? – get HDMI-CEC remote control status (on/off = enabled/disabled)
	n = ON/OFF – turn HDMI-CEC remote control enabled/disabled
	
	HDMI-CEC remote control feature enables remote control commands to be passed through HDMI from other CEC-enabled devices within the system (e.g. from TV remote control).

<CEC STANDBY n>	Enable/Disable HDMI-CEC automatic standby Get HDMI-CEC automatic standby status n = [?] | [ON,OFF] n = ? – get HDMI-CEC automatic standby status n = ON/OFF – turn HDMI-CEC automatic standby enabled/disabled HDMI-CEC automatic standby feature enables multiple CEC-enabled devices to switch to or from standby synchronously through HDMI. Turning the device on/off will also turn on/off CEC-enabled TV. The device will be turned on/off when TV will send turn-on/-off signal.
<GSS>	Get RS232 Baud Rate Example: #COMMAND: <GSS>

	#RET: 115200 #OK
<GPI>	Get Program Info This command returns an information regarding the current program: - current mode (TV or Radio) - number of the channel - name of the channel - title of the currently broadcasted event - duration of the currently broadcasted event (in seconds) - time remaining to the end of the event (in seconds) #RET: TV;7;ProSieben;taff.;3660;1262 This data is taken from Transport Stream. Therefore, host should wait some time before <GPI> command is sent (typically several seconds) after switching on a new channel. Otherwise, empty strings may be returned: #RET: ;;;;;
<GTI>	Get current Multiplex (Transponder) Info This command returns information string for the current transponder including standard (DVB-S/-C/-T, IPTV, DAB, FM), satellite, frequency, symbol rate etc. Example: #COMMAND: <GTI> #RET: DVB-S Astra (19.2E) 11494 H 22000 #OK or #COMMAND: <GTI> #RET: IPTV TS udp://239.35.10.55:1234 #OK or #COMMAND: <GTI> #RET: IPTV URL http://send.blackbeats.fm:13000 #OK or #COMMAND: <GTI> #RET: DAB (5C) 178.352 MHz #OK or #COMMAND: <GTI> #RET: FM 102.400 MHz #OK
<GCT>	Get Current Time (Day, Month, Year, Hour, Minute) Example: #COMMAND: <GCT> #RET: 24;10;07;11;26 #OK
<DATE ?>	Get current date and time

	Example: #COMMAND: <DATE ?> #RET: Mon May 10 21:31:26 2020 #OK
<DATE SET s>	Set system date and time manually s = system datetime in format 'YYYY;MM;DD;hh;mm;ss' Example: #COMMAND: <DATE SET 2020;05;10;21;31;30> #RET: Mon May 10 21:31:30 2020 #OK
<RTC ?>	Get RTC unit date and time Note: Available only for MP form factor devices equipped with RTC unit Example: #COMMAND: <DATE ?> #RET: Mon May 10 21:33:21 2020 #OK
<RTC SYNC>	Sync RTC unit date and time with system time Note: Available only for MP form factor devices equipped with RTC unit Example: #COMMAND: <RTC SYNC> #RET: Mon May 10 21:34:20 2020 #OK
<GPS>	Get power supply status #RET: on or #RET: fail Note: Command should be used only for the IRD devices with a second redundant power supply.
<NTP n>	Enable/Disable time sync via NTP Get current status of the time sync via NTP n = [?] | [ON,OFF] n = ? – get NTP time sync status n = ON/OFF – enable/disable time sync via NTP

<NTP SERVER n>	Set NTP server used for time sync
	Get current NTP server used for time sync
	n = [?] | [*NTP_SERVER*]
	n = ? – get current NTP server setting
	n = *NTP_SERVER* – set NTP server (IP-address or hostname)

	Examples: #COMMAND: <NTP SERVER de.pool.ntp.org> #RET: de.pool.ntp.org #OK or #COMMAND: <NTP SERVER 10.1.1.254> #RET: 10.1.1.254 #OK
<HTTP n>	Enable/Disable remote control via HTTP (HTTP Server) Get current status of the remote control via HTTP n = [?] | [ON,OFF] n = ? – get HTTP Server status n = ON/OFF – enable/disable remote control via HTTP
<HTTP PORT n>	Set HTTP Server incoming port Get current HTTP Server incoming port value n = [?] | [1 .. 65535] n = ? – get current HTTP Server incoming port value n = 1 .. 65535 – set HTTP Server incoming port Examples: #COMMAND: <HTTP PORT ?> #RET: 80 #OK or #COMMAND: <HTTP PORT 8080> #RET: 8080 #OK
<HTTP AUTH n>	Set HTTP Server authentication settings (enable/disable, login data) Get current HTTP Server authentication settings n = [?] | [ON,OFF] | [*LOGIN_DATA*] n = ? – get current HTTP Server authentication settings n = ON/OFF – enable/disable HTTP Server authentication n = *LOGIN_DATA* – HTTP Server login data (username and password), the list of string attributes separated with “;” delimiter in format: *username;password* where *username* = sting (size (1..32)) - login username (user ID) *password* = sting (size (1..32)) - login password The valid characters for username are: 'A..Z' | 'a..z' | '0..9' The valid characters for password are: 'A..Z' | 'a..z' | '0..9' | '+' | '?' | '!' | '.' | ',' | ':' | '-' | '=' | '*' | '@' | '_' | '(' | ')' | '[' | ']' | '#' | '$' | '%' | '&' Examples: #COMMAND: <HTTP AUTH ?> #RET: off;elium;elium #OK or #COMMAND: <HTTP AUTH ON>

	#RET: on;elium;elium #OK or #COMMAND: <HTTP AUTH admin;Pass$Phrase#> #RET: on;admin;Pass$Phrase# #OK
<STREAM GET PARAM>	Get current IPTV Streaming settings Returns information string in the following format: #RET: *<status>;<mode>;<url>* where *status* is the one of the following: on – streaming enabled; off – streaming disabled; *mode* is the one of the following: MPTS – multiprogram transport stream; SPTS – single program transport stream; *url* is output stream URL in format: <proto>://<address>:<port>; Example response: #RET: on;MPTS;udp://239.35.10.55:1234 or if the streaming is disabled #RET: off;SPTS;http://10.1.1.55:31339
<STREAM GET STATE>	Get current IPTV Streaming state Example response when the streaming is running: #RET: streaming If the streaming is by any reason not running (e.g. device in menu mode or streaming is disabled): #RET: idle
<STREAM GET CHANNELS>	Get current IPTV Stream programs Each table row is sent immediately after line: '#RET: '. At the end of transmission line '#OK' is sent. Example response for MPTS stream: #RET: TV;1;Das Erste HD;10301;MPTS;udp://239.35.10.55:1234 #RET: TV;2;arte HD;10302;MPTS;udp://239.35.10.55:1234 #RET: TV;3;SWR BW HD;10303;MPTS;udp://239.35.10.55:1234 #OK or for SPTS stream: #RET: TV;8;tagesschau24;28721;URL;udp://239.35.10.55:1234
<CMXC ?>	Get CPU cooling fan maximum temperature Example: #COMMAND: <CMXC ?> #RET: 70 #OK

<CMXL ?>	Get left cooling fan maximum temperature
<CMXR ?>	Get right cooling fan maximum temperature
<CMNL ?>	Get left cooling fan minimum temperature
<CMNR ?>	Get right cooling fan minimum temperature
<CFANL ?>	Get left cooling fan state (enabled/disabled) Example: #COMMAND: <CFANL ?> #RET: on #OK
<CFANR ?>	Get right cooling fan state (enabled/disabled)
<CFANC SPEED>	Get CPU cooling fan speed in percent Example: #COMMAND: <CFANC SPEED> #RET: 60 #OK
<CFANL SPEED>	Get left cooling fan speed in percent
<CFANR SPEED>	Get right cooling fan speed in percent
<GTC> <GTC C>	Get temperature from CPU cooling fan sensor Example: #COMMAND: <GTC C> #RET: 57 #OK
<GTC L>	Get temperature from left cooling fan sensor
<GTC R>	Get temperature from right cooling fan sensor
<APK LIST>	Get available APKs list Each available APK in the list is sent in new row in the following format: #RET: <*package_name*>;<*app_label*> where *package_name* APK package name (unique app identifier) *app_label* End-user-visible app name At the end of transmission '#OK' line is sent. Example: #COMMAND: <APK LIST> #RET: com.netflix.mediaclient;Netflix

	#RET: tunein.player;TuneIn Radio #RET: com.lgi.upcch;UPC TV #OK
<APK INFO n>	Get APK information (check whether the package available) n = APK package name (retrieved e.g. from <APK LIST> command) The return row format is the same as for <APK LIST> command above. The respective error message is sent in the case when there is no APK with the given package name available. Examples: #COMMAND: <APK INFO com.netflix.mediaclient> #RET: com.netflix.mediaclient;Netflix #OK or #COMMAND: <APK INFO com.swisscom.tv2> #ERROR: Not available
<APK ?>	Get currently running APK The return row format is the same as for <APK LIST> command above. Command should be used only in App Mode. Otherwise the respective error message is sent. Examples: #COMMAND: <APK ?> #RET: com.netflix.mediaclient;Netflix #OK or #COMMAND: <APK ?> #ERROR: Not running
<WEBVIEW ?>	Get status of Ads WebView Return format when Ads WebView is enabled: #RET: on;*video_x;video_y;video_w;video_h;video_crop;load_state;page_url* where *video_x* video rect left point in 1920x1080 units *video_y* video rect top point in 1920x1080 units *video_w* video rect width in 1920x1080 units *video_h* video rect height in 1920x1080 units *video_crop* 0 = video scaled in the given rect; 1 = video cropped *load_state* page loading status, one of the following: show - the page loaded successfully (WebView is visible) error - the page is unavailable load - currently loading the page *page_url* web page URL Return format when Ads WebView is disabled: #RET: off Example: #COMMAND: <WEBVIEW ?> #RET: on;0;0;320;240;0;show;http://10.1.1.200/ad1.html

	#OK
<SUBSCR REC TIME n>	Subscribe to / Disable the notification of the recording duration Get subscription status of the notification of the recording duration n = [?] | [0 .. 600] n = ? – get subscription status n = 0 – disable the notification n = 1 .. 600 – enable the notification and notify the recording duration continuously every n seconds If enabled, the notification of the recording duration in format: #SUBSCR REC TIME: HH:MM:SS will be sent continuously every n seconds during recording. Example: #SUBSCR REC TIME: 01:25:02 #SUBSCR REC TIME: 01:25:03 … #SUBSCR REC TIME: 01:25:13
<SUBSCR REC FILE n>	Subscribe to / Disable the notification of the recording file change Get subscription status of the notification of the recording file change n = [?] | [ON,OFF] n = ? – get subscription status n = OFF – disable the notification n = ON – enable the notification and notify each time when the recording file changes If enabled, the notification of the recording file change in format: #SUBSCR REC FILE: <*number*>;<*container*>;<*filename*> where <*number*> recording file numer (total recording files count) <*container*> is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive <*filename*> recording file name will be sent each time when the file currently used by recording changes (due to recording split) during recording. Example: #SUBSCR REC FILE: 1;HDD1;arte__20181107_162652__Sports.ts … #SUBSCR REC FILE: 2;HDD1;arte__20181107_162717__heute.ts
<SUBSCR TSHFT TIME n>	Subscribe to / Disable the notification of the timeshift duration Get subscription status of the notification of the timeshift duration n = [?] | [0 .. 600] n = ? – get subscription status n = 0 – disable the notification n = 1 .. 600 – enable the notification and notify the timeshift duration continuously every n seconds If enabled, the notification of the timeshift duration in format:

	#SUBSCR TSHFT TIME: HH:MM:SS will be sent continuously every n seconds during timeshifting playback.
<SUBSCR TSHFT POS n>	Subscribe to / Disable the notification of the timeshift position Get subscription status of the notification of the timeshift position n = [?] | [0 .. 600] n = ? – get subscription status n = 0 – disable the notification n = 1 .. 600 – enable the notification and notify the timeshift position continuously every n seconds If enabled, the notification of the timeshift position in format: #SUBSCR TSHFT POS: HH:MM:SS will be sent continuously every n seconds during timeshifting playback. Example: #SUBSCR TSHFT POS: 01:24:32 #SUBSCR TSHFT POS: 01:24:33 … #SUBSCR TSHFT POS: 01:24:43
<SUBSCR TSHFT BFR n>	Subscribe to / Disable the notification of the timeshift buffering Get subscription status of the notification of the timeshift buffering n = [?] | [ON,OFF] n = ? – get subscription status n = OFF – disable the notification n = ON – enable the notification and notify each time when the timeshift buffering state changes If enabled, the notification of the timeshift buffering in format: #SUBSCR TSHFT BFR: Buffering ON or #SUBSCR TSHFT BFR: Buffering OFF will be sent each time when the buffering state changes (i.e. once pro each buffering event) during timeshifting playback.
<SUBSCR FPLAY POS n>	Subscribe to / Disable the notification of the file playback timeline Get subscription status of the notification of the file playback timeline n = [?] | [0 .. 600] n = ? – get subscription status n = 0 – disable the notification n = 1 .. 600 – enable the notification and notify the file playback timeline continuously every n seconds If enabled, the notification of the file playback timeline including position inside and duration of the currently played file in format: #SUBSCR FPLAY POS: *<position>;<duration>* where *<position>* playback position in format HH:MM:SS *<duration>* playback duration in format HH:MM:SS will be sent continuously every n seconds during file playback or timeshift. Example:

	#SUBSCR FPLAY POS: 01:24:32;02:25:30 #SUBSCR FPLAY POS: 01:24:33;02:25:30 … #SUBSCR FPLAY POS: 01:24:43;02:25:30
<SUBSCR FPLAY FILE n>	Subscribe to / Disable the notification of the playback file change Get subscription status of the notification of the playback file change n = [?] | [ON,OFF] n = ? – get subscription status n = OFF – disable the notification n = ON – enable the notification and notify each time when the playback file changes If enabled, the notification of the playback file change in format: #SUBSCR FPLAY FILE: *<type>*;*<container>*;*<filename>* where *<type>* multimedia file type, is the one of the following: PVR = PVR recording file MOVIE = media player movie MUSIC = media player audio file FILE = some custom multimedia file *<container>* is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive USB = USB drive *<filename>* playback file name (path) will be sent each time when the file currently used by playback changes during media player playback or timeshifting. Example: #SUBSCR FPLAY FILE: PVR;HDD1;arte__20181107_162652__Sports.ts … #SUBSCR FPLAY FILE: PVR;HDD1;arte__20181107_162717__heute.ts
<LIC KEY ?>	Get License Manager key (device S/N) used to purchase licenses Example: #COMMAND: <LIC KEY ?> #RET: 37190702009155 #OK
<LIC FEATURES>	Get License Manager available features status Example: #COMMAND: <LIC FEATURES> #FEATURE: UHD 4Kx2K (OFF) #FEATURE: UHD 4:2:2 (OFF) #FEATURE: AC3 Downmix (ON) #FEATURE: H.265/HEVC (ON) #FEATURE: Apps (ON) #OK
<LIC LIST>	Get License Manager available licenses (including licensing features)

<table>
<tr><td></td><td>Example:
#COMMAND: <LIC LIST>
#LIC: 4752731679643762
#LIC FEATURE: AC3 Downmix
#LIC FEATURE: H.265/HEVC
#LIC: 4752181890275954
#LIC FEATURE: Apps
#OK</td></tr>
<tr><td><ALARM STATUS></td><td>Get device alarm notifications status

Each alarm notification is sent in new row in the following format:
#RET: <i><alarm_name>;<alarm_status></i>
where
<i>alarm_name</i> Alarm notification title
<i>alarm_status</i> Alarm notification status (on/off)
on = alarm is enabled/active
off = alarm is disabled/inactive
At the end of transmission '#OK' line is sent.

The possible alarm notifications (<i>alarm_name</i> values) are:
NO SIGNAL Enabled when the tuner unit receives weak signal not suitable for the program playback.
NO STREAM Enabled when there is neither audio nor video streams found in the incoming TS.
SCRAMBLED Enabled when the incoming stream is scrambled or corrupted.

Example:
#COMMAND: <ALARM STATUS>
#RET: NO SIGNAL;on
#RET: NO STREAM;off
#RET: SCRAMBLED;off
#OK</td></tr>
<tr><td><DEVINFO NAME n></td><td>Set device name
Get current device name

n = [?] | [<i>STRING (SIZE (1..20))</i>]
n = ? – get current device name
n = <i>STRING (SIZE (1..20))</i> – set device name

The valid characters for device name are:
'A..Z' | 'a..z' | '0..9' | ' ' | '+' | '?' | '!' | '.' | ',' | ':' | '-' | '=' | '*' | '@' | '_' | '(' | ')' | '[' | ']' | '#' | '$' | '%' | '&'

Examples:
#COMMAND: <DEVINFO NAME ?>
#RET: ELIUM IRD
#OK
or
#COMMAND: <DEVINFO NAME My IRD 1>
#RET: My IRD 1
#OK</td></tr>
</table>

<LOGLEVEL ?>	Get current log level (logging verbosity) Example: #COMMAND: <LOGLEVEL ?> #RET: 3 #OK
<LOG DATETIME ? >	Get log messages date/time info setting Example: #COMMAND: <LOG DATETIME ?> #RET: 1;TimeOnly #OK
<LOG SERIAL n>	Enable/Disable RS-232 (serial) logging Get RS-232 (serial) logging status n = ? | ON | OFF n = ? - view serial logging status n = ON | OFF - enable/disable serial logging

<LOG USB n>	Enable/Disable USB Drive logging
	Get USB Drive logging status
	
	n = ? | ON | OFF
	n = ? - view USB Drive logging status
	n = ON | OFF - enable/disable USB Drive logging
	
	Please refer <LOG USB CLR> command description to clear USB drive logs.

<LOG MMC n>	Enable/Disable MMC (Internal Storage) logging Get MMC (Internal Storage) logging status n = ? | ON | OFF n = ? - view MMC (Internal Storage) logging status n = ON | OFF - enable/disable MMC (Internal Storage) logging Please refer <LOG MMC CLR> command description to clear MMC logs. Please refer <LOG MMC CP>, <LOG MMC MV> commands to copy the log files from MMC (Internal Storage) into connected USB drive.
<WEBAPP ?>	Get type of the running GUI Application Possible reply values are: #RET: custom - using Custom WebApp GUI Application #RET: system - using System GUI Application
<WEBAPP CUSTOM ?>	Get status of the Custom WebApp GUI Application Possible reply values are: #RET: unavailable - Custom WebApp GUI is not installed #RET: enabled - Custom WebApp GUI is running #RET: denied - Custom WebApp GUI is blocked (installed but denied)

<WEBAPP CUSTOM DENY n>	Deny/Allow use of Custom WebApp GUI Application
	Get Custom WebApp GUI Application lock status (denied or allowed)
	
	n = [?] | [ON,OFF]
	n = ? - get Custom WebApp GUI lock status:
	on = locked - Custom WebApp GUI is denied, running System GUI;
	off = unlocked - Custom WebApp GUI is allowed (used if installed);
	n = ON or OFF - deny (lock) or allow (unlock) Custom WebApp GUI

<WEBAPP SYNC n s>	Update/Install Custom WebApp GUI Application n = [NET | USB] – update container s = list of parameters depending of n (see below) Possible update containers are: 1. n = NET Update/Install Custom WebApp GUI from network share (NAS, file server). In this case 's' is the list of string attributes separated with “;” delimiter in format: *server_ip;share_name;webapp_path;username;password* where *server_ip* IP-address of the network file server *share_name* network shared resource name *webapp_path* is the one of the following: - relative path inside shared folder to the ZIP archive file containing WebApp GUI Application; - relative path inside shared folder to the folder containing WebApp GUI Application or “-” if WebApp is inside shared folder (in filesystem root); *username* network file server user name or “-” if empty for anonymous (guest) login *password* network file server user password or “-” if empty (no password) 2. n = USB Update/Install Custom WebApp GUI from USB drive. In this case 's' is the one of the following: - relative path inside the USB drive root filesystem to the ZIP archive file containing WebApp GUI Application; - relative path inside the USB drive root filesystem to the folder containing WebApp GUI Application or “-” if WebApp is inside USB filesystem root; Device sends error message if the update/installation fails for some reason: #ERROR: *error_description* <WEBAPP REMOVE> or <WEBAPP CUSTOM DENY n> commands can be used to remove or temporary deny the previously installed Custom WebApp GUI Application (System GUI Application will be used in such case). Note: WebApp GUI Application content should be stored on the root level of the zip archive file (without a parent directory on a top) in the case when ZIP file is used as a WebApp GUI Application container. Examples: To update WebApp GUI Application from network share (ZIP archive) #COMMAND: <WEBAPP SYNC NET 10.1.1.5;HDD;webapp/GUI.zip;-;->

	#OK To update WebApp GUI Application from network share (WebApp in folder) #COMMAND: <WEBAPP SYNC NET 10.1.1.5;HDD;webapp/mygui;-;-> #OK To update WebApp GUI Application from USB drive (ZIP archive) #COMMAND: <WEBAPP SYNC USB webapp/GUI.zip> #OK To update WebApp GUI Application from USB drive (WebApp in USB root) #COMMAND: <WEBAPP SYNC USB -> #OK
<HTDATA ?>	Get status of the Custom HTTP Views Possible reply values are: #RET: unavailable - Custom HTTP Views are not installed #RET: enabled - Custom HTTP Views are used #RET: denied - Custom HTTP Views are blocked (installed but denied)

<HTDATA DENY n>	Deny/Allow use of Custom HTTP Views
	Get Custom HTTP Views lock status (denied or allowed)
	
	n = [?] | [ON,OFF]
	n = ? - get Custom HTTP Views lock status:
	on = locked - Custom HTTP Views are denied;
	off = unlocked - Custom HTTP Views are allowed (used if installed);
	n = ON or OFF - deny (lock) or allow (unlock) Custom HTTP Views

<HTDATA SYNC n s>	Update/Install Custom HTTP Views n = [NET | USB] – update container s = list of parameters depending of n (see below) Possible update containers are: 1. n = NET Update/Install Custom HTTP Views from network share (NAS, file server). In this case 's' is the list of string attributes separated with “;” delimiter in format: *server_ip;share_name;htdata_path;username;password* where *server_ip* IP-address of the network file server *share_name* network shared resource name *htdata_path* is the one of the following: - relative path inside shared folder to the ZIP archive file containing Custom HTTP Views; - relative path inside shared folder to the folder containing Custom HTTP Views or “-” if the HTTP Views are inside shared folder (in filesystem root); *username* network file server user name or “-” if empty for anonymous (guest) login *password* network file server user password or “-” if empty (no password) 2. n = USB Update/Install Custom HTTP Views from USB drive.

	In this case 's' is the one of the following: - relative path inside the USB drive root filesystem to the ZIP archive file containing Custom HTTP Views; - relative path inside the USB drive root filesystem to the folder containing Custom HTTP Views or “-” if the HTTP Views are inside USB filesystem root; Device sends error message if the update/installation fails for some reason: #ERROR: *error_description* <HTDATA REMOVE> or <HTDATA DENY n> commands can be used to remove or temporary deny the previously installed Custom HTTP Views. Note: Custom HTTP Views content should be stored on the root level of the zip archive file (without a parent directory on a top) in the case when ZIP file is used as a Custom HTTP Views container. Examples: To update Custom HTTP Views from network share (ZIP archive) #COMMAND: <HTDATA SYNC NET 10.1.1.5;HDD;htdata/MyViews.zip;-;-> #OK To update Custom HTTP Views from network share (HTTP Views in folder) #COMMAND: <HTDATA SYNC NET 10.1.1.5;HDD;htdata/myviews;-;-> #OK To update Custom HTTP Views from USB drive (ZIP archive) #COMMAND: <HTDATA SYNC USB htdata/MyViews.zip> #OK To update Custom HTTP Views from USB drive (HTTP Views in USB root) #COMMAND: <HTDATA SYNC USB -> #OK
<APPDATA ?>	Get status of the Custom Application Data Possible reply values are: #RET: available - some custom application data present in internal storage #RET: unavailable - there is no custom application data
<APPDATA SYNC n s>	(Re)write Custom Application Data file(s) into internal storage (e.g. playback media files, logos for Custom WebApp GUI and Custom HTTP Views etc) n = [NET | USB] – update container s = list of parameters depending of n (see below) 1. n = NET (Re)write custom application data from network share (NAS, file server). In this case 's' is the list of string attributes separated with “;” delimiter in format: *server_ip;share_name;appdata_path;username;password* where *server_ip* IP-address of the network file server *share_name* network shared resource name *appdata_path* relative path to the custom application data file or folder inside shared folder; *username* network file server user name or “-” if empty for anonymous (guest) login *password* network file server user password or “-” if empty (no password)

	2. n = USB (Re)write custom application data from USB drive. In this case 's' is relative path to custom application data file or folder inside the USB drive root filesystem. Device sends error message if (re)write fails for some reasons: #ERROR: *error_description* <APPDATA REMOVE s> command can be used to remove all or any of the previously written custom application data. Examples: #COMMAND: <APPDATA SYNC NET 10.1.1.200;Public;MP/bootvideo.mp4;-;-> #OK or #COMMAND: <APPDATA SYNC USB bootvideo.mp4> #OK

<OSDCFG PROGINFO n>	Enable/Disable OSD Program Info (on channels switching etc)
	Get current status of OSD Program Info setting
	
	n = [?] | [ON,OFF]
	n = ? – get OSD Program Info setting status (on = enabled, off = disabled)
	n = ON or OFF – enable or disable OSD Program Info setting

<802.1X n>	Enable/Disable 802.1X security Get current status of 802.1X security (enabled or disabled) n = [?] | [ON,OFF] n = ? - view 802.1X security status n = ON | OFF - enable/disable 802.1X security Note: 802.1X security is disabled by default. The 802.1X setup via network should be made before the device is connected to the secure environment. Otherwise the device will be inaccessible in 802.1X network. Alternatively, in the case when the device is already connected to the 802.1X network and cannot be accessible over network the 802.1X settings can be set via serial (RS-232) connection or via device GUI. Example: #COMMAND: <802.1X on> #RET: on #OK
<802.1X AUTH METHOD n>	Set EAP authentication method Get current EAP authentication method value n = ? - view current authentication method n = MD5 | PWD | TTLS | PEAP - set the respective authentication method Examples: #COMMAND: <802.1X AUTH METHOD TTLS> #RET: TTLS #OK or #COMMAND: <802.1X AUTH METHOD PEAP>

<table>
<tr><td></td><td>#RET: PEAP
#OK
or
#COMMAND: <802.1X AUTH METHOD ?>
#RET: PEAP
#OK</td></tr>
<tr><td>&lt;802.1X AUTH USER n&gt;</td><td>Set EAP identity/username string
Get current EAP identity/username string

n = ? - view current EAP identity/username string
n = STRING (SIZE (1..64)) - set EAP identity/username string
The valid characters for username string are:
'A..Z' | 'a..z' | '0..9' | '+' | '?' | '!' | '.' | ',' | ':' | '-' | '=' | '*' | '@' | '_' | '(' | ')' | '[' | ']' | '#' | '$' | '%' | '&amp;'

Examples:
#COMMAND: <802.1X AUTH USER ?>
#RET: elium
#OK
#COMMAND: <802.1X AUTH USER some-911-user>
#RET: some-911-user
#OK</td></tr>
<tr><td>&lt;802.1X AUTH PASSWORD n&gt;</td><td>Set EAP password string
Get current EAP password string

n = ? - view current EAP password string
n = STRING (SIZE (1..64)) - set EAP password string
The valid characters for password string are:
'A..Z' | 'a..z' | '0..9' | '+' | '?' | '!' | '.' | ',' | ':' | '-' | '=' | '*' | '@' | '_' | '(' | ')' | '[' | ']' | '#' | '$' | '%' | '&amp;'

Examples:
#COMMAND: <802.1X AUTH PASSWORD ?>
#RET: elium
#OK
#COMMAND: <802.1X AUTH PASSWORD AbC_dEfgH314+dfge>
#RET: AbC_dEfgH314+dfge
#OK</td></tr>
<tr><td>&lt;802.1X AUTH n&gt;</td><td>Set EAP authentication credentials (username and password)
Get current EAP authentication credentials

n = ? - view current EAP authentication credentials (username and password)
n = LOGIN_DATA - set EAP authentication credentials (username and password), the list of string attributes separated with “;” delimiter in format:
  <i>username;password</i>
where
  <i>username</i> = sting (size (1..64)) - EAP identity/username string
  <i>password</i> = sting (size (1..64)) - EAP password string

The valid characters for username and password strings are:
'A..Z' | 'a..z' | '0..9' | '+' | '?' | '!' | '.' | ',' | ':' | '-' | '=' | '*' | '@' | '_' | '(' | ')' | '[' | ']' | '#' | '$' | '%' | '&amp;'</td></tr>
</table>

	Examples: #COMMAND: <802.1X AUTH ?> #RET: elium;elium #OK #COMMAND: <802.1X AUTH some-911-user;AbC_dEfgH314+dfge> #RET: some-911-user;AbC_dEfgH314+dfge #OK
<802.1X TTLS INNER AUTH n>	Set Phase 2 authentication method for EAP-TTLS Get Phase 2 authentication method for EAP-TTLS n = ? - get TTLS inner authentication n = PAP | CHAP | MSCHAP | MSCHAPv2 | MD5 | GTC - set TTLS inner authentication Note: The setting is only applicable for EAP-TTLS, i.e. when EAP authentication method is set to TTLS respectively. Please refer <802.1X AUTH METHOD n> command above. Examples: #COMMAND: <802.1X TTLS INNER AUTH ?> #RET: MSCHAPv2 #OK #COMMAND: <802.1X TTLS INNER AUTH CHAP> #RET: CHAP #OK
<802.1X PEAP INNER AUTH n>	Set Phase 2 authentication method for EAP-PEAP Get Phase 2 authentication method for EAP-PEAP n = ? - get PEAP inner authentication n = MSCHAPv2 | MD5 | GTC - set PEAP inner authentication Note: The setting is only applicable for EAP-PEAP, i.e. when EAP authentication method is set to PEAP respectively. Please refer <802.1X AUTH METHOD n> command above. Examples: #COMMAND: <802.1X PEAP INNER AUTH ?> #RET: MSCHAPv2 #OK #COMMAND: <802.1X PEAP INNER AUTH MD5> #RET: MD5 #OK
<802.1X INFO>	Get actual 802.1X configuration Examples: #COMMAND: <802.1X INFO> #802.1X: on #AUTH METHOD: PEAP #AUTH USER: some-911-user #AUTH PASSWORD: AbC_dEfgH314+dfge #PEAP INNER AUTH: MSCHAPv2 #OK or

	#COMMAND: <802.1X INFO> #802.1X: off #AUTH METHOD: MD5 #AUTH USER: elium #AUTH PASSWORD: elium #OK
<HLS BR n>	Set IPTV HLS input streams preferable/maximum bitrate Get current IPTV HLS input streams preferable/maximum bitrate n = [?] | [MIN,MAX] | [br_kbs] n = ? - get urrent IPTV HLS input streams preferable/maximum bitrate n = MIN - always prefer HLS stream with minimum bitrate n = MAX - always prefer HLS stream with maximum bitrate n = br_kbs - set the maximum bitrate value in kbs for HLS stream to select from multivariant playlist Examples: #COMMAND: <HLS BR ?> #RET: MAX #OK #COMMAND: <HLS BR 8000> #RET: 8000 #OK
<SCHED REB n>	Enable/Disable scheduled reboot with the respective parameters Get the status (enabled/disabled) of the scheduled reboot n = [?] | [OFF] | [*PARAM_STRING*] n = ? - get the status (enabled/disabled) of the scheduled reboot n = OFF - disable scheduled reboot n = *PARAM_STRING* - enable scheduled reboot with the respective parameters, defined with *PARAM_STRING*. *PARAM_STRING* is the list of string attributes separated with “;” delimiter in format: *weekdays;hour;minute* *where* *weekdays* = comma separated weekdays (days of each week) used for scheduled reboot in following form or "*" char for every day; Mon or Mo = Monday Tue or Tu = Tuesday Wed or We = Wednesday Thu or Th = Thursday Fri or Fr = Friday Sat or Sa = Saturay Sun or Su = Sunday or * = reboot every day *hour* = scheduled reboot hour, 00..23 *minute* = scheduled reboot minute, 00..59 Examples: Get the status of the scheduled reboot - scheduled reboot is disabled #COMMAND: <SCHED REB ?> #RET: off #OK Get the status - scheduled reboot is set to 23:55 each Tuesday and Friday

	#COMMAND: <SCHED REB ?> #RET: on;Tue,Fri;23;55 #OK Disable scheduled reboot #COMMAND: <SCHED REB OFF> #RET: off #OK Setup scheduled reboot to 06:55 each Friday #COMMAND: <SCHED REB Fri;06;55> #RET:on;Fri;05;55 #OK Setup scheduled reboot to 20:05 each Tuesday, Thursday and Saturday #COMMAND: <SCHED REB Tue,Thu,Sat;20;05> #RET:on;Tue,Thu,Sat;20;05 #OK Setup scheduled reboot every day at 07:00 #COMMAND: <SCHED REB *;07;00> #RET:on;*;07;00 #OK