

## **ELIUM IRD-UHD Device RS232/Network Remote Control Description**

### **1. General**

Date: **06.03.2019**

Revision: **0.11**

Scope: The goal of this document is to describe how ELIUM IRD-UHD 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 IRD-UHD 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 IRD-UHD 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 IRD-UHD 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 IRD-UHD 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.
<SIP s>	Change device network configuration  s = list of string attributes separated with ";" delimiter in format: <i>ipaddr;netmask;gateway;dns</i> where <i>ipaddr</i> = device IP-address or "- " if no change; <i>netmask</i> = device network mask or "- " if no change; <i>gateway</i> = network gateway IP-address or "- " if no change;

	<p><i>dns</i> = DNS server IP-address or "- " if no change;</p> <p>Example: #COMMAND: &lt;SIP 10.1.1.54;-;10.1.1.1;10.1.1.1&gt; #OK</p>
<SDM n>	<p>Set video display mode</p> <p>n = decimal code for video display mode Possible decimal codes for video display mode are:</p> <ul style="list-style-type: none"> <li>0 PAL (analog video output via CVBS)</li> <li>1 NTSC (analog video output via CVBS)</li> <li>2 480i 60Hz</li> <li>3 576i 50Hz</li> <li>4 480p 60Hz</li> <li>5 576p 50Hz</li> <li>6 720p 50Hz</li> <li>7 720p 60Hz</li> <li>8 1080i 50Hz</li> <li>9 1080i 60Hz</li> <li>10 1080p 24Hz</li> <li>11 1080p 50Hz</li> <li>12 1080p 60Hz</li> <li>13 2160p 50Hz 420</li> <li>14 2160p 60Hz 420</li> <li>15 2160p 24Hz 422</li> <li>16 2160p 25Hz 422</li> <li>17 2160p 30Hz 422</li> </ul> <p>Example: #COMMAND: &lt;SDM 7&gt; #OK</p>
<SAO n>	<p>Set audio output</p> <p>n = decimal code for audio output Possible decimal codes for audio output are:</p> <ul style="list-style-type: none"> <li>0 HDMI</li> <li>1 L/R</li> </ul> <p>Note: Available only for 'EL-19843 Rev.00.00' series Backend Boards where audio output is shared between HDMI and L/R. Otherwise the following error message is sent: #ERROR: Command not allowed</p> <p>Example: #COMMAND: &lt;SAO 1&gt; #OK</p>

<ON>	<p>Turn on device</p> <p>Note: Command should be used only in Standby mode. Otherwise the following error message is sent: #ERROR: Not in standby</p>
<OFF>	<p>Turn off device</p> <p>Note: Command should not be used in Standby mode. Otherwise the following error message is sent: #ERROR: Already in standby</p>
<RMCC n>	<p>Simulates an input via remote control</p> <p>n = remote control key code for the given key Possible key codes (for device remote control) are:</p> <ul style="list-style-type: none"> <li>22 Key ON/OFF</li> <li>2 Key '1'</li> <li>5 Key '2'</li> <li>6 Key '3'</li> <li>9 Key '4'</li> <li>10 Key '5'</li> <li>13 Key '6'</li> <li>14 Key '7'</li> <li>17 Key '8'</li> <li>18 Key '9'</li> <li>21 Key '0'</li> <li>34 Key MODE</li> <li>37 Key RADIO/TV</li> <li>38 Key MUTE</li> <li>41 Key LAST</li> <li>25 Key UP</li> <li>26 Key DOWN</li> <li>29 Key LEFT</li> <li>33 Key RIGHT</li> <li>30 Key OK</li> <li>42 Key MENU</li> <li>45 Key EXIT</li> <li>58 Key Red (PVR / DVR)</li> <li>61 Key Green (MOVIE / DVD)</li> <li>62 Key Yellow (MUSIC / MP3/JPEG)</li> <li>65 Key Blue (MEDIA / GAME)</li> <li>46 Key &lt;&lt; (rew. back)</li> <li>49 Key PLAY/PAUSE</li> <li>50 Key &gt;&gt; (rew. forward)</li> <li>53 Key  &lt;&lt; (go prev.)</li> <li>54 Key REC/STOP</li> <li>57 Key &gt;&gt;  (go next)</li> <li>66 Key INFO</li> <li>69 Key EPG</li> </ul>

	<p>70 Key TIMER  73 Key TXT  74 Key PIP  77 Key SEARCH / FREEZE  78 Key TECH.INFO / ZOOM  81 Key AUDIO VIDEO</p>
<RMC c>	<p>Simulates an input via remote control</p> <p>Possible 'c' chars are:</p> <p>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 &lt;&lt; (rew. back)  E Key PLAY/PAUSE  F Key &gt;&gt; (rew. forward)  G Key  &lt;&lt; (go prev.)  H Key &gt;&gt;  (go next)  I Key TIMER</p>

<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
<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
<SAC n>	<p>Set audio stream for the current channel</p> <p>n = audio stream ID (received with &lt;GAC&gt; command) Available audio streams can retrieved by &lt;GAC&gt; command.</p> <p>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.</p> <p>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.</p> <p>Example: #COMMAND: &lt;SAC 5406&gt; #OK</p>
<REC 1> <REC ON>	<p>Start recording of the current program</p> <p>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 &lt;HDD INFO&gt;, &lt;NAS GET CONFIG n&gt; commands description). Otherwise the respective error message is sent.</p>
<REC 0> <REC OFF>	<p>Stop recording of the current program</p> <p>The recording should be previously started (please refer &lt;REC 1&gt; command description). Otherwise the respective error message is sent. In the case of Timeshifting mode the command will also stop timeshifting.</p>
<REC D>	<p>Stop recording of the current program and remove recording files</p> <p>The recording should be previously started (please refer &lt;REC 1&gt; 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.</p>
<TSHFT 1> <TSHFT ON>	Start timeshifting with the current recording

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

	<p>s = media file path in format of &lt;GRL&gt;/&lt;GML&gt;/&lt;GAL&gt; command reply (please refer &lt;GRL&gt;,&lt;GML&gt;,&lt;GAL&gt; commands description) or custom media file path in the following format:              &lt;container&gt;;&lt;filename&gt;</p> <p>where          The &lt;container&gt; is the one of the following:              HDD[1..2] = HDD connected to eSATA              NAS[1..4] = NAS network drive          The &lt;filename&gt; is the full file path (including subfolders) to the media file (using '/' char as subfolder delimiter).</p> <p>Examples:          #COMMAND: &lt;MPLAYLIST +=PVR;HDD1;M1__20171203_162430.ts&gt;          #OK          or          #COMMAND: &lt;MPLAYLIST +=MOVIE;NAS1;movie1.avi&gt;          #OK          or          #COMMAND: &lt;MPLAYLIST +=HDD1;myaudio/mysong1.mp3&gt;          #OK          or          #COMMAND: &lt;MPLAY =HDD1;nonexistent_file.mp4&gt;          #ERROR: File not available</p>
<p>&lt;MPLAYLIST clear&gt;</p>	<p>Clear the media player play queue</p> <p>Example:          #COMMAND: &lt;MPLAYLIST CLEAR&gt;          #OK</p>
<p>&lt;MPLAY list&gt;</p>	<p>Start the media player play queue</p> <p>The media player play queue should be previously filled up with the media files (please refer &lt;MPLAYLIST +=s&gt; command description).</p> <p>Examples:          #COMMAND: &lt;MPLAY LIST&gt;          #OK          or          #COMMAND: &lt;MPLAY LIST&gt;          #ERROR: Empty playlist</p>
<p>&lt;MPLAY LOOP n&gt;</p>	<p>Set media player playback loop setting</p> <p>n = decimal code for the media player playback loop setting          Possible decimal codes for the media player playback loop setting:</p> <ul style="list-style-type: none"> <li>0 playback loop is disabled</li> <li>1 playback loop is enabled for the current file</li> <li>2 playback loop is enabled for the play queue</li> </ul>

<MPSTOP>	Stop media player playback (returns back to Recordings Browser GUI) Stop timeshifting with the current recording
<MPPAUSE>	Pause playback in the media player Pause timeshifting
<MPPLAY>	Resume playback in the media player Resume timeshifting
<MPFF>	Jump 5% (minimum 30 seconds) forward in the current media player playback Jump 5% (minimum 30 seconds) forward in timeshifting
<MPFF n>	Jump n minutes forward in the current media player playback Jump n minutes forward in timeshifting
<MPRW>	Jump 5% (minimum 30 seconds) backward in the current media player playback Jump 5% (minimum 30 seconds) backward in timeshifting
<MPRW n>	Jump n minutes backward in the current media player playback Jump n minutes backward in timeshifting
<MPSTA>	Jump to the start of the current media player 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 playback Jump to the middle of the current timeshifting file
<MPEND>	Jump to end of the current media player 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
<REC RMFILE s>	Remove 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

	<p>following error message is sent: #ERROR: File is used by recording</p> <p>The recording storage should be available (please refer &lt;HDD INFO&gt;, &lt;NAS GET CONFIG n&gt; commands description). Otherwise the respective error message is sent.</p> <p>Examples: #COMMAND: &lt;REC RMFILE PVR;HDD1;R1__20181107_123001__Re.ts&gt; #OK or #COMMAND: &lt;REC RMFILE HDD1;ZDF__20181107_170001__heute.ts&gt; #OK</p>										
<REC RMDATE s>	<p>Remove PVR recording files by date (created earlier than the given date)</p> <p>s = date string as a row of the attributes separated with ";" char in the following format:                   yyyy;mm;dd;HH;MM</p> <p>where</p> <table> <tr> <td>yyyy</td> <td>full year, e.g. 2018</td> </tr> <tr> <td>mm</td> <td>month as a decimal number (01-12)</td> </tr> <tr> <td>dd</td> <td>day of the month (01-31)</td> </tr> <tr> <td>HH</td> <td>hour in 24h format (00-23)</td> </tr> <tr> <td>MM</td> <td>minute (00-59)</td> </tr> </table> <p>The recording storage should be available (please refer &lt;HDD INFO&gt;, &lt;NAS GET CONFIG n&gt; commands description). Otherwise the respective error message is sent.</p> <p>Example: #COMMAND: &lt;REC RMDATE 2018;11;07;12;30&gt; #OK</p>	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)
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)										
<REC RMDAYS n>	<p>Remove PVR recording files older than the given number of days ago</p> <p>n = number of days (how long ago the recording files should be stored)</p> <p>Note: Command is used to remove recordings manually. Please refer &lt;RECCFG RMDAYS n&gt; command to configure the automatic recordings removal.</p> <p>Example: #COMMAND: &lt;REC RMDAYS 7&gt; #OK</p>										
<SETPLAYPOS n>	<p>Set position / Seek inside the current file playback (in seconds) for media player / timeshifting</p> <p>n = [+/-] seconds</p>										

	<p>n without a leading sign – set the absolute position from start. n with a leading sign – relative seek from the current position.</p> <p>Example: #COMMAND: &lt;SETPLAYPOS -120&gt; #OK</p>
<SETPLAYTIME s>	<p>Set position inside the current file playback (as preformatted string) for media player / timeshifting</p> <p>s = absolute position as string in format HH:MM:SS</p> <p>Example: #COMMAND: &lt; SETPLAYTIME 00:02:30&gt; #OK</p>
<TSHFT SETPLAYPOS n>	<p>Set position / Seek inside playback (in seconds) for timeshifting</p> <p>n = [+/-] seconds n without a leading sign – set the absolute position from start. n with a leading sign – relative seek from the current position.</p> <p>The command (unlike &lt;SETPLAYPOS n&gt; 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 &lt;SETPLAYPOS n&gt; command.</p>
<TSHFT SETPLAYTIME s>	<p>Set position inside playback (as preformatted string) for timeshifting</p> <p>s = absolute position as string in format HH:MM:SS</p> <p>The command (unlike &lt; SETPLAYTIME s&gt; 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 &lt; SETPLAYTIME s&gt; command.</p>
<TSHFT GOPREV>	<p>Jump to the previous timeshifting file if available Jump to the start of the timeshifting otherwise</p>
<TSHFT GONEXT>	<p>Jump to the next timeshifting file if available Jump to the end of the timeshifting otherwise</p>
<CLT RM n>	<p>Remove TV channel(s) from TV channels list</p>

	<p>n = [ALL]   [SCRAMBLED]   [1 .. <i>channellist-size</i>]  n = ALL - remove all TV channels (empty TV channels list)  n = SCRAMBLED - remove all scrambled TV channels  n = 1 .. <i>channellist-size</i> - channel number in TV list to remove</p> <p>Examples:  #COMMAND: &lt;CLT RM ALL&gt;  #OK  or  #COMMAND: &lt;CLT RM SCRAMBLED&gt;  #OK  or  #COMMAND: &lt;CLT RM 19&gt;  #OK</p>
<CLR RM n>	<p>Remove Radio channel(s) from Radio channels list</p> <p>n = [ALL]   [SCRAMBLED]   [1 .. <i>channellist-size</i>]  n = ALL - remove all Radio channels (empty Radio channels list)  n = SCRAMBLED - remove all scrambled Radio channels  n = 1 .. <i>channellist-size</i> - channel number in Radio list to remove</p> <p>Example:  #COMMAND: &lt;CLR RM 81&gt;  #OK</p>
<SCAN STOP>	<p>Stop currently running channels search</p> <p>Command should be used only within the search of type 'Auto' or 'Network' (please refer &lt;SCAN STATUS&gt; command description). The search should be currently running. Otherwise the respective error message is sent.</p> <p>Example:  #COMMAND: &lt;SCAN STOP&gt;  #OK</p>
<SCAN EXIT>	<p>GUI Channel Search menu Off request  Stop currently running channels search</p> <p>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.</p> <p>Example:  #COMMAND: &lt;SCAN EXIT&gt;  #OK</p>

<p>&lt;SCAN DVB-S AUTO s&gt;</p>	<p>Request DVB-S automatic channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>sat_pos;scan_type;fta_only</i></p> <p>where</p> <ul style="list-style-type: none"> <li><i>sat_pos</i>            satellite position, e.g. '19.2E' or '26.0W'</li> <li><i>scan_type</i>        channels scan type <ul style="list-style-type: none"> <li>0 = TV &amp; Radio</li> <li>1 = TV Only</li> <li>2 = Radio Only</li> <li>3 = All Services</li> </ul> </li> <li><i>fta_only</i>           scan only free programs (FTA Only) <ul style="list-style-type: none"> <li>on = FTA Only (skip scrambled programs)</li> <li>off = scan also for scrambled programs</li> </ul> </li> </ul> <p>Example: #COMMAND: &lt;SCAN DVB-S AUTO 19.2E;0;off&gt; #OK</p>
<p>&lt;SCAN DVB-S MANUAL s&gt;</p>	<p>Request DVB-S manual channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>sat_pos;freq;pol;sr;scan_type;fta_only</i></p> <p>where</p> <ul style="list-style-type: none"> <li><i>sat_pos</i>            satellite position, e.g. '19.2E' or '26.0W'</li> <li><i>freq</i>                transponder frequency in MHz, e.g. 10744</li> <li><i>pol</i>                 transponder polarization ('H' or 'V')</li> <li><i>sr</i>                 transponder symbol rate, e.g. 22000</li> <li><i>scan_type</i>        channels scan type <ul style="list-style-type: none"> <li>0 = TV &amp; Radio</li> <li>1 = TV Only</li> <li>2 = Radio Only</li> <li>3 = All Services</li> </ul> </li> <li><i>fta_only</i>           scan only free programs (FTA Only) <ul style="list-style-type: none"> <li>on = FTA Only (skip scrambled programs)</li> <li>off = scan also for scrambled programs</li> </ul> </li> </ul> <p>Example: #COMMAND: &lt;SCAN DVB-S MANUAL 19.2E;10744;H;22000;1;on&gt; #OK</p>
<p>&lt;SCAN DVB-C AUTO s&gt;</p>	<p>Request DVB-C automatic channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>mod;sr;scan_type;fta_only</i></p> <p>where</p> <ul style="list-style-type: none"> <li><i>mod</i>                modulation ('auto', 'QAM64' or 'QAM256')</li> <li><i>sr</i>                 symbol rate ('auto', '6111', '6875', '6900' or '6950')</li> <li><i>scan_type</i>        channels scan type <ul style="list-style-type: none"> <li>0 = TV &amp; Radio</li> <li>1 = TV Only</li> </ul> </li> </ul>

	<p>2 = Radio Only 3 = All Services</p> <p><i>fta_only</i> scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs</p> <p>Example: #COMMAND: &lt;SCAN DVB-C AUTO Auto;Auto;0;on&gt; #OK</p>
<p>&lt;SCAN DVB-C MANUAL s&gt;</p>	<p>Request DVB-C manual channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>freq;mod;sr;scan_type;fta_only</i></p> <p>where</p> <p><i>freq</i> transponder frequency in MHz, e.g. 322 <i>mod</i> transponder modulation ('auto', 'QAM16', 'QAM32', 'QAM64', 'QAM128' or 'QAM256') <i>sr</i> transponder symbol rate, e.g. 6900 or 'auto' <i>scan_type</i> channels scan type 0 = TV &amp; Radio 1 = TV Only 2 = Radio Only 3 = All Services</p> <p><i>fta_only</i> scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs</p> <p>Example: #COMMAND: &lt;SCAN DVB-C MANUAL 322;QAM256;6900;2;off&gt; #OK</p>
<p>&lt;SCAN DVB-C NETWORK s&gt;</p>	<p>Request DVB-C network (LCN) channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>nid;freq;mod;sr;scan_type;fta_only</i></p> <p>where</p> <p><i>nid</i> network ID, e.g. 1234 <i>freq</i> transponder frequency in MHz, e.g. 322 <i>mod</i> transponder modulation ('QAM16', 'QAM32', 'QAM64', 'QAM128' or 'QAM256') <i>sr</i> transponder symbol rate, e.g. 6900 <i>scan_type</i> channels scan type 0 = TV &amp; Radio 1 = TV Only 2 = Radio Only 3 = All Services</p> <p><i>fta_only</i> scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs</p>

	<p>Example: #COMMAND: &lt;SCAN DVB-C NETWORK 1234;322;QAM259;6900;0;off&gt; #OK</p>
<SCAN DVB-T AUTO s>	<p>Request DVB-T automatic channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>sat_pos;scan_type;fta_only</i></p> <p>where</p> <p><i>scan_type</i> channels scan type 0 = TV &amp; Radio 1 = TV Only 2 = Radio Only 3 = All Services</p> <p><i>fta_only</i> scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs</p> <p>Example: #COMMAND: &lt;SCAN DVB-T 3;on&gt; #OK</p>
<SCAN DVB-T MANUAL s>	<p>Request DVB-T manual channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>freq;mod;sr;scan_type;fta_only</i></p> <p>where</p> <p><i>freq</i> transponder frequency in kHz, e.g. 360000 <i>scan_type</i> channels scan type 0 = TV &amp; Radio 1 = TV Only 2 = Radio Only 3 = All Services</p> <p><i>fta_only</i> scan only free programs (FTA Only) on = FTA Only (skip scrambled programs) off = scan also for scrambled programs</p> <p>Example: #COMMAND: &lt;SCAN DVB-T MANUAL 360000;0;off&gt; #OK</p>
<SCAN IPTV s>	<p>Request IPTV (manual) channels search</p> <p>s = row of the attributes separated with ";" char in the following format: <i>iptv_proto;iptv_address;iptv_port;scan_type;fta_only</i></p> <p>where</p> <p><i>iptv_proto</i> IPTV stream proto ('UDP') <i>iptv_address</i> IPTV stream multicast/broadcast address, e.g. 239.35.10.231 <i>iptv_port</i> IPTV stream port, e.g. 1234</p>

	<p><i>scan_type</i> channels scan type  0 = TV &amp; Radio  1 = TV Only  2 = Radio Only  3 = All Services</p> <p><i>fta_only</i> scan only free programs (FTA Only)  on = FTA Only (skip scrambled programs)  off = scan also for scrambled programs</p> <p>Example:  #COMMAND: &lt;SCAN IPTV UDP;239.35.10.231;1234;0;off&gt;  #OK</p>
<SCAN DAB>	<p>Request DAB band scan (automatic channels search)</p> <p>Example:  #COMMAND: &lt;SCAN DAB&gt;  #OK</p>
<SCAN DAB OVERWRITE>	<p>Request DAB band scan (automatic channels search)  Overwrite (remove) existing DAB programs on search</p>
<SCAN FM>	<p>Request FM band scan (automatic channels search)</p> <p>Example:  #COMMAND: &lt;SCAN FM&gt;  #OK</p>
<SCAN FM OVERWRITE>	<p>Request FM band scan (automatic channels search)  Overwrite (remove) existing FM programs on search</p>

## 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>	<p>Get device firmware and hardware information</p> <p>Example:            #COMMAND: &lt;VER&gt;            #Mainboard: Rev.01.00            #Firmware: Ver.00.07            #S/N: 01928374657410            #OK</p>
<FWINFO>	<p>Get extended device firmware information</p> <p>Example:            #COMMAND: &lt;FWINFO&gt;            #Firmware: Ver.00.07 build 02 (01.08.2017 11:42)            #OK</p>
<UPDATE n s>	<p>Update device firmware</p> <p>n = [NET   USB] - update container (firmware image container)            s = list of parameters depending of n (see below)</p> <p>Possible update containers are:</p> <p>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:                <i>image_file;server_ip;share_name;username;password</i>            where                <i>image_file</i> = relative path to firmware image file inside shared folder;                <i>server_ip</i> = IP-address of the network file server;</p>

	<p><i>share_name</i> = network shared resource name;  <i>username</i> = network file server user name or          "-" if empty for anonymous (guest) login;  <i>password</i> = network file server user password or          "-" if empty (no password);</p> <p>2. n = USB          Update from USD drive.          In this case 's' is relative path to firmware image file inside the USB drive root filesystem.</p> <p>The operation progress can also be observed during the update process. Device sends progress information as:          #RET: <i>progress_percent</i>          Device sends error message if the update process fails for some reason:          #ERROR: <i>error_description</i></p> <p>Examples:          To update from network share          #COMMAND:          &lt;UPDATE NET elium_IRD-UHD_v00.07.img;10.1.1.5;Public;-;-&gt;          #RET: 20%          #RET: 40%          #RET: 60%          #RET: 80%          #RET: 100%          #OK          To update from USB drive          #COMMAND: &lt;UPDATE USB elium_IRD-UHD_v00.07.img&gt;          #RET: 20%          ...          #RET: 80%          #RET: 100%          #OK</p>
<UPDATE ELIUM>	<p>Update device firmware with the latest release via Web</p> <p>The operation progress can be observed during the update process. Device sends progress information as:          #RET: <i>progress_percent</i>          Device sends error message if the update process fails for some reason:          #ERROR: <i>error_description</i></p> <p>Example:          #COMMAND: &lt;UPDATE ELIUM&gt;          #RET: 20%          ...          #RET: 80%          #RET: 100%          #OK</p>

<p>&lt;IPC&gt;</p>	<p>Get device network configuration</p> <p>Example:  #COMMAND: &lt;IPC&gt;  #MACADDR: 00:E0:4C:02:0B:31  #IP: 10.1.1.55  #MASK: 255.255.255.0  #GW: 10.1.1.3  #DNS: 10.1.1.3  #OK</p>
<p>&lt;DISPLAY MODES&gt;</p>	<p>Get all available video display modes</p> <p>Example:  #COMMAND: &lt;DISPLAY MODES&gt;  #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</p>
<p>&lt;DISPLAY MODE n&gt;</p>	<p>Set video display mode (as string value)  Get video display mode (as string value)</p> <p>n = [?]   [<i>mode_string</i>]  n = ? - get current video display mode as string  n = <i>mode_string</i> - set video mode with the string value</p> <p>Possible string values for video display mode are:</p> <ul style="list-style-type: none"> <li>PAL</li> <li>NTSC</li> <li>480i 60Hz</li> <li>576i 50Hz</li> <li>480p 60Hz</li> <li>576p 50Hz</li> <li>720p 50Hz</li> <li>720p 60Hz</li> <li>1080i 50Hz</li> </ul>

	<p>1080i 60Hz          1080p 24Hz          1080p 50Hz          1080p 60Hz          2160p 50Hz 420          2160p 60Hz 420          2160p 24Hz 422          2160p 25Hz 422          2160p 30Hz 422</p> <p>Examples:          #COMMAND: &lt;DISPLAY MODE ?&gt;          #RET: 1080p 60Hz          #OK          or          #COMMAND: &lt;DISPLAY MODE 720p 50Hz&gt;          #RET: 720p 50Hz          #OK</p>
<GDM>	<p>Get video display mode (decimal code and string value)</p> <p>Example:          #COMMAND: &lt;GDM&gt;          #RET: 12;1080p 60Hz          #OK</p>
<AUDIO OUTS>	<p>Get all available (selectable) audio outputs</p> <p>Note: Available only for 'EL-19843 Rev.00.00' series Backend Boards where audio output is shared between HDMI and L/R. Otherwise the following error message is sent:          #ERROR: Command not allowed</p> <p>Example:          #COMMAND: &lt;AUDIO OUTS&gt;          #RET: 0;HDMI          #RET: 1;L/R          #OK</p>
<AUDIO OUT n>	<p>Set audio output (as string value)          Get audio output (as string value)</p> <p>n = [?]   [<i>output_string</i>]          n = ? - get current audio output as string          n = <i>output_string</i> - set audio output with the string value</p> <p>Possible string values for audio output are:          HDMI          L/R</p>

	<p>Note: Available only for 'EL-19843 Rev.00.00' series Backend Boards where audio output is shared between HDMI and L/R. Otherwise the following error message is sent: #ERROR: Command not allowed</p> <p>Examples: #COMMAND: &lt;AUDIO OUT ?&gt; #RET: HDMI #OK or #COMMAND: &lt;AUDIO OUT L/R&gt; #RET: L/R #OK</p>
<GAO>	<p>Get audio output (decimal code and string value)</p> <p>Note: Available only for 'EL-19843 Rev.00.00' series Backend Boards where audio output is shared between HDMI and L/R. Otherwise the following error message is sent: #ERROR: Command not allowed</p> <p>Example: #COMMAND: &lt;GAO&gt; #RET: 0;HDMI #OK</p>
<GCV>	<p>Get current volume (mute state and volume level)</p> <p>Example: #COMMAND: &lt;GCV&gt; #RET: on;100 #OK</p>
<VOL n>	<p>Set/change volume (mute state or volume level) Get current volume (mute state and volume level)</p> <p>n = [?]   [+/-][0 .. 100]   [ON,OFF] n = ? - get current volume setting (same as &lt;GCV&gt; 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</p> <p>Examples: #COMMAND: &lt;VOL ?&gt; #RET: on;100 #OK or #COMMAND: &lt;VOL -10&gt; #RET: on;90 #OK</p>

	<p>or  #COMMAND: &lt;VOL 50&gt;  #RET: on;50  #OK</p> <p>or  #COMMAND: &lt;VOL OFF&gt;  #RET: off;50  #OK</p>
<LCK n>	<p>Lock/Unlock Frontpanel Keys  Get Lock status for Frontpanel Keys</p> <p>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</p> <p>Example:  #COMMAND: &lt;LCK ON&gt;  #RET: on  #OK</p>
<LCI n>	<p>Lock/Unlock IR Remote  Get Lock status for IR Remote</p> <p>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</p> <p>Example:  #COMMAND: &lt;LCI ?&gt;  #RET: off  #OK</p>
<VRMC n>	<p>Enable/Disable Virtual IR Remote Control (VRMC) mode  Get VRMC mode status (enabled/disabled)</p> <p>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</p> <p>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.</p> <p>Please refer &lt;RMC c&gt; command for the possible VRMC chars.</p>

<p>&lt;GCS&gt;</p>	<p>Get current device state (On or Standby)</p> <p>Possible return:</p> <table border="0"> <tr> <td>#RET: on</td> <td>device is On (active)</td> </tr> <tr> <td>#RET: off</td> <td>device is in Standby</td> </tr> </table>	#RET: on	device is On (active)	#RET: off	device is in Standby																
#RET: on	device is On (active)																				
#RET: off	device is in Standby																				
<p>&lt;GCM&gt;</p>	<p>Get current device mode</p> <p>Possible return:</p> <table border="0"> <tr> <td>#RET: Restart</td> <td>Device is booting up</td> </tr> <tr> <td>#RET: Standby</td> <td>Device is in standby</td> </tr> <tr> <td>#RET: Idle</td> <td>Device is in menu (idle)</td> </tr> <tr> <td>#RET: TV</td> <td>TV program playback</td> </tr> <tr> <td>#RET: Radio</td> <td>Radio program playback</td> </tr> <tr> <td>#RET: Time Shift</td> <td>Timeshifting mode with the current recording</td> </tr> <tr> <td>#RET: Media Player</td> <td>Media Player mode with the file playback</td> </tr> <tr> <td>#RET: Search</td> <td>Searching for the channels</td> </tr> <tr> <td>#RET: Firmware Update</td> <td>Updating firmware</td> </tr> <tr> <td>#RET: HDD Format</td> <td>Formatting HDD</td> </tr> </table>	#RET: Restart	Device is booting up	#RET: Standby	Device is in standby	#RET: Idle	Device is in menu (idle)	#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 playback	#RET: Search	Searching for the channels	#RET: Firmware Update	Updating firmware	#RET: HDD Format	Formatting HDD
#RET: Restart	Device is booting up																				
#RET: Standby	Device is in standby																				
#RET: Idle	Device is in menu (idle)																				
#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 playback																				
#RET: Search	Searching for the channels																				
#RET: Firmware Update	Updating firmware																				
#RET: HDD Format	Formatting HDD																				
<p>&lt;TTT&gt;</p>	<p>Turn to TV mode</p> <p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) and when TV channels are available. Otherwise the respective error message is sent.</p> <p>Example:          #COMMAND: &lt;TTT&gt;          #RET: TV          #OK</p>																				
<p>&lt;TTR&gt;</p>	<p>Turn to Radio mode</p> <p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio) and when Radio channels are available. Otherwise the respective error message is sent.</p> <p>Example:          #COMMAND: &lt;TTR&gt;          #RET: Radio          #OK</p>																				
<p>&lt;PRT U&gt;</p>	<p>Switch TV channel Up (current channel -1)</p> <p>Command should be used only in TV mode. Otherwise the respective error message is sent.</p> <p>Example:</p>																				

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

	<p>message is sent.</p> <p>Example:  #COMMAND: &lt;PRT *Sport News&gt;  #RET: TV;51;Sky Sport News  #OK</p>
<PRR =pn>	<p>Switch to Radio channel whose name is equal with pn</p> <p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). Otherwise the respective error message is sent.</p> <p>Example:  #COMMAND: &lt;PRR =NDR Kultur&gt;  #RET: Radio;26;NDR Kultur  #OK</p>
<PRR *pn>	<p>Switch to Radio channel whose name contains pn</p> <p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). Otherwise the respective error message is sent.</p> <p>Example:  #COMMAND: &lt;PRR *Kultur&gt;  #RET: Radio;26;NDR Kultur  #OK</p>
<PRT n>	<p>Switch to TV channel by number (n is a number)</p> <p>n = channel number in TV list (received e.g. from &lt;GCL&gt; command)</p> <p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). Otherwise the respective error message is sent.</p> <p>Example:  #COMMAND: &lt;PRT 11&gt;  #RET: TV;11;Das Erste HD  #OK</p>
<PRR n>	<p>Switch to Radio channel by number (n is a number)</p> <p>n = channel number in Radio list (received e.g. from &lt;GCL&gt; command)</p> <p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). Otherwise the respective error message is sent.</p>

	<p>Example:  #COMMAND: &lt;PRR 32&gt;  #RET: Radio;32;NDR Plus  #OK</p>
<GNT>	<p>Get number of TV channels</p> <p>Example:  #COMMAND: &lt;GNT&gt;  #RET: 1170  #OK</p>
<GNR>	<p>Get number of Radio channels</p> <p>Example:  #COMMAND: &lt;GNR&gt;  #RET: 162  #OK</p>
<GCC>	<p>Get current channel (channel number)</p> <p>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.</p> <p>Example:  #COMMAND: &lt;GCC&gt;  #RET: 19  #OK</p>
<GCP>	<p>Get current program (name of program)</p> <p>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.</p> <p>Example:  #COMMAND: &lt;GCP&gt;  #RET: SES UHD Demo Channel  #OK</p>
<GCL>	<p>Get channel list (list of TV and Radio programs including numbering)</p> <p>Each table row is sent immediately after line: '#RET: '.</p> <p>At the end of transmission lines '#END' and '#OK' are sent.</p> <p>Example:</p>

	<pre>#COMMAND: &lt;GCL&gt; #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</pre>						
<p>&lt;GCLPT n m&gt;</p>	<p>Get the part of the current TV channels list (including numbering)</p> <p>n = the program number to start from m = the total (maximum) number of programs to get from start</p> <p>Example: #COMMAND: &lt;GCLPT 10 2&gt; #RET: TV;10;ZDFinfo #RET: TV;11;Das Erste HD #END #OK</p>						
<p>&lt;GCLPR n m&gt;</p>	<p>Get the part of the current Radio channels list (including numbering)</p> <p>n = the program number to start from m = the total (maximum) number of programs to get from start</p> <p>Example: #COMMAND: &lt;GCLPR 161 1000&gt; #RET: Radio;161;F. INTER #RET: Radio;162;CULTURE #END #OK</p>						
<p>&lt;GCPMUX&gt;</p>	<p>Get current channel (including type, number, name, mux and TS info)</p> <p>Return format: <i>type;number;muxinfo;SID;ONID;TSID;name</i></p> <p>where</p> <table border="0"> <tr> <td><i>type</i></td> <td>channel type (either TV or Radio)</td> </tr> <tr> <td><i>number</i></td> <td>channel number</td> </tr> <tr> <td><i>muxinfo</i></td> <td>mux info string depending from delivery system (please refer examples below)</td> </tr> </table>	<i>type</i>	channel type (either TV or Radio)	<i>number</i>	channel number	<i>muxinfo</i>	mux info string depending from delivery system (please refer examples below)
<i>type</i>	channel type (either TV or Radio)						
<i>number</i>	channel number						
<i>muxinfo</i>	mux info string depending from delivery system (please refer examples below)						

	<p> <i>SID</i> service id / DAB service id / FM RDS PI (dec)  <i>ONID</i> original network id / DAB component id  <i>TSID</i> transport stream id / DAB ensemble id  <i>name</i> name of program         </p> <p>Examples:</p> <pre> #COMMAND: &lt;GCPMUX&gt; #RET: TV;2;DVB-S 19.2E 10744 H 22000;28722;1;1051;ONE #OK or #COMMAND: &lt;GCPMUX&gt; #RET: TV;800;DVB-C 322 QAM256 6900;28721;1;1051;tagesschau24 #OK or #COMMAND: &lt;GCPMUX&gt; #RET: TV;900;DVB-T 514000 kHz;769;8468;27136;Das Erste HD #OK or #COMMAND: &lt;GCPMUX&gt; #RET: TV;950;IPTV TS udp://239.35.10.55:1234;762;0;0;Das Erste #OK or #COMMAND: &lt;GCPMUX&gt; #RET: Radio;5;IPTV URL http://panel.nadaje.com:11038;0;0;0;BIZZ.fm #OK or #COMMAND: &lt;GCPMUX&gt; #RET: Radio;180;DAB 178352 kHz;28368;6540;3525;Absolut relax #OK or #COMMAND: &lt;GCPMUX&gt; #RET: Radio;190;FM 102400 kHz;54161;0;0;1LIVE #OK         </pre>
<p>&lt;GCLMUX&gt;</p>	<p>Get channel list (including type, numbering, name, mux and TS info)</p> <p>Each table row is sent immediately after line: '#RET: '.</p> <p>At the end of transmission lines '#END' and '#OK' are sent.</p> <p>The return row format is the same as for &lt;GCPMUX&gt; command above.</p> <p>Example:</p> <pre> #COMMAND: &lt;GCLMUX&gt; #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 ...         </pre>

	<pre>#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</pre>				
<GCLMUXPT n m>	<p>Get the part of the current TV channels list (including type, numbering, name, mux and TS info)</p> <p>n = the program number to start from m = the total (maximum) number of programs to get from start</p> <p>The return row format is the same as for &lt;GCPMUX&gt; command above.</p> <p>Example: #COMMAND: &lt;GCLMUXPT 10 2&gt; #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</p>				
<GCLMUXPR n m>	<p>Get the part of the current Radio channels list (including type, numbering, name, mux and TS info)</p> <p>n = the program number to start from m = the total (maximum) number of programs to get from start</p> <p>The return row format is the same as for &lt;GCPMUX&gt; command above.</p> <p>Example: #COMMAND: &lt;GCLMUXPR 161 1000&gt; #RET: Radio;161;DVB-S 19.2E 12363 V 27500;9157;1;1098;F. INTER #RET: Radio;162;IPTV URL http://panel.nadaje.com:11038;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</p>				
<PRMUX s>	<p>Switch to TV/Radio channel by mux info (including SID)</p> <p>s = row of the attributes (retrieved e.g. from &lt;GCLMUX&gt; command) separated with ";" char in the following format: <i>muxinfo;SID</i></p> <p>where</p> <table border="0"> <tr> <td><i>muxinfo</i></td> <td>mux info string depending from delivery system (please refer &lt;GCPMUX&gt;, &lt;GCLMUX&gt; commans)</td> </tr> <tr> <td><i>SID</i></td> <td>service id / DAB service id / FM don't care (0)</td> </tr> </table>	<i>muxinfo</i>	mux info string depending from delivery system (please refer <GCPMUX>, <GCLMUX> commans)	<i>SID</i>	service id / DAB service id / FM don't care (0)
<i>muxinfo</i>	mux info string depending from delivery system (please refer <GCPMUX>, <GCLMUX> commans)				
<i>SID</i>	service id / DAB service id / FM don't care (0)				

	<p>Command should be used only in TV/Radio program playback (when the current mode is either TV or Radio). Otherwise the respective error message is sent.</p> <p>Examples:</p> <pre>#COMMAND: &lt;PRMUX DVB-S 19.2E 11954 H 27500;28011&gt; #RET: TV;10;ZDFinfo #OK or #COMMAND: &lt;PRMUX DVB-S 19.2E 12363 V 27500;9158&gt; #RET: Radio;162;CULTURE #OK or #COMMAND: &lt;PRMUX IPTV TS udp://239.35.10.55:1234;762&gt; #RET: TV;950;Das Erste #OK or #COMMAND: &lt;PRMUX DAB 178352 kHz;28367&gt; #RET: Radio;163;1LIVE #OK or #COMMAND: &lt;PRMUX FM 102400 kHz;0&gt; #RET: Radio;164;1LIVE #OK</pre>				
<p>&lt;PRTMUX s&gt;</p>	<p>Switch to TV channel by mux info (including SID)</p> <p>s = row of the attributes (retrieved e.g. from &lt;GCLMUX&gt; command) separated with ";" char in the following format:</p> <p style="text-align: center;"><i>muxinfo;SID</i></p> <p>where</p> <table style="margin-left: 40px;"> <tr> <td><i>muxinfo</i></td> <td>mux info string depending from delivery system</td> </tr> <tr> <td><i>SID</i></td> <td>service id</td> </tr> </table> <p>Example:</p> <pre>#COMMAND: &lt;PRTMUX DVB-S 19.2E 11954 H 27500;28011&gt; #RET: TV;10;ZDFinfo #OK</pre>	<i>muxinfo</i>	mux info string depending from delivery system	<i>SID</i>	service id
<i>muxinfo</i>	mux info string depending from delivery system				
<i>SID</i>	service id				
<p>&lt;PRRMUX s&gt;</p>	<p>Switch to Radio channel by mux info (including SID)</p> <p>s = row of the attributes (retrieved e.g. from &lt;GCLMUX&gt; command) separated with ";" char in the following format:</p> <p style="text-align: center;"><i>muxinfo;SID</i></p> <p>where</p> <table style="margin-left: 40px;"> <tr> <td><i>muxinfo</i></td> <td>mux info string depending from delivery system</td> </tr> <tr> <td><i>SID</i></td> <td>service id / DAB service id / FM don't care (0)</td> </tr> </table> <p>Example:</p> <pre>#COMMAND: &lt;PRRMUX DVB-S 19.2E 12363 V 27500;9158&gt; #RET: Radio;162;CULTURE</pre>	<i>muxinfo</i>	mux info string depending from delivery system	<i>SID</i>	service id / DAB service id / FM don't care (0)
<i>muxinfo</i>	mux info string depending from delivery system				
<i>SID</i>	service id / DAB service id / FM don't care (0)				

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

	<p>Example:  #COMMAND: &lt;PRRTS DVB-S;28011;1;1079&gt;  #RET: TV;10;ZDFinfo  #OK</p>								
<PRRTS s>	<p>Switch to Radio channel by TS info (including delivery system)</p> <p>s = row of the attributes (retrieved e.g. from &lt;GCLMUX&gt; command) separated with ";" char in the following format:  <i>delsys;SID;ONID;TSID</i></p> <p>where</p> <table> <tr> <td><i>delsys</i></td> <td>delivery system ('DVB-S', 'DVB-C', 'DVB-T', 'DAB' or 'FM')</td> </tr> <tr> <td><i>SID</i></td> <td>service id / DAB service id / FM RDS PI (dec)</td> </tr> <tr> <td><i>ONID</i></td> <td>original network id / DAB component id FM don't care (0)</td> </tr> <tr> <td><i>TSID</i></td> <td>transport stream id / DAB ensemble id FM don't care (0)</td> </tr> </table> <p>Example:  #COMMAND: &lt;PRRTS DVB-S;9158;1;1098&gt;  #RET: Radio;162;CULTURE  #OK</p> <p>or</p> <p>#COMMAND: &lt;PRRTS DAB;28367;6540;3525&gt;  #RET: Radio;163;1LIVE  #OK</p> <p>or</p> <p>#COMMAND: &lt;PRRTS FM;54161;0;0&gt;  #RET: Radio;164;1LIVE  #OK</p>	<i>delsys</i>	delivery system ('DVB-S', 'DVB-C', 'DVB-T', 'DAB' or 'FM')	<i>SID</i>	service id / DAB service id / FM RDS PI (dec)	<i>ONID</i>	original network id / DAB component id FM don't care (0)	<i>TSID</i>	transport stream id / DAB ensemble id FM don't care (0)
<i>delsys</i>	delivery system ('DVB-S', 'DVB-C', 'DVB-T', 'DAB' or 'FM')								
<i>SID</i>	service id / DAB service id / FM RDS PI (dec)								
<i>ONID</i>	original network id / DAB component id FM don't care (0)								
<i>TSID</i>	transport stream id / DAB ensemble id FM don't care (0)								
<CLT ADD URL s>	<p>Add custom IPTV stream URL to TV channels list</p> <p>s = row of the attributes separated with ";" char in the following format:  <i>name;stream_url</i></p> <p>where</p> <table> <tr> <td><i>name</i></td> <td>TV channel name STRING (SIZE (1..32))</td> </tr> <tr> <td><i>stream_url</i></td> <td>IPTV stream URL STRING (SIZE (1..127))</td> </tr> </table> <p>Example:  #COMMAND: &lt;CLT ADD URL TLC;http://s-hls.iptv.net/hls/tlc.m3u8&gt;  #RET: TV;15;TLC  #OK</p>	<i>name</i>	TV channel name STRING (SIZE (1..32))	<i>stream_url</i>	IPTV stream URL STRING (SIZE (1..127))				
<i>name</i>	TV channel name STRING (SIZE (1..32))								
<i>stream_url</i>	IPTV stream URL STRING (SIZE (1..127))								
<CLR ADD URL s>	<p>Add custom Internet Radio stream URL to Radio channels list</p>								

	<p>s = row of the attributes separated with ";" char in the following format:  <i>name;stream_url</i></p> <p>where</p> <table border="0"> <tr> <td><i>name</i></td> <td>Radio channel name STRING (SIZE (1..32))</td> </tr> <tr> <td><i>stream_url</i></td> <td>Internet Radio stream URL STRING (SIZE (1..127))</td> </tr> </table> <p>Example:  #COMMAND: &lt;CLR ADD URL Beats FM;http://send.blackbeats.fm:13000&gt;  #RET: Radio;34;Beats FM  #OK</p>	<i>name</i>	Radio channel name STRING (SIZE (1..32))	<i>stream_url</i>	Internet Radio stream URL STRING (SIZE (1..127))
<i>name</i>	Radio channel name STRING (SIZE (1..32))				
<i>stream_url</i>	Internet Radio stream URL STRING (SIZE (1..127))				
<CLT SETNAME s>	<p>Set/Edit TV channel name</p> <p>s = row of the attributes separated with ";" char in the following format:  <i>number;name</i></p> <p>where</p> <table border="0"> <tr> <td><i>number</i></td> <td>TV channel number 1 .. <i>channellist-size</i></td> </tr> <tr> <td><i>name</i></td> <td>TV channel name STRING (SIZE (1..32))</td> </tr> </table> <p>Example:  #COMMAND: &lt;CLT SETNAME 15;TLC HD&gt;  #RET: TV;15;TLC HD  #OK</p>	<i>number</i>	TV channel number 1 .. <i>channellist-size</i>	<i>name</i>	TV channel name STRING (SIZE (1..32))
<i>number</i>	TV channel number 1 .. <i>channellist-size</i>				
<i>name</i>	TV channel name STRING (SIZE (1..32))				
<CLR SETNAME s>	<p>Set/Edit Radio channel name</p> <p>s = row of the attributes separated with ";" char in the following format:  <i>number;name</i></p> <p>where</p> <table border="0"> <tr> <td><i>number</i></td> <td>Radio channel number 1 .. <i>channellist-size</i></td> </tr> <tr> <td><i>name</i></td> <td>Radio channel name STRING (SIZE (1..32))</td> </tr> </table> <p>Example:  #COMMAND: &lt;CLR SETNAME 34;Black Beats FM&gt;  #RET: Radio;34;Black Beats FM  #OK</p>	<i>number</i>	Radio channel number 1 .. <i>channellist-size</i>	<i>name</i>	Radio channel name STRING (SIZE (1..32))
<i>number</i>	Radio channel number 1 .. <i>channellist-size</i>				
<i>name</i>	Radio channel name STRING (SIZE (1..32))				
<CLT MOVE s>	<p>Set/Edit TV channel number (move TV channel in the list)</p> <p>s = row of the attributes separated with ";" char in the following format:  <i>number;set_number</i></p> <p>where</p> <table border="0"> <tr> <td><i>number</i></td> <td>TV channel number</td> </tr> <tr> <td><i>set_number</i></td> <td>assigned TV channel number</td> </tr> </table>	<i>number</i>	TV channel number	<i>set_number</i>	assigned TV channel number
<i>number</i>	TV channel number				
<i>set_number</i>	assigned TV channel number				

	<p>Example:  #COMMAND: &lt;CLT MOVE 15;1&gt;  #RET: TV;1;TLC HD  #OK</p>
<CLR MOVE s>	<p>Set/Edit Radio channel number (move Radio channel in the list)</p> <p>s = row of the attributes separated with ";" char in the following format:  <i>number;set_number</i></p> <p>where  <i>number</i>            Radio channel number  <i>set_number</i>      assigned Radio channel number</p> <p>Example:  #COMMAND: &lt;CLR MOVE 34;30&gt;  #RET: Radio;30;Black Beats FM  #OK</p>
<SCAN STATUS>	<p>Get current/last channels search status</p> <p>Return format:  <i>state;delsys;scantype;progress;TV total/ins;Radio total/ins</i></p> <p>where  <i>state</i>                Scan state                           on = search is enabled (currently running)                           off = search is disabled (complete/not running)</p> <p><i>delsys</i>               Scanned delivery system  DVB-S, DVB-C, DVB-T, IPTV, DAB or FM</p> <p><i>scantype</i>            Scan type  Auto = Automatic Search  Manual = Manual Search  Network = DVB-C Network/LCN Search</p> <p><i>progress</i>            Scan progress in percent</p> <p><i>TV total/ins</i>        Found TV channels  <i>total</i> = total found channels number  <i>ins</i> = number of the inserted (new) channels</p> <p><i>Radio total/ins</i>    Found Radio channels  <i>total</i> = total found channels number  <i>ins</i> = number of the inserted (new) channels</p> <p>Command shows the status of the currently running channels search or the last complete channels search.</p> <p>Examples:  When the search is currently running  #COMMAND: &lt;SCAN STATUS&gt;  #RET: on;DVB-S;Auto;40%;TV 400/42;Radio 120/20  #OK  or  #COMMAND: &lt;SCAN STATUS&gt;</p>

	<pre>#RET: on;DAB;Auto;94%;TV 0/0;Radio 26/13 #OK When the search is complete #COMMAND: &lt;SCAN STATUS&gt; #RET: off;DVB-C;Manual;100%;TV 6/0;Radio 4/4 #OK When there was no search before #COMMAND: &lt;SCAN STATUS&gt; #RET: off;-;-;0%;TV 0/0;Radio 0/0 #OK</pre>
<SCAN PRT n>	<p>Get found TV channels within the current/last search</p> <p>n = the channel number to start from</p> <p>Each table row is sent immediately after line: '#RET: '.</p> <p>At the end of transmission lines '#END' and '#OK' are sent.</p> <p>Example:</p> <pre>#COMMAND: &lt;SCAN PRT 1&gt; #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</pre>
<SCAN PRR n>	<p>Get found Radio channels within the current/last search</p> <p>n = the channel number to start from</p> <p>Each table row is sent immediately after line: '#RET: '.</p> <p>At the end of transmission lines '#END' and '#OK' are sent.</p> <p>Example:</p> <pre>#COMMAND: &lt;SCAN PRR 14&gt; #RET: 14;Absolute Relax #RET: 15;SCHLAGERPARADIES #END #OK</pre>
<GAC>	<p>Get available audio streams for the current channel</p> <p>Each table row is sent immediately after line: '#RET: '.</p> <p>At the end of transmission lines '#END' and '#OK' are sent.</p> <p>Each return row is send in the following format:</p> <p style="text-align: center;"><i>apid (aformat) - adesc</i></p>

	<p>where</p> <p><i>apid</i> audio stream ID</p> <p><i>aformat</i> audio format (MPGA, AC3, AAC, AAC+, DTS etc)</p> <p><i>adesc</i> audio language or description (optional)</p> <p>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.</p> <p>Example:  #COMMAND: &lt;GAC&gt;  #RET: 5102 (MPGA) - German  #RET: 5103 (MPGA) - Miscellaneous  #RET: 5106 (AC3) - German  #END  #OK</p>
<GCA>	<p>Get current audio stream for the current channel</p> <p>The return row format is the same as for &lt;GAC&gt; command above.</p> <p>Example:  #COMMAND: &lt;GCA&gt;  #RET: 5102 (MPGA) - German  #OK</p>
<EVT n>	<p>Get available EPG events for TV channel by number (n is a number)</p> <p>The information for each event is sent immediately after line: '#RET: '. At the end of transmission lines '#END' and '#OK' are sent.</p> <p>The event information for every event consists of:</p> <ul style="list-style-type: none"> <li>- channel type (TV)</li> <li>- channel number</li> <li>- name of program</li> <li>- Event ID</li> <li>- date and time of the event in format dd/mm/yy HH:MM</li> <li>- duration of the event (in minutes)</li> <li>- title of the event</li> <li>- FSK decimal code</li> </ul> <p>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 &lt;EVT&gt; command for the selected channel (typically few minutes for the next several days). Otherwise, not all EPG events may be stored in the cache.</p> <p>Example:  #COMMAND: &lt;EVT 3&gt;  #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</p>

	<pre>... #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</pre>
<EVR n>	<p>Get available EPG events for Radio channel by number (n is a number)</p> <p>The answer format is the same like the above &lt;EVT&gt; command except that the event information for every event contains</p> <ul style="list-style-type: none"> <li>- channel type (Radio) instead of (TV).</li> </ul> <p>All restrictions are also the same as for &lt;EVT&gt; command.</p>
<EVDESC n>	<p>Get detailed description for the selected EPG event.</p> <p>n = Event ID of the selected EPG event</p> <p>Available events (Event IDs) can be retrieved by &lt;EVT&gt; or &lt;EVR&gt; command (please refer the description above).</p> <p>Example:</p> <pre>#COMMAND: &lt;EVDESC 286823006238237861&gt; #RET: Sportschau #RET: Fußball - Audi Cup: SSC Neapel - FC Bayern München #RET: * Reporter: Marc Schlömer * Übertragung aus München * Moderation: Julia Scharf</pre> <p>Produziert in HD</p> <pre>#END #OK</pre>
<GSQ>	<p>Get signal strength and quality (SSI and SQI) in percent</p> <p>Example:</p> <pre>#COMMAND: &lt;GSQ&gt; #RET: 50;100 #OK</pre>
<GSCNR>	<p>Get signal carrier-to-noise ratio (CNR) in dB</p> <p>Example:</p> <pre>#COMMAND: &lt;GSCNR&gt; #RET: 12.2 dB #OK</pre>
<GSSNR>	<p>Get signal signal-to-noise ratio (SNR) in dB</p> <pre>#RET: 27.0 dB or if SNR is not available #RET: N/A</pre>

<GSRSSI>	<p>Get signal RSSI in dBuV</p> <p>Example:          #COMMAND: &lt;GSRSSI&gt;          #RET: 39.7 dBuV          #OK</p>
<GSQEXT>	<p>Get extended info over signal strength and quality</p> <p>Return format:  <i>SSI;SQI;RSSI;CNR</i></p> <p>where  <i>SSI</i>            SSI (strength) in percent  <i>SQI</i>            SQI (quality) in percent  <i>RSSI</i>          signal RSSI in dBuV  <i>CNR</i>            signal CNR in dB</p> <p>Example:          #COMMAND: &lt;GSQEXT&gt;          #RET: 50;100;39.7 dBuV;12.0 dB          #OK</p>
<GSBER>	<p>Get signal bit-error-rate (BER)</p> <p>#RET: 1E-8          or if BER is not available          #RET: N/A</p>
<GSPER>	<p>Get signal TS Packet Error Ratio (PER)</p> <p>#RET: 1E-8          or if PER is not available          #RET: N/A</p>
<GSFEC>	<p>Get code rate (FEC) for DVB-S/S2 signal</p> <p>#RET: 2/3          or if FEC is not available          #RET: N/A</p>
<GSPSK>	<p>Get demodulated constellation for DVB-S/S2 signal</p> <p>#RET: 8PSK          or if DVB-S/S2 constellation is not available          #RET: N/A</p>
<GSSTD>	<p>Get signal modulation          (DVB-S/DVB-S2/DVB-C/DVB-C2/DVB-T/DVB-T2/DAB/FM)</p> <p>#RET: DVB-S2</p>
<GSUCBLK>	<p>Get signal uncorrectable TS packets counter (since signal lock)</p> <p>#RET: 0</p>
<GSTSBR>	<p>Get TS bit rate estimated by the demodulator</p> <p>#RET: 3308 kbps          or if not available          #RET: N/A</p>

<GSTSCLK>	<p>Get current TS output clock frequency from demodulator            #RET: 45084 kHz            or if not available            #RET: N/A</p>
<DAB METRICS>	<p>Get current DAB metrics</p> <p>Return format:  <i>SSI;SQI;RSSI;CNR;SNR;SIGVALID;ACQVALID;AMODE;ABR</i>            where  <i>SSI</i>            SSI (strength) in percent  <i>SQI</i>            SQI (quality) in percent  <i>RSSI</i>          signal RSSI in dBuV  <i>CNR</i>           signal CNR in dB  <i>SNR</i>           signal SNR in dB  <i>SIGVALID</i>    signal validation (1=valid, 0=invalid)  <i>ACQVALID</i>    ensemble acquisition (1=valid, 0=invalid)  <i>AMODE</i>        audio mode ('Dual', 'Mono', 'Stereo', 'Joint Stereo'                             or '-' if not available)  <i>ABR</i>            audio bitrate in kbps or '-' if not available</p> <p>Example:            #COMMAND: &lt;DAB METRICS&gt;            #RET: 42;100;27.0 dBuV;16.0 dB;7.0 dB;1;1;Stereo;72 kbps            #OK</p>
<DAB DLS>	<p>Get DLS text of the currently playing DAB channel</p> <p>Example:            #COMMAND: &lt;DAB DLS&gt;            #RET: MISSING YOU - JOHN WAITE            #OK</p>
<DAB DLSSTATUS>	<p>Get DLS text and status of the currently playing DAB channel</p> <p>Return format:  <i>status;timestamp;dls</i>            where  <i>status</i>            DLS update status                              (1 = DLS received, 0 = DLS empty)  <i>timestamp</i>        DLS reception timestamp in format                              dd.mm.yyyy HH:MM:SS  <i>dls</i>                DLS text</p> <p>Example:            #COMMAND: &lt;DAB DLSSTATUS&gt;            #RET: 1;06.03.2019 14:20:22;GET LUCKY - DAFT PUNK            #OK</p>
<FM METRICS>	<p>Get current FM metrics</p>

	<p>Return format:  <i>SSI;SQI;RSSI;SNR;MPATH;SIGVALID;RSQVALID;AMODE;ASPLT;ASBL</i>          where</p> <table border="0"> <tr> <td><i>SSI</i></td> <td>SSI (strength) in percent</td> </tr> <tr> <td><i>SQI</i></td> <td>SQI (quality) in percent</td> </tr> <tr> <td><i>RSSI</i></td> <td>signal RSSI in dBuV</td> </tr> <tr> <td><i>SNR</i></td> <td>signal SNR in dB</td> </tr> <tr> <td><i>MPATH</i></td> <td>signal multipath</td> </tr> <tr> <td><i>SIGVALID</i></td> <td>signal validation (1=valid, 0=invalid)</td> </tr> <tr> <td><i>RSQVALID</i></td> <td>quality validation (1=valid, 0=invalid)</td> </tr> <tr> <td><i>AMODE</i></td> <td>audio mode ('Mono', 'Stereo' or '-' if not available)</td> </tr> <tr> <td><i>ASPLT</i></td> <td>stereo pilot indicator (1=present, 0=not present)</td> </tr> <tr> <td><i>ASBL</i></td> <td>stereo blend (separation) in percent</td> </tr> </table> <p>Example:          #COMMAND: &lt;FM METRICS&gt;          #RET: 100;100;67.0 dBuV;46.0 dB;2;1;1;Stereo;1;100          #OK</p>	<i>SSI</i>	SSI (strength) in percent	<i>SQI</i>	SQI (quality) in percent	<i>RSSI</i>	signal RSSI in dBuV	<i>SNR</i>	signal SNR in dB	<i>MPATH</i>	signal multipath	<i>SIGVALID</i>	signal validation (1=valid, 0=invalid)	<i>RSQVALID</i>	quality validation (1=valid, 0=invalid)	<i>AMODE</i>	audio mode ('Mono', 'Stereo' or '-' if not available)	<i>ASPLT</i>	stereo pilot indicator (1=present, 0=not present)	<i>ASBL</i>	stereo blend (separation) in percent
<i>SSI</i>	SSI (strength) in percent																				
<i>SQI</i>	SQI (quality) in percent																				
<i>RSSI</i>	signal RSSI in dBuV																				
<i>SNR</i>	signal SNR in dB																				
<i>MPATH</i>	signal multipath																				
<i>SIGVALID</i>	signal validation (1=valid, 0=invalid)																				
<i>RSQVALID</i>	quality validation (1=valid, 0=invalid)																				
<i>AMODE</i>	audio mode ('Mono', 'Stereo' or '-' if not available)																				
<i>ASPLT</i>	stereo pilot indicator (1=present, 0=not present)																				
<i>ASBL</i>	stereo blend (separation) in percent																				
<FM RDS>	<p>Get RDS info of the currently playing FM channel</p> <p>Return format:  <i>pi;pty;ptystr;pst;rt</i>          where</p> <table border="0"> <tr> <td><i>pi</i></td> <td>RDS PI (program id) in hex, e.g. D391</td> </tr> <tr> <td><i>pty</i></td> <td>RDS PTY (program type) code in dec, e.g. 10</td> </tr> <tr> <td><i>ptystr</i></td> <td>RDS PTY (program type) string representation, e.g. 'Pop Music' or '-' if not available</td> </tr> <tr> <td><i>pst</i></td> <td>RDS PST (program station name)</td> </tr> <tr> <td><i>rt</i></td> <td>RDS RT (radio text)</td> </tr> </table> <p>Example:          #COMMAND: &lt;FM RDS&gt;          #RET: D391;10;Pop Music;1LIVE;Bruno Mars - 24K Magic          #OK</p>	<i>pi</i>	RDS PI (program id) in hex, e.g. D391	<i>pty</i>	RDS PTY (program type) code in dec, e.g. 10	<i>ptystr</i>	RDS PTY (program type) string representation, e.g. 'Pop Music' or '-' if not available	<i>pst</i>	RDS PST (program station name)	<i>rt</i>	RDS RT (radio text)										
<i>pi</i>	RDS PI (program id) in hex, e.g. D391																				
<i>pty</i>	RDS PTY (program type) code in dec, e.g. 10																				
<i>ptystr</i>	RDS PTY (program type) string representation, e.g. 'Pop Music' or '-' if not available																				
<i>pst</i>	RDS PST (program station name)																				
<i>rt</i>	RDS RT (radio text)																				
<FM RDSSTATUS>	<p>Get RDS info and status of the currently playing FM channel</p> <p>Return format:  <i>status;timestamp;pi;pty;ptystr;pst;rt</i>          where</p> <table border="0"> <tr> <td><i>status</i></td> <td>RDS update status in dec, bitmask of                  0x01 – RDS PI received                  0x02 – RDS PTY received                  0x04 – RDS RT received                  0x08 – RDS PST received</td> </tr> <tr> <td><i>timestamp</i></td> <td>RDS reception/check timestamp in format dd.mm.yyyy HH:MM:SS</td> </tr> <tr> <td><i>pi</i></td> <td>RDS PI (program id) in hex, e.g. D391</td> </tr> <tr> <td><i>pty</i></td> <td>RDS PTY (program type) code in dec, e.g. 10</td> </tr> <tr> <td><i>ptystr</i></td> <td>RDS PTY (program type) string representation,</td> </tr> </table>	<i>status</i>	RDS update status in dec, bitmask of 0x01 – RDS PI received 0x02 – RDS PTY received 0x04 – RDS RT received 0x08 – RDS PST received	<i>timestamp</i>	RDS reception/check timestamp in format dd.mm.yyyy HH:MM:SS	<i>pi</i>	RDS PI (program id) in hex, e.g. D391	<i>pty</i>	RDS PTY (program type) code in dec, e.g. 10	<i>ptystr</i>	RDS PTY (program type) string representation,										
<i>status</i>	RDS update status in dec, bitmask of 0x01 – RDS PI received 0x02 – RDS PTY received 0x04 – RDS RT received 0x08 – RDS PST received																				
<i>timestamp</i>	RDS reception/check timestamp in format dd.mm.yyyy HH:MM:SS																				
<i>pi</i>	RDS PI (program id) in hex, e.g. D391																				
<i>pty</i>	RDS PTY (program type) code in dec, e.g. 10																				
<i>ptystr</i>	RDS PTY (program type) string representation,																				

	<p><i>pst</i> e.g. 'Pop Music' or '-' if not available RDS PST (program station name) <i>rt</i> RDS RT (radio text)</p> <p>Example: #COMMAND: &lt;FM RDSSTATUS&gt; #RET: 5;06.03.2019 14:54:07;D79B;0;-;-;Meine Stadt. Mein Radio. #OK</p>
<FM RDS PST>	<p>Get RDS PST (program station) of the currently playing FM channel</p> <p>Example: #COMMAND: &lt;FM RDS PST&gt; #RET: BERG #OK</p>
<FM RDS RT>	<p>Get RDS RT (radio text) of the currently playing FM channel</p> <p>Example: #COMMAND: &lt;FM RDS RT&gt; #RET: 5 Seconds Of Summer - Valentine #OK</p>
<REC ?>	<p>Get current recording status</p> <p>Returns information for the currently running recording in the following format: #RET: on;&lt;container&gt;;&lt;filename&gt; where the &lt;container&gt; is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive</p> <p>If the recording is currently disabled the command reply will be: #RET: off</p> <p>Examples: #COMMAND: &lt;REC ?&gt; #RET: on;HDD1;ZDF_HD__20171205_170001__heute.ts #OK or #COMMAND: &lt;REC ?&gt; #RET: off #OK</p>
<REC DURATION>	<p>Get current recording duration (total time) in seconds</p> <p>The recording should be previously started (please refer &lt;REC 1&gt; command description). Otherwise the respective error message is sent: #ERROR: Not running</p>

	<p>Example:          #COMMAND: &lt;REC DURATION&gt;          #RET: 4802          #OK</p>
<REC TIME>	<p>Get current recording timeline information including start time and duration (total time) as preformatted strings</p> <p>Return format:          #RET: &lt;start_time&gt;;&lt;duration&gt;          where              &lt;start_time&gt; recording start time as string in format                            dd/mm/yy HH:MM:SS              &lt;duration&gt; recording duration as string in format                            HH:MM:SS</p> <p>The recording should be previously started (please refer &lt;REC 1&gt; command description). Otherwise the respective error message is sent:          #ERROR: Not running</p> <p>Example:          #COMMAND: &lt;REC TIME&gt;          #RET: 07/11/18 10:20:38;01:20:02          #OK</p>
<REC FILES>	<p>Get current recording files list</p> <p>Each recording file in the list is sent in new row in the following format:          #RET: &lt;container&gt;;&lt;filename&gt;;&lt;start_time&gt;;&lt;duration&gt;          where              &lt;container&gt; is the one of the following:                            HDD[1..2] = HDD connected to eSATA                            NAS[1..4] = NAS network drive              &lt;filename&gt; recording file name              &lt;start_time&gt; recording file start time as string in format                            dd/mm/yy HH:MM:SS              &lt;duration&gt; recording duration as string in format                            HH:MM:SS</p> <p>At the end of transmission '#OK' line is sent.</p> <p>The recording could be split into multiple files for the several reasons including EPG program change (split can be disabled: please refer &lt;RECCFG EPGSPPLIT n&gt; command description), audio stream change within the current recording (please refer &lt;SAC n&gt; command description) or stream PSI tables updates (e.g. caused by local channel / dynamic program change).</p> <p>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.</p>

	<p>Example:  #COMMAND: &lt;REC FILES&gt;  #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  #OK</p>
<REC FSIZE>	<p>Get current recording file size</p> <p>Returns information for the file currently used by recording in the following format:  #RET: &lt;file_size&gt;;&lt;container&gt;;&lt;filename&gt;  where  &lt;file_size&gt; recording file size, e.g. '750 MB' or '2.1 GB'  &lt;container&gt; is the one of the following:  HDD[1..2] = HDD connected to eSATA  NAS[1..4] = NAS network drive  &lt;filename&gt; recording file name</p> <p>The recording should be previously started (please refer &lt;REC 1&gt; command description). Otherwise the respective error message is sent:  #ERROR: Not running</p> <p>Example:  #COMMAND: &lt;REC FSIZE&gt;  #RET: 1.2 GB;HDD1;ZDF_HD__20171205_170001__heute.ts  #OK</p>
<RECCFG EPGSPLIT n>	<p>Enable/Disable split of recording into multiple files by EPG  Get current status of the split of recording into multiple files by EPG</p> <p>n = [?]   [ON,OFF]  n = ? – get recordings split status (on = enabled, off = disabled)  n = ON or OFF – enable or disable split of recording by EPG</p> <p>The setting can be changed only when recording is currently disabled. Otherwise the respective error message is sent.</p>
<RECCFG RMDAYS n>	<p>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</p> <p>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)</p>

	<p>Note: Command is used to configure the automatic recordings removal. Please refer &lt;REC RMDAYS n&gt; command to remove recordings manually.</p> <p>Examples:  #COMMAND: &lt;RECCFG RMDAYS 7&gt;  #RET: 7  #OK  or  #COMMAND: &lt;RECCFG RMDAYS ?&gt;  #RET: 7  #OK  or  #COMMAND: &lt;RECCFG RMDAYS OFF&gt;  #RET: off  #OK</p>
<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.
<MPLAY LOOP ?>	Get media player playback 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
<ISPAUSE>	Check whether the playback in the media player or timeshifting is paused  0 : no pause 1 : playback is paused  Example: #COMMAND: <ISPAUSE> #RET: 1 #OK

<p>&lt;GETDURATION&gt;</p>	<p>Get current file playback duration (total time) in seconds for media player / timeshifting</p> <p>Example:            #COMMAND: &lt;GETDURATION&gt;            #RET: 3722            #OK</p>
<p>&lt;GETDURTIME&gt;</p>	<p>Get current file playback duration (total time) as preformatted string for media player / timeshifting</p> <p>Example:            #COMMAND: &lt;GETDURTIME&gt;            #RET: 01:20:02            #OK</p>
<p>&lt;GETPLAYPOS&gt;</p>	<p>Get current file playback position (elapsed time) in seconds for media player / timeshifting</p> <p>Example:            #COMMAND: &lt;GETPLAYPOS&gt;            #RET: 68            #OK</p>
<p>&lt;GETPLAYTIME&gt;</p>	<p>Get current file playback position (elapsed time) as preformatted string for media player / timeshifting</p> <p>Example:            #COMMAND: &lt;GETPLAYTIME&gt;            #RET: 00:01:08            #OK</p>
<p>&lt;GETPLAYFILE&gt;</p>	<p>Get current playback file for media player / timeshifting</p> <p>Examples:            #COMMAND: &lt;GETPLAYFILE&gt;            #RET: PVR;HDD1;arte__20171201_203000__News.ts            #OK            or            #COMMAND: &lt;GETPLAYFILE&gt;            #RET: MOVIE;NAS1;MyMovie1.avi            #OK</p>
<p>&lt;TSHFT GETDURATION&gt;</p>	<p>Get total playback duration (total time) in seconds for timeshifting</p> <p>The command (unlike &lt;GETDURATION&gt; 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</p>

	duration) with the respects to the buffering inside timeshifting.
<TSHFT GETDURTIME>	<p>Get total playback duration (total time) as preformatted string for timeshifting</p> <p>The command (unlike &lt;GETDURTIME&gt; 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.</p>
<TSHFT GETPLAYPOS>	<p>Get total playback position (elapsed time) in seconds for timeshifting</p> <p>The command (unlike &lt;GETPLAYPOS&gt; 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 &lt;GETPLAYPOS&gt; command.</p>
<TSHFT GETPLAYTIME>	<p>Get total playback position (elapsed time) as preformatted string for timeshifting</p> <p>The command (unlike &lt;GETPLAYTIME&gt; 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 &lt;GETPLAYTIME&gt; command.</p>
<GRL>	<p>Get available PVR recordings list</p> <p>Each available recording in the list is sent in new row in the following format: #RET: PVR;&lt;container&gt;;&lt;filename&gt; At the end of transmission '#OK' line is sent.</p> <p>The &lt;container&gt; is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive</p> <p>Example: #COMMAND: &lt;GRL&gt; #RET: PVR;HDD1;arte__20171201_203000__News.ts #RET: PVR;NAS1;ZDF__20190117_213005__Sport.ts #OK</p>
<GML>	<p>Get available media player movies list</p> <p>Each available movie in the list is sent in new row in the following format: #RET: MOVIE;&lt;container&gt;;&lt;filename&gt;</p>

	<p>At the end of transmission '#OK' line is sent.</p> <p>The &lt;container&gt; is the one of the following:  HDD[1..2] = HDD connected to eSATA  NAS[1..4] = NAS network drive</p> <p>Example:  #COMMAND: &lt;GML&gt;  #RET: MOVIE;HDD1;MyMovie1.avi  #RET: MOVIE;NAS4;MyMovie2.m2ts  #OK</p>
<GAL>	<p>Get available audio files list</p> <p>Each available audio file in the list is sent in new row in the following format:  #RET: MUSIC;&lt;container&gt;;&lt;filename&gt;  At the end of transmission '#OK' line is sent.</p> <p>The &lt;container&gt; is the one of the following:  HDD[1..2] = HDD connected to eSATA  NAS[1..4] = NAS network drive</p> <p>Example:  #COMMAND: &lt;GAL&gt;  #RET: MUSIC;HDD1;MySong1.mp3  #RET: MUSIC;NAS3;MySong2.MP3  #OK</p>
<HDD INFO>	<p>Get eSATA HDD status information</p> <p>Examples:  When 500GB HDD is connected and partitioned  The partition is formatted (NTFS), 430GB from total 500GB is available  #COMMAND: &lt;HDD INFO&gt;  #HDD1: 500 GB  #PART1: NTFS;430 GB;500 GB  #OK  When 500GB HDD is connected and partitioned  The partition is not formatted  #COMMAND: &lt;HDD INFO&gt;  #HDD1: 500 GB  #PART1: Not formatted  #OK  When 500GB HDD is connected but not partitioned  #COMMAND: &lt;HDD INFO&gt;  #HDD1: 500 GB  #PART1: Not available  #OK</p> <p>When HDD is not connected</p>

	<p>#COMMAND: &lt;HDD INFO&gt;          #HDD1: Not available          #PART1: Not available          #OK</p>
<HDD FORMAT>	<p>Format eSATA HDD (perform partitioning and formatting)</p> <p>The operation progress can be observed during the formatting process. Device sends progress information as:          #RET: <i>progress_percent</i>          Device sends error message if the formatting process fails for some reason:          #ERROR: <i>error_description</i></p> <p>Example:          #COMMAND: &lt;HDD FORMAT&gt;          #RET: 20%          #RET: 40%          #RET: 60%          #RET: 80%          #RET: 100%          #OK</p>
<NAS GET CONFIG n>	<p>Get NAS network drive configuration</p> <p>n = 1..4 – NAS network drive number</p> <p>Returns NAS network drive settings in the following format:          #RET: &lt;type&gt;;&lt;ip&gt;;&lt;share&gt;;&lt;enabled&gt;;&lt;user&gt;;&lt;pwd&gt;;&lt;status&gt;          where</p> <ul style="list-style-type: none"> <li>&lt;type&gt; is the one of the following:              CIFS = Samba (Windows) network file server              NFS = NFS (Linux/Unix) network file server</li> <li>&lt;ip&gt; IP-address of the network file server</li> <li>&lt;share&gt; network shared resource name</li> <li>&lt;enabled&gt; is the one of the following:              on = NAS drive is enabled (used by automounter)              off = NAS drive is disabled (not used)</li> <li>&lt;user&gt; network file server user name (only for CIFS)              empty for anonymous/guest login</li> <li>&lt;pwd&gt; network file server user password (only for CIFS)              empty for anonymous/guest login</li> <li>&lt;status&gt; 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</li> </ul>

	<p>Example:          #COMMAND: &lt;NAS GET CONFIG 1&gt;          #RET: CIFS;10.1.1.200;Public;on;;;on          #OK</p>
<p>&lt;NAS SET CONFIG n;s&gt;</p>	<p>Set NAS network drive configuration</p> <p>n = 1..4 – NAS network drive number          s = NAS network drive settings as the list of string attributes separated with ";" delimiter in format:              &lt;type&gt;;&lt;ip&gt;;&lt;share&gt;;&lt;enabled&gt;;&lt;user&gt;;&lt;pwd&gt;</p> <p>where</p> <ul style="list-style-type: none"> <li>&lt;type&gt; 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</li> <li>&lt;ip&gt; IP-address of the network file server              or "-" to disable network drive and remove settings</li> <li>&lt;share&gt; network shared resource name              or "-" to disable network drive and remove settings</li> <li>&lt;enabled&gt; 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</li> <li>&lt;user&gt; network file server user name (only for CIFS)              "-" if empty for anonymous/guest login</li> <li>&lt;pwd&gt; network file server user password (only for CIFS)              "-" if empty for anonymous/guest login</li> </ul> <p>The command reply is the same as for &lt;NAS GET CONFIG n&gt; command.</p> <p>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.</p>
<p>&lt;NAS c n&gt;</p>	<p>Enable/Disable NAS network drive          Get NAS network drive status</p> <p>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</p> <p>NAS network drive can be enabled only when it is properly configured (please refer &lt;NAS SET CONFIG n;s&gt; 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.</p>

	<p>The return value can be the one of the following:</p> <p>#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</p> <p>Example:          #COMMAND: &lt;NAS ON 1&gt;          #RET: readonly          #OK</p>
<p>&lt;NAS DEF STORAGE c&gt;</p>	<p>Enable/Disable NAS as PVR preferred storage          Get current status of NAS as PVR preferred storage setting</p> <p>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</p> <p>The command should be used only when both HDD and NAS network drives are available for PVR recording.</p> <p>Alternatively the PVR preferred storage setting can be set with &lt;RECCFG PREFSTORAGE n&gt; command (please refer the command description).</p>
<p>&lt;SNMP COMMUNITY n&gt;</p>	<p>Set SNMP community string          Get current SNMP community string</p> <p>n = [?]   [<i>STRING (SIZE (1..32))</i>]          n = ? - get current SNMP community string          n = <i>STRING (SIZE (1..32))</i> - set SNMP community string</p> <p>The valid characters for community string are:          'A..Z'   'a..z'   '0..9'   '_'   '-' (string should not start with '_' and '-').</p> <p>Examples:          #COMMAND: &lt;SNMP COMMUNITY ?&gt;          #RET: public          #OK          or          #COMMAND: &lt;SNMP COMMUNITY elium-private&gt;          #RET: elium-private          #OK</p>
<p>&lt;SNMP n&gt;</p>	<p>Enable/Disable SNMP agent          Get SNMP agent status</p> <p>n = [?]   [ON,OFF]</p>

	<p>n = ? – get SNMP agent status (on = enabled, off = disabled)  n = ON or OFF – enable or disable SNMP agent</p> <p>Note: SNMP TRAP notifications still can be used when SNMP agent is disabled (please refer &lt;SNMPTRAP n&gt; command description).</p>
<SNMPTRAP n>	<p>Enable/Disable SNMP TRAP notifications  Get SNMP TRAP notifications status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP TRAP notifications status (on/off = enabled/disabled)  n = ON or OFF – turn SNMP TRAP notifications enabled or disabled</p> <p>Note: SNMP TRAP host IP address should be set before the SNMP TRAP notifications are turned ON (please refer &lt;SNMPTRAP HOST n&gt; command description).</p>
<SNMPTRAP HOST n>	<p>Set SNMP TRAP host IP address  Get SNMP TRAP host IP address</p> <p>n = [?]   [IP-ADDRESS]  n = ? – get SNMP TRAP host IP address  n = IP-ADDRESS – set SNMP TRAP host IP address</p> <p>Example:  #COMMAND: &lt;SNMPTRAP HOST 10.1.1.220&gt;  #RET: 10.1.1.220  #OK</p>
<SNMPTRAP DELAY n>	<p>Set SNMP TRAP notifications delay value  Get current SNMP TRAP notifications delay value value</p> <p>n = [?]   [1 .. 60]  n = ? – get current SNMP TRAP notifications delay value (in minutes)  n = [1 .. 60] – set SNMP TRAP notifications delay value (in minutes)</p> <p>SNMP TRAP notifications delay value represents the delay in minutes between two consecutive alarm notifications of the same type.</p> <p>Example:  #COMMAND: &lt;SNMPTRAP DELAY 5&gt;  #RET: 5  #OK</p>
<SNMPTRAP alarmNotifications n>	<p>Enable/Disable SNMP TRAP alarm notifications  Get SNMP TRAP alarm notifications status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP TRAP alarm notifications status  n = ON/OFF – turn SNMP TRAP alarm notifications enabled/disabled</p>

	<p>Device alarm notifications are generated continuously with respective delay (please refer &lt;SNMPTRAP DELAY n&gt;command description) between two consecutive notifications until alarm disappears.</p> <p>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.</p>
<SNMPTRAP noSignal n>	<p>Enable/Disable SNMP noSignal TRAP alarm notification Get SNMP noSignal TRAP alarm notification status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP noSignal TRAP status n = ON/OFF – turn SNMP noSignal TRAP enabled/disabled</p> <p>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.</p>
<SNMPTRAP noStream n>	<p>Enable/Disable SNMP noStream TRAP alarm notification Get SNMP noStream TRAP alarm notification status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP noStream TRAP status n = ON/OFF – turn SNMP noStream TRAP enabled/disabled</p> <p>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.</p>
<SNMPTRAP scrambledStream n>	<p>Enable/Disable SNMP scrambledStream TRAP alarm notification Get SNMP scrambledStream TRAP alarm notification status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP scrambledStream TRAP status n = ON/OFF – turn SNMP scrambledStream TRAP enabled/disabled</p> <p>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.</p>
<SNMPTRAP firmwareEvents n>	<p>Enable/Disable SNMP TRAP firmware events Get SNMP TRAP firmware events status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP TRAP firmware events status</p>

	<p>n = ON/OFF – turn SNMP TRAP firmware events enabled/disabled</p> <p>Firmware event notifications are generated once pro each event.</p> <p>Command enables/disables all respective notifications at once and reports 'ON' in the case when at least one notification is enabled.</p> <p>To set each notification apart please use respective commands below.</p>
<SNMPTRAP firmwareReady n>	<p>Enable/Disable SNMP firmwareReady TRAP firmware event Get SNMP firmwareReady TRAP firmware event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP firmwareReady TRAP status n = ON/OFF – turn SNMP firmwareReady TRAP enabled/disabled</p> <p>The notification is generated after the device software startup. It notifies that the device is initialized and ready for use.</p>
<SNMPTRAP firmwareFault n>	<p>Enable/Disable SNMP firmwareFault TRAP firmware event Get SNMP firmwareFault TRAP firmware event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP firmwareFault TRAP status n = ON/OFF – turn SNMP firmwareFault TRAP enabled/disabled</p> <p>The notification is generated in the case when the device software reaches hard fault and is normally followed by the software restart.</p>
<SNMPTRAP firmwareUpdate n>	<p>Enable/Disable SNMP firmwareUpdate TRAP firmware event Get SNMP firmwareUpdate TRAP firmware event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP firmwareUpdate TRAP status n = ON/OFF – turn SNMP firmwareUpdate TRAP enabled/disabled</p> <p>The notification is sent when the device firmware update starts.</p>
<SNMPTRAP fwUpdateFAIL n>	<p>Enable/Disable SNMP fwUpdateFAIL TRAP firmware event Get SNMP fwUpdateFAIL TRAP firmware event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP fwUpdateFAIL TRAP status n = ON/OFF – turn SNMP fwUpdateFAIL TRAP enabled/disabled</p> <p>The notification is sent in the case when the device firmware update process has failed for some reason.</p>
<SNMPTRAP standbyEvents n>	<p>Enable/Disable SNMP TRAP standby events Get SNMP TRAP standby events status</p>

	<p>n = [?]   [ON,OFF]  n = ? – get SNMP TRAP standby events status  n = ON/OFF – turn SNMP TRAP standby events enabled/disabled</p> <p>Standby event notifications are generated once pro each event.</p> <p>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.</p>
<SNMPTRAP enterStandby n>	<p>Enable/Disable SNMP enterStandby TRAP standby event  Get SNMP enterStandby TRAP standby event status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP enterStandby TRAP status  n = ON/OFF – turn SNMP enterStandby TRAP enabled/disabled</p> <p>The notification is sent when the device enters Standby.</p>
<SNMPTRAP leaveStandby n>	<p>Enable/Disable SNMP leaveStandby TRAP standby event  Get SNMP leaveStandby TRAP standby event status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP leaveStandby TRAP status  n = ON/OFF – turn SNMP leaveStandby TRAP enabled/disabled</p> <p>The notification is sent when the device leaves Standby.</p>
<SNMPTRAP playbackEvents n>	<p>Enable/Disable SNMP TRAP playback events  Get SNMP TRAP playback events status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP TRAP playback events status  n = ON/OFF – turn SNMP TRAP playback events enabled/disabled</p> <p>Playback event notifications are generated once pro each event.</p> <p>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.</p>
<SNMPTRAP playbackStart n>	<p>Enable/Disable SNMP playbackStart TRAP playback event  Get SNMP playbackStart TRAP playback event status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP playbackStart TRAP status  n = ON/OFF – turn SNMP playbackStart TRAP enabled/disabled</p> <p>The notification is sent when the device starts playback. It also signifies</p>

	that the current playback status was changed: e.g. on playback mode change or on channel switching.
<SNMPTRAP playbackStop n>	<p>Enable/Disable SNMP playbackStop TRAP playback event Get SNMP playbackStop TRAP playback event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP playbackStop TRAP status n = ON/OFF – turn SNMP playbackStop TRAP enabled/disabled</p> <p>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.</p>
<SNMPTRAP recordingEvents n>	<p>Enable/Disable SNMP TRAP recording events Get SNMP TRAP recording events status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP TRAP recording events status n = ON/OFF – turn SNMP TRAP recording events enabled/disabled</p> <p>Recording event notifications are generated once pro each event.</p> <p>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.</p>
<SNMPTRAP recordingStart n>	<p>Enable/Disable SNMP recordingStart TRAP recording event Get SNMP recordingStart TRAP recording event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP recordingStart TRAP status n = ON/OFF – turn SNMP recordingStart TRAP enabled/disabled</p> <p>The notification is sent when the device starts DVB recording.</p>
<SNMPTRAP recordingStop n>	<p>Enable/Disable SNMP recordingStop TRAP recording event Get SNMP recordingStop TRAP recording event status</p> <p>n = [?]   [ON,OFF] n = ? – get SNMP recordingStop TRAP status n = ON/OFF – turn SNMP recordingStop TRAP enabled/disabled</p> <p>The notification is sent when the device stops DVB recording.</p>
<SNMPTRAP streamingEvents n>	<p>Enable/Disable SNMP TRAP streaming events Get SNMP TRAP streaming events status</p> <p>n = [?]   [ON,OFF]</p>

	<p>n = ? – get SNMP TRAP streaming events status  n = ON/OFF – turn SNMP TRAP streaming events enabled/disabled</p> <p>Streaming event notifications are generated once pro each event.</p> <p>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.</p>
<SNMPTRAP streamingStart n>	<p>Enable/Disable SNMP streamingStart TRAP streaming event  Get SNMP streamingStart TRAP streaming event status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP streamingStart TRAP status  n = ON/OFF – turn SNMP streamingStart TRAP enabled/disabled</p> <p>The notification is sent when the device starts IPTV streaming.</p>
<SNMPTRAP streamingStop n>	<p>Enable/Disable SNMP streamingStop TRAP streaming event  Get SNMP streamingStop TRAP streaming event status</p> <p>n = [?]   [ON,OFF]  n = ? – get SNMP streamingStop TRAP status  n = ON/OFF – turn SNMP streamingStop TRAP enabled/disabled</p> <p>The notification is sent when the device stops IPTV streaming.</p>
<CEC CONTROL n>	<p>Enable/Disable HDMI-CEC remote control  Get HDMI-CEC remote control status</p> <p>n = [?]   [ON,OFF]  n = ? – get HDMI-CEC remote control status (on/off = enabled/disabled)  n = ON/OFF – turn HDMI-CEC remote control enabled/disabled</p> <p>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).</p>
<CEC STANDBY n>	<p>Enable/Disable HDMI-CEC automatic standby  Get HDMI-CEC automatic standby status</p> <p>n = [?]   [ON,OFF]  n = ? – get HDMI-CEC automatic standby status  n = ON/OFF – turn HDMI-CEC automatic standby enabled/disabled</p> <p>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.</p>

<p>&lt;GSS&gt;</p>	<p>Get RS232 Baud Rate</p> <p>Example:          #COMMAND: &lt;GSS&gt;          #RET: 115200          #OK</p>
<p>&lt;GPI&gt;</p>	<p>Get Program Info</p> <p>This command returns an information regarding the current program:</p> <ul style="list-style-type: none"> <li>- current mode (TV or Radio)</li> <li>- number of the channel</li> <li>- name of the channel</li> <li>- title of the currently broadcasted event</li> <li>- duration of the currently broadcasted event (in seconds)</li> <li>- time remaining to the end of the event (in seconds)</li> </ul> <p>#RET: TV;7;ProSieben;taff.;3660;1262</p> <p>This data is taken from Transport Stream. Therefore, host should wait some time before &lt;GPI&gt; command is sent (typically several seconds) after switching on a new channel. Otherwise, empty strings may be returned:</p> <p>#RET: ;;;;</p>
<p>&lt;GTI&gt;</p>	<p>Get current Multiplex (Transponder) Info</p> <p>This command returns information string for the current transponder including standard (DVB-S/-C/-T, <b>IPTV, DAB, FM</b>), satellite, frequency, symbol rate etc.</p> <p>Examples:</p> <p>#COMMAND: &lt;GTI&gt;          #RET: DVB-S Astra (19.2E) 11494 H 22000          #OK</p> <p>or</p> <p>#COMMAND: &lt;GTI&gt;          #RET: IPTV TS udp://239.35.10.55:1234          #OK</p> <p>or</p> <p>#COMMAND: &lt;GTI&gt;          #RET: IPTV URL http://send.blackbeats.fm:13000          #OK</p> <p>or</p> <p>#COMMAND: &lt;GTI&gt;          #RET: DAB (5C) 178.352 MHz          #OK</p> <p>or</p> <p>#COMMAND: &lt;GTI&gt;          #RET: FM 102.400 MHz          #OK</p>

<p>&lt;GCT&gt;</p>	<p>Get Current Time (Day, Month, Year, Hour, Minute)</p> <p>Example:          #COMMAND: &lt;GCT&gt;          #RET: 24;10;07;11;26          #OK</p>
<p>&lt;NTP n&gt;</p>	<p>Enable/Disable time sync via NTP          Get current status of the time sync via NTP</p> <p>n = [?]   [ON,OFF]          n = ? - get NTP time sync status          n = ON/OFF - enable/disable time sync via NTP</p>
<p>&lt;NTP SERVER n&gt;</p>	<p>Set NTP server used for time sync          Get current NTP server used for time sync</p> <p>n = [?]   [NTP_SERVER]          n = ? - get current NTP server setting          n = NTP_SERVER - set NTP server (IP-address or hostname)</p> <p>Examples:          #COMMAND: &lt;NTP SERVER de.pool.ntp.org&gt;          #RET: de.pool.ntp.org          #OK          or          #COMMAND: &lt;NTP SERVER 10.1.1.254&gt;          #RET: 10.1.1.254          #OK</p>
<p>&lt;GTC&gt;          &lt;GTC C&gt;</p>	<p>Get temperature from CPU thermal sensor</p> <p>Example:          #COMMAND: &lt;GTC C&gt;          #RET: 57          #OK</p>
<p>&lt;SUBSCR          REC TIME n&gt;</p>	<p>Subscribe to / Disable the notification of the recording duration          Get subscription status of the notification of the recording duration</p> <p>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</p> <p>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.</p>

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

	<p>continuously every n seconds</p> <p>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.</p> <p>Example: #SUBSCR TSHFT POS: 01:24:32 #SUBSCR TSHFT POS: 01:24:33 ... #SUBSCR TSHFT POS: 01:24:43</p>
<p>&lt;SUBSCR TSHFT BFR n&gt;</p>	<p>Subscribe to / Disable the notification of the timeshift buffering Get subscription status of the notification of the timeshift buffering</p> <p>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</p> <p>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.</p>
<p>&lt;SUBSCR FPLAY POS n&gt;</p>	<p>Subscribe to / Disable the notification of the file playback timeline Get subscription status of the notification of the file playback timeline</p> <p>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</p> <p>If enabled, the notification of the file playback timeline including position inside and duration of the currently played file in format: #SUBSCR FPLAY POS: &lt;position&gt;;&lt;duration&gt; where     &lt;position&gt;    playback position in format HH:MM:SS     &lt;duration&gt;    playback duration in format HH:MM:SS will be sent continuously every n seconds during file playback or timeshift.</p> <p>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</p>

<p>&lt;SUBSCR FPLAY FILE n&gt;</p>	<p>Subscribe to / Disable the notification of the playback file change Get subscription status of the notification of the playback file change</p> <p>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</p> <p>If enabled, the notification of the playback file change in format: #SUBSCR FPLAY FILE: &lt;type&gt;;&lt;container&gt;;&lt;filename&gt; where</p> <p>&lt;type&gt; 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</p> <p>&lt;container&gt; is the one of the following: HDD[1..2] = HDD connected to eSATA NAS[1..4] = NAS network drive</p> <p>&lt;filename&gt; playback file name (path) will be sent each time when the file currently used by playback changes during media player playback or timeshifting.</p> <p>Example: #SUBSCR FPLAY FILE: PVR;HDD1;arte__20181107_162652__Sports.ts ... #SUBSCR FPLAY FILE: PVR;HDD1;arte__20181107_162717__heute.ts</p>
--	---