



# HW-LUMA-E4211 | HW-LUMA-D4111

LumaStream Encoder & Decoder

## API Command Reference

April 2026



# Table of Contents

|                                 |    |
|---------------------------------|----|
| Table of Contents .....         | 2  |
| Command Quick Reference .....   | 8  |
| Overview .....                  | 13 |
| About This Document .....       | 13 |
| Connection .....                | 13 |
| JSON-RPC 2.0 Protocol .....     | 13 |
| Data Types .....                | 14 |
| Shared Commands .....           | 15 |
| System Commands .....           | 15 |
| Platform.FactoryReset .....     | 15 |
| Platform.Reboot .....           | 15 |
| System.Get .....                | 15 |
| SystemCurrentTime.Get .....     | 15 |
| TimeSync.Set .....              | 16 |
| TimeSync.Get .....              | 16 |
| DeviceDiscovery.Get .....       | 16 |
| DeviceDiscovery.Set .....       | 17 |
| Network Commands .....          | 17 |
| Network.Get .....               | 17 |
| Network.Set .....               | 17 |
| NetworkHostname.Get .....       | 18 |
| NetworkHostname.Set .....       | 18 |
| Security Commands .....         | 18 |
| LoginPassword.Set .....         | 18 |
| Verify8021X.Get .....           | 19 |
| Verify8021X.Set .....           | 19 |
| AuthEnabled.Set .....           | 20 |
| AuthEnabled.Get .....           | 20 |
| Auth.Set .....                  | 20 |
| TelnetEnable.Get .....          | 20 |
| TelnetEnable.Set .....          | 21 |
| SshEnable.Get .....             | 21 |
| SshEnable.Set .....             | 21 |
| Front Panel Commands .....      | 21 |
| FrontPanelLockButtons.Set ..... | 21 |

|                                        |    |
|----------------------------------------|----|
| FrontPanelLockButtons.Get .....        | 22 |
| FrontPanelLed.Set .....                | 22 |
| FrontPanelLed.Get .....                | 22 |
| FrontPanelBlinkLed.Set .....           | 22 |
| OSD Commands .....                     | 22 |
| OsdText.Get .....                      | 22 |
| OsdText.Set .....                      | 23 |
| OsdLogo.Get .....                      | 23 |
| OsdLogo.Set .....                      | 23 |
| Thumbnail Commands .....               | 24 |
| ThumbnailStatus.Get .....              | 24 |
| ThumbnailStatus.Set .....              | 24 |
| HDCP Scrambling Commands .....         | 24 |
| Scrambling.Set .....                   | 24 |
| Scrambling.Get .....                   | 25 |
| Network Port Status Commands .....     | 25 |
| AllEthernetLanStatus.Get .....         | 25 |
| AllEthernetStatus.Get .....            | 25 |
| Serial Port Commands .....             | 26 |
| SerialTransParam.Get .....             | 26 |
| SerialControlParams.Get .....          | 26 |
| SerialControlParams.Reset .....        | 26 |
| SerialControlMode.Set .....            | 26 |
| SerialControlProxy.Set .....           | 26 |
| ControlSerialCommand.Set .....         | 27 |
| ControlSerialParam.Set .....           | 27 |
| IR Control Commands .....              | 27 |
| IrControlParams.Get .....              | 27 |
| IrControlParams.Reset .....            | 27 |
| IrControlMode.Set .....                | 27 |
| IrControlProxy.Set .....               | 28 |
| ControlIrCommand.Set .....             | 28 |
| Encoder Commands (HW-LUMA-E4211) ..... | 29 |
| System Commands (Encoder) .....        | 29 |
| SystemFuncPowerStatus.Set .....        | 29 |
| SystemFuncPowerStatus.Get .....        | 29 |
| SystemFuncDantePowerStatus.Set .....   | 29 |
| SystemFuncDantePowerStatus.Get .....   | 29 |

|                                                 |           |
|-------------------------------------------------|-----------|
| PowerConsumption.Get .....                      | 30        |
| SystemLcdBrigtness.Set .....                    | 30        |
| SystemLcdBrigtness.Get .....                    | 30        |
| DanteInfo.Get .....                             | 30        |
| NetworkSpeed.Get .....                          | 31        |
| NetworkSpeed.Set .....                          | 31        |
| Dscp.Get .....                                  | 31        |
| Dscp.Set .....                                  | 31        |
| <b>HDCP &amp; EDID Commands (Encoder) .....</b> | <b>32</b> |
| HdcpAnnouncement.Get .....                      | 32        |
| HdcpAnnouncement.Set .....                      | 32        |
| DisplayHdcpStatus.Get .....                     | 32        |
| EdidInput.Get .....                             | 32        |
| EdidInput.Set .....                             | 33        |
| EdidInputBuildFile.Set .....                    | 33        |
| EdidMem1File.Get .....                          | 33        |
| <b>Video Commands (Encoder) .....</b>           | <b>33</b> |
| VideoInputStatus.Get .....                      | 33        |
| VideoInputMute.Get .....                        | 34        |
| VideoInputMute.Set .....                        | 34        |
| MainStreamVideoEncode.Get .....                 | 34        |
| MainStreamVideoEncode.Set .....                 | 35        |
| MainStreamVideoQp.Get .....                     | 35        |
| MainStreamVideoQp.Set .....                     | 35        |
| ProfileSelection.Get .....                      | 35        |
| ProfileSelection.Set .....                      | 36        |
| ImageDisplay.Get .....                          | 36        |
| ImageDisplay.Set .....                          | 36        |
| ImageReset.Set .....                            | 36        |
| <b>Video Streaming Commands (Encoder) .....</b> | <b>37</b> |
| VideoMultilpStream.Get .....                    | 37        |
| VideoMultilpStream.Set .....                    | 37        |
| StreamTtl.Get .....                             | 37        |
| StreamTtl.Set .....                             | 37        |
| DanteStreamLanSelection.Get .....               | 38        |
| DanteStreamLanSelection.Set .....               | 38        |
| <b>RTMP Commands (Encoder) .....</b>            | <b>38</b> |
| Rtmp.Get .....                                  | 38        |

|                                                                |           |
|----------------------------------------------------------------|-----------|
| Rtmp.Set .....                                                 | 38        |
| RtmpRunStatus.Get .....                                        | 39        |
| <b>Preview Stream Commands (Encoder) .....</b>                 | <b>39</b> |
| PreviewStreamEnable.Get .....                                  | 39        |
| PreviewStreamEnable.Set .....                                  | 39        |
| PreviewClientCnt.Get .....                                     | 40        |
| <b>Audio Commands (Encoder) .....</b>                          | <b>40</b> |
| AudioInputFormat.Get .....                                     | 40        |
| AnalogAudioInputStatus.Get .....                               | 40        |
| AudioInSelection.Get .....                                     | 40        |
| AudioInSelection.Set .....                                     | 41        |
| AudioOutputVol.Get .....                                       | 41        |
| AudioOutputVol.Set .....                                       | 41        |
| AudioInputMute.Get .....                                       | 41        |
| AudioInputMute.Set .....                                       | 42        |
| AudioInputDownmixing.Get .....                                 | 42        |
| AudioInputDownmixing.Set .....                                 | 42        |
| AudioEncode.Get .....                                          | 42        |
| AudioEncode.Set .....                                          | 43        |
| <b>AES Encryption Commands (Encoder) .....</b>                 | <b>43</b> |
| AesEncryptEnabled.Get .....                                    | 43        |
| AesEncryptEnabled.Set .....                                    | 43        |
| AesEncryptPassword.Get .....                                   | 43        |
| AesEncryptPassword.Set .....                                   | 44        |
| AesEncryptPassword.Reset .....                                 | 44        |
| <b>AudioLAN &amp; Dante Streaming Commands (Encoder) .....</b> | <b>44</b> |
| AudiolanEnabled.Get .....                                      | 44        |
| AudiolanEnabled.Set .....                                      | 45        |
| AudioMultilpStream.Get .....                                   | 45        |
| AudioMultilpStream.Set .....                                   | 45        |
| <b>Decoder Commands (HW-LUMA-D4111) .....</b>                  | <b>46</b> |
| <b>HDCP &amp; CEC Commands (Decoder) .....</b>                 | <b>46</b> |
| OutHdcpAnnouncement.Get .....                                  | 46        |
| OutHdcpAnnouncement.Set .....                                  | 46        |
| OutHdcpAnnouncementTest.Set .....                              | 46        |
| VideoHDMIOut5V.Get .....                                       | 46        |
| VideoHDMIOut5V.Set .....                                       | 47        |
| CecCommand.Get .....                                           | 47        |

|                                                              |           |
|--------------------------------------------------------------|-----------|
| CecCommand.Set .....                                         | 47        |
| CecCommandControl.Set .....                                  | 48        |
| <b>Video Output Commands (Decoder) .....</b>                 | <b>48</b> |
| VideoOutputStatus.Get .....                                  | 48        |
| NoVideoSource.Get .....                                      | 48        |
| NoVideoSource.Set .....                                      | 48        |
| VideoOutColorspace.Get .....                                 | 49        |
| VideoOutColorspace.Set .....                                 | 49        |
| VideoOutSupportedTimingList.Get .....                        | 49        |
| VideoOutTiming.Get .....                                     | 50        |
| VideoOutTiming.Set .....                                     | 50        |
| LowLatencyMode.Get .....                                     | 50        |
| LowLatencyMode.Set .....                                     | 50        |
| VideoOutColorRange.Get .....                                 | 50        |
| VideoOutColorRange.Set .....                                 | 51        |
| ImageReset.Set .....                                         | 51        |
| DisplayEdid.Get .....                                        | 51        |
| <b>Video Stream Selection Commands (Decoder) .....</b>       | <b>51</b> |
| VideoStreamNameList.Get .....                                | 51        |
| SelectVideoStream.Get .....                                  | 52        |
| SelectVideoStream.Set .....                                  | 52        |
| <b>Audio Output Commands (Decoder) .....</b>                 | <b>52</b> |
| AnalogAudioOutputVol.Set .....                               | 52        |
| AudioOutputMute.Get .....                                    | 53        |
| AudioOutputMute.Set .....                                    | 53        |
| AudioOutputDownmixing.Get .....                              | 53        |
| AudioOutputDownmixing.Set .....                              | 53        |
| AudioOutDelay.Get .....                                      | 54        |
| AudioOutDelay.Set .....                                      | 54        |
| <b>Audio Stream Selection Commands (Decoder) .....</b>       | <b>54</b> |
| AudioStreamNameList.Get .....                                | 54        |
| SelectAudioStream.Get .....                                  | 54        |
| SelectAudioStream.Set .....                                  | 55        |
| <b>AES Decryption Commands (Decoder) .....</b>               | <b>55</b> |
| AesDecryptPassword.Get .....                                 | 55        |
| AesDecryptPassword.Set .....                                 | 55        |
| AesDecryptPassword.Reset .....                               | 56        |
| <b>Network &amp; Source Control Commands (Decoder) .....</b> | <b>56</b> |

|                                                    |                              |
|----------------------------------------------------|------------------------------|
| LanEnabled.Get .....                               | 56                           |
| LanEnabled.Set .....                               | 56                           |
| SwitchVideoSource.Set .....                        | 57                           |
| Video Wall Commands (Decoder) .....                | 57                           |
| StandardVideoWallInfo.Get .....                    | 57                           |
| StandardVideoWallInfo.Set .....                    | 57                           |
| VideoWallBezel.Set .....                           | 57                           |
| VideoWallBezel.Get .....                           | 58                           |
| VideoWallDisplayPref.Set .....                     | 58                           |
| VideoWallDisplayPref.Get .....                     | 58                           |
| Events & Notifications .....                       | 60                           |
| Encoder Events (HW-LUMA-E4211) .....               | 60                           |
| Decoder Events (HW-LUMA-D4111) .....               | 60                           |
| Version History .....                              | 62                           |
| Encoder API Document History (HW-LUMA-E4211) ..... | Error! Bookmark not defined. |
| Decoder API Document History (HW-LUMA-D4111) ..... | Error! Bookmark not defined. |
| Atlona Document History .....                      | 62                           |

## Command Quick Reference

The following table lists all API commands with applicability indicators. ● (Enc) indicates the command is available on the HW-LUMA-E4211 Encoder. ● (Dec) indicates the command is available on the HW-LUMA-D4111 Decoder.

| Command                          | Enc | Dec | Description                                         |
|----------------------------------|-----|-----|-----------------------------------------------------|
| <b>Platform.FactoryReset</b>     | ●   | ●   | Restore device to factory defaults.                 |
| <b>Platform.Reboot</b>           | ●   | ●   | Restart the device.                                 |
| <b>System.Get</b>                | ●   | ●   | Return firmware version, model, and serial number.  |
| <b>SystemCurrentTime.Get</b>     | ●   | ●   | Return current system time.                         |
| <b>TimeSync.Set</b>              | ●   | ●   | Configure system time sync (NTP or manual).         |
| <b>TimeSync.Get</b>              | ●   | ●   | Return time sync settings.                          |
| <b>LoginPassword.Set</b>         | ●   | ●   | Change the HTTP login password.                     |
| <b>Verify8021X.Get</b>           | ●   | ●   | Return 802.1X authentication settings.              |
| <b>Verify8021X.Set</b>           | ●   | ●   | Configure 802.1X network authentication.            |
| <b>AuthEnabled.Get</b>           | ●   | ●   | Return WebSocket/Telnet authentication status.      |
| <b>AuthEnabled.Set</b>           | ●   | ●   | Enable/disable WebSocket and Telnet authentication. |
| <b>Auth.Set</b>                  | ●   | ●   | Set WebSocket/Telnet authentication credentials.    |
| <b>TelnetEnable.Get</b>          | ●   | ●   | Return Telnet enable status.                        |
| <b>TelnetEnable.Set</b>          | ●   | ●   | Enable or disable Telnet.                           |
| <b>SshEnable.Get</b>             | ●   | ●   | Return SSH enable status.                           |
| <b>SshEnable.Set</b>             | ●   | ●   | Enable or disable SSH.                              |
| <b>Network.Get</b>               | ●   | ●   | Return IP mode, address, subnet, gateway, DNS.      |
| <b>Network.Set</b>               | ●   | ●   | Configure network parameters.                       |
| <b>NetworkHostname.Get</b>       | ●   | ●   | Return device hostname.                             |
| <b>NetworkHostname.Set</b>       | ●   | ●   | Set device hostname.                                |
| <b>DeviceDiscovery.Get</b>       | ●   | ●   | Return network discovery broadcast status.          |
| <b>DeviceDiscovery.Set</b>       | ●   | ●   | Enable/disable network discovery broadcast.         |
| <b>FrontPanelLockButtons.Get</b> | ●   | ●   | Return front panel button lock status.              |
| <b>FrontPanelLockButtons.Set</b> | ●   | ●   | Lock or unlock front panel buttons.                 |
| <b>FrontPanelLed.Get</b>         | ●   | ●   | Return front panel LED status.                      |
| <b>FrontPanelLed.Set</b>         | ●   | ●   | Enable or disable front panel LEDs.                 |
| <b>FrontPanelBlinkLed.Set</b>    | ●   | ●   | Control front panel LED blinking.                   |
| <b>OsdText.Get</b>               | ●   | ●   | Return OSD text overlay settings.                   |
| <b>OsdText.Set</b>               | ●   | ●   | Configure OSD text overlay.                         |
| <b>OsdLogo.Get</b>               | ●   | ●   | Return OSD logo overlay status.                     |
| <b>OsdLogo.Set</b>               | ●   | ●   | Configure OSD logo overlay.                         |
| <b>ThumbnailStatus.Get</b>       | ●   | ●   | Return thumbnail preview enable status.             |
| <b>ThumbnailStatus.Set</b>       | ●   | ●   | Enable or disable thumbnail preview.                |

| Command                     | Enc | Dec | Description                                       |
|-----------------------------|-----|-----|---------------------------------------------------|
| Scrambling.Get              | •   | •   | Return HDCP scrambling mode.                      |
| Scrambling.Set              | •   | •   | Set HDCP scrambling mode.                         |
| SerialTransParam.Get        | •   | •   | Return serial port communication parameters.      |
| SerialControlParams.Get     | •   | •   | Return serial port control configuration.         |
| SerialControlParams.Reset   | •   | •   | Reset serial port parameters to factory defaults. |
| SerialControlMode.Set       | •   | •   | Set serial port control mode.                     |
| SerialControlProxy.Set      | •   | •   | Configure serial port proxy parameters.           |
| ControlSerialCommand.Set    | •   | •   | Send a serial command through the serial port.    |
| ControlSerialParam.Set      | •   | •   | Set serial control parameters for command mode.   |
| IrControlParams.Get         | •   | •   | Return IR control configuration.                  |
| IrControlParams.Reset       | •   | •   | Reset IR parameters to factory defaults.          |
| IrControlMode.Set           | •   | •   | Set IR control operating mode.                    |
| IrControlProxy.Set          | •   | •   | Configure IR proxy parameters.                    |
| ControlIrCommand.Set        | •   | •   | Send an IR command through the device IR port.    |
| AllEthernetLanStatus.Get    | •   |     | Return ethernet and Dante port status.            |
| AesEncryptEnabled.Get       | •   |     | Return AES encryption enable status.              |
| AesEncryptEnabled.Set       | •   |     | Enable/disable AES stream encryption.             |
| AesEncryptPassword.Get      | •   |     | Return AES encryption passwords.                  |
| AesEncryptPassword.Reset    | •   |     | Reset AES encryption passwords to defaults.       |
| AesEncryptPassword.Set      | •   |     | Set AES encryption passwords.                     |
| AnalogAudioInputStatus.Get  | •   |     | Return analog audio input signal status.          |
| AudioEncode.Get             | •   |     | Return audio encoding parameters.                 |
| AudioEncode.Set             | •   |     | Set audio encoding parameters.                    |
| AudioInputDownmixing.Get    | •   |     | Return audio input downmixing state.              |
| AudioInputDownmixing.Set    | •   |     | Enable/disable audio input downmixing to mono.    |
| AudioInputMute.Get          | •   |     | Return audio input mute state.                    |
| AudioInputMute.Set          | •   |     | Enable/disable audio input mute.                  |
| AudioInSelection.Get        | •   |     | Return selected audio input source.               |
| AudioInSelection.Set        | •   |     | Set audio input source (HDMI or analog).          |
| AudioInputFormat.Get        | •   |     | Return audio input format and parameters.         |
| AudioLanEnabled.Get         | •   |     | Return AudioLAN (Dante) port enable status.       |
| AudioLanEnabled.Set         | •   |     | Enable or disable AudioLAN (Dante) port.          |
| AudioMulticastStream.Get    | •   |     | Return audio multicast stream configuration.      |
| AudioMulticastStream.Set    | •   |     | Configure audio multicast stream.                 |
| AudioOutputVol.Get          | •   |     | Return analog audio output volume.                |
| AudioOutputVol.Set          | •   |     | Set analog audio output volume.                   |
| DanteInfo.Get               | •   |     | Return Dante network name, IP, and MAC.           |
| DanteStreamLanSelection.Get | •   |     | Return LAN port used for Dante stream.            |

| Command                        | Enc | Dec | Description                                     |
|--------------------------------|-----|-----|-------------------------------------------------|
| DanteStreamLanSelection.Set    | •   |     | Set LAN port for Dante audio stream.            |
| DisplayHdcpStatus.Get          | •   |     | Return HDCP capability of HDMI loop-out sink.   |
| Dscp.Get                       | •   |     | Return DSCP QoS settings.                       |
| Dscp.Set                       | •   |     | Configure DSCP QoS settings.                    |
| EdidInput.Get                  | •   |     | Return EDID setting for HDMI input.             |
| EdidInput.Set                  | •   |     | Set EDID for HDMI input.                        |
| EdidInputBuildFile.Set         | •   |     | Build and set a custom EDID file.               |
| EdidMem1File.Get               | •   |     | Return EDID data from memory slot 1.            |
| HdcpAnnouncement.Get           | •   |     | Return HDMI input HDCP version.                 |
| HdcpAnnouncement.Set           | •   |     | Set HDMI input HDCP version.                    |
| ImageDisplay.Get               | •   |     | Return basemap display policy settings.         |
| ImageDisplay.Set               | •   |     | Set basemap display policy.                     |
| ImageReset.Set                 | •   | •   | Restore image basemap to factory default.       |
| MainStreamVideoEncode.Get      | •   |     | Return main stream video encoding parameters.   |
| MainStreamVideoEncode.Set      | •   |     | Set main stream video encoding parameters.      |
| MainStreamVideoQp.Get          | •   |     | Return QP values for main video stream.         |
| MainStreamVideoQp.Set          | •   |     | Set QP values for main video stream.            |
| NetworkSpeed.Get               | •   |     | Return network port speed setting.              |
| NetworkSpeed.Set               | •   |     | Set network port speed.                         |
| PowerConsumption.Get           | •   |     | Return current device power consumption.        |
| PreviewClientCnt.Get           | •   |     | Return number of preview stream clients.        |
| PreviewStreamEnable.Get        | •   |     | Return preview stream enable status.            |
| PreviewStreamEnable.Set        | •   |     | Enable or disable preview stream.               |
| ProfileSelection.Get           | •   |     | Return current encoding profile mode.           |
| ProfileSelection.Set           | •   |     | Set encoding profile mode.                      |
| Rtmp.Get                       | •   |     | Return RTMP stream configuration.               |
| Rtmp.Set                       | •   |     | Configure RTMP streaming parameters.            |
| RtmpRunStatus.Get              | •   |     | Return RTMP streaming run status.               |
| StreamTtl.Get                  | •   |     | Return multicast stream TTL value.              |
| StreamTtl.Set                  | •   |     | Set multicast stream TTL value.                 |
| SystemFuncDantePowerStatus.Get | •   |     | Return Dante audio function state.              |
| SystemFuncDantePowerStatus.Set | •   |     | Enable/disable Dante audio function.            |
| SystemFuncPowerStatus.Get      | •   |     | Return video interlaced input function state.   |
| SystemFuncPowerStatus.Set      | •   |     | Enable/disable video interlaced input function. |
| SystemLcdBrigtness.Get         | •   |     | Return current LCD screen brightness level.     |
| SystemLcdBrigtness.Set         | •   |     | Set LCD screen brightness level.                |
| VideoInputMute.Get             | •   |     | Return video input mute state.                  |

| Command                     | Enc | Dec | Description                                       |
|-----------------------------|-----|-----|---------------------------------------------------|
| VideoInputMute.Set          | •   |     | Enable/disable video input mute.                  |
| VideoInputStatus.Get        | •   |     | Return HDMI input video status and format.        |
| VideoMultilpStream.Get      | •   |     | Return multicast video stream configuration.      |
| VideoMultilpStream.Set      | •   |     | Configure multicast video stream parameters.      |
| AesDecryptPassword.Get      |     | •   | Return AES decryption passwords.                  |
| AesDecryptPassword.Reset    |     | •   | Reset AES decryption passwords to defaults.       |
| AesDecryptPassword.Set      |     | •   | Set AES decryption passwords.                     |
| AllEthernetStatus.Get       |     | •   | Return ethernet and LAN port status.              |
| AnalogAudioOutputVol.Get    |     | •   | Return analog audio output volume.                |
| AnalogAudioOutputVol.Set    |     | •   | Set analog audio output volume.                   |
| AudioOutDelay.Get           |     | •   | Return audio output delay for lip sync.           |
| AudioOutDelay.Set           |     | •   | Set audio output delay for lip sync.              |
| AudioOutputDownmixing.Get   |     | •   | Return audio output downmixing state.             |
| AudioOutputDownmixing.Set   |     | •   | Enable/disable audio output downmixing.           |
| AudioOutputMute.Get         |     | •   | Return audio output mute state.                   |
| AudioOutputMute.Set         |     | •   | Enable or disable audio output mute.              |
| AudioStreamNameList.Get     |     | •   | Return available audio stream names.              |
| CecCommand.Get              |     | •   | Return configured CEC command codes.              |
| CecCommand.Set              |     | •   | Set a CEC command code.                           |
| CecCommandControl.Set       |     | •   | Execute a stored CEC command.                     |
| DisplayEdid.Get             |     | •   | Return EDID from connected display.               |
| LanEnabled.Get              |     | •   | Return LAN port enable status.                    |
| LanEnabled.Set              |     | •   | Enable or disable the LAN port.                   |
| LowLatencyMode.Get          |     | •   | Return low latency mode state.                    |
| LowLatencyMode.Set          |     | •   | Enable or disable low latency video mode.         |
| NoVideoSource.Get           |     | •   | Return no-video display mode setting.             |
| NoVideoSource.Set           |     | •   | Set display mode when no video source is present. |
| OutHdcpAnnouncement.Get     |     | •   | Return HDMI output HDCP setting.                  |
| OutHdcpAnnouncement.Set     |     | •   | Set HDMI output HDCP version.                     |
| OutHdcpAnnouncementTest.Set |     | •   | Set HDCP output version in test mode.             |
| SelectAudioStream.Get       |     | •   | Return current audio stream pairing.              |
| SelectAudioStream.Set       |     | •   | Set multicast audio stream to decode.             |
| SelectVideoStream.Get       |     | •   | Return current video stream pairing.              |
| SelectVideoStream.Set       |     | •   | Set multicast video stream to decode.             |
| StandardVideoWallInfo.Get   |     | •   | Return video wall configuration.                  |
| StandardVideoWallInfo.Set   |     | •   | Configure video wall parameters.                  |
| VideoWallBezel.Get          |     | •   | Return video wall bezel compensation settings.    |

| Command                         | Enc | Dec | Description                                            |
|---------------------------------|-----|-----|--------------------------------------------------------|
| VideoWallBezel.Set              |     | •   | Set video wall bezel compensation parameters.          |
| VideoWallDisplayPref.Get        |     | •   | Return image rotation angle setting.                   |
| VideoWallDisplayPref.Set        |     | •   | Set image rotation angle (0, 90, 180, or 270 degrees). |
| SwitchVideoSource.Set           |     | •   | Switch video source up or down.                        |
| VideoHDMIOut5V.Get              |     | •   | Return HDMI output 5V disabled state.                  |
| VideoHDMIOut5V.Set              |     | •   | Set HDMI output 5V power disabled state.               |
| VideoOutColorRange.Get          |     | •   | Return video output color range.                       |
| VideoOutColorRange.Set          |     | •   | Set video output color range.                          |
| VideoOutColorspace.Get          |     | •   | Return HDMI output color space.                        |
| VideoOutColorspace.Set          |     | •   | Set HDMI output color space.                           |
| VideoOutSupportedTimingList.Get |     | •   | Return list of supported output timings.               |
| VideoOutTiming.Get              |     | •   | Return current video output timing.                    |
| VideoOutTiming.Set              |     | •   | Set video output timing.                               |
| VideoOutputStatus.Get           |     | •   | Return HDMI output video status.                       |
| VideoStreamNameList.Get         |     | •   | Return available video stream names.                   |

## Overview

### About This Document

This document describes the JSON-RPC API commands available for the Atlona HW-LUMA-E4211 LumaStream Encoder and HW-LUMA-D4111 LumaStream Decoder. Commands are organized into three categories: Shared Commands (applicable to both devices), Encoder-Only Commands (HW-LUMA-E4211), and Decoder-Only Commands (HW-LUMA-D4111).

### Connection

All API commands are transmitted over a WebSocket or Telnet connection to the device. The device must be on the same network as the controlling system. Connect to the device using its IP address. The default port for WebSocket is 80; Telnet port is 23.

### JSON-RPC 2.0 Protocol

All commands use the JSON-RPC 2.0 format. Commands are sent as JSON objects and responses are returned as JSON objects.

Command request structure:

```
{ "jsonrpc": "2.0", "id": "X", "method": "Y", "params": "Z" }
```

| Command        | Parameter(s) | Description                                                                                                                           |
|----------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>jsonrpc</b> | "2.0"        | Required. Must always be "2.0".                                                                                                       |
| <b>id</b>      | string       | Optional (recommended). Message identifier used to correlate requests with replies. The device echoes the same id value in the reply. |
| <b>method</b>  | string       | Required. The API method name (e.g., "Platform.Reboot").                                                                              |
| <b>params</b>  | object       | Optional. Parameters for the command. Not all commands require params.                                                                |

Command reply structure:

```
{ "jsonrpc": "2.0", "id": "X", "result": "Z" }
```

| Command        | Parameter(s) | Description              |
|----------------|--------------|--------------------------|
| <b>jsonrpc</b> | "2.0"        | Always "2.0".            |
| <b>id</b>      | string       | Echoed from the request. |

| Command       | Parameter(s) | Description                                                                                                       |
|---------------|--------------|-------------------------------------------------------------------------------------------------------------------|
| <b>result</b> | varies       | Required. The result of the command. Format varies by command — may be a boolean, string, number, or JSON object. |

## Data Types

| Command       | Parameter(s) | Description           |
|---------------|--------------|-----------------------|
| <b>bool</b>   | true / false | Boolean value.        |
| <b>string</b> | text         | Text string.          |
| <b>int32</b>  | integer      | 4-byte integer.       |
| <b>int16</b>  | integer      | 2-byte short integer. |

## Shared Commands

The following commands apply to both the HW-LUMA-E4211 Encoder and the HW-LUMA-D4111 Decoder.

### System Commands

#### Platform.FactoryReset

| Command                      | Parameter(s) | Description                                                                                                                                                                        |
|------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Platform.FactoryReset</b> | (none)       | Restores the device to factory default settings. <ul style="list-style-type: none"><li>• <b>result:</b> [true, false] — true indicates the factory reset was successful.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "Platform.FactoryReset"}
```

Reply:

```
{"id": "User1", "result": "true", "jsonrpc": "2.0"}
```

#### Platform.Reboot

| Command                | Parameter(s) | Description                                                                                                                                                         |
|------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Platform.Reboot</b> | (none)       | Restarts the device. <ul style="list-style-type: none"><li>• <b>result:</b> [true, false] — true indicates the restart command was executed successfully.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "Platform.Reboot"}
```

Reply:

```
{"id": "User1", "result": "true", "jsonrpc": "2.0"}
```

#### System.Get

| Command           | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>System.Get</b> | (none)       | Returns device system information including firmware versions, model name, and serial number. <ul style="list-style-type: none"><li>• <b>FwVer:</b> Software package version (e.g., v1.0.0).</li><li>• <b>MCUVer:</b> MCU firmware version.</li><li>• <b>ARMVer:</b> ARM firmware version.</li><li>• <b>Model:</b> Device model name (e.g., AT-LUMA-E4211).</li><li>• <b>SN:</b> Device serial number.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "System.Get"}
```

#### SystemCurrentTime.Get

| Command                      | Parameter(s) | Description                                                                                                                                                           |
|------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemCurrentTime.Get</b> | (none)       | Returns the current system time including time zone. <ul style="list-style-type: none"> <li>• <b>datetime:</b> Current time in format YYYY-MM-DD hh:mm:ss.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "SystemCurrentTime.Get"}
```

## TimeSync.Set

| Command             | Parameter(s)                                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|-----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TimeSync.Set</b> | synctype,<br>ntp{ntpserver, interval,<br>timezone},<br>datetime{date, time} | Configures the system time synchronization method. <ul style="list-style-type: none"> <li>• <b>synctype:</b> [ntp, manual] — ntp uses network time server; manual sets time directly.</li> <li>• <b>ntpserver:</b> NTP server URL (e.g., pool.ntp.org).</li> <li>• <b>interval:</b> [1–120] Sync interval in minutes.</li> <li>• <b>timezone:</b> Time zone string (e.g., UTC).</li> <li>• <b>date:</b> Date in YYYY-MM-DD format. Used when synctype is manual.</li> <li>• <b>time:</b> Time in hh:mm:ss format. Used when synctype is manual.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "TimeSync.Set", "params": {"synctype": "ntp", "ntp": {"ntpserver": "pool.ntp.org", "interval": 2, "timezone": "UTC"}}
```

## TimeSync.Get

| Command             | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TimeSync.Get</b> | (none)       | Returns the current time synchronization settings including NTP server, interval, time zone, and full timezone list. <ul style="list-style-type: none"> <li>• <b>synctype:</b> [ntp, manual].</li> <li>• <b>ntpserver:</b> NTP server URL. Default: pool.ntp.org.</li> <li>• <b>interval:</b> Sync interval in minutes. Default: 5.</li> <li>• <b>timezone:</b> Current time zone. Default: UTC.</li> <li>• <b>timezonelist:</b> JSON array of all available time zones.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "TimeSync.Get"}
```

## DeviceDiscovery.Get

| Command                    | Parameter(s) | Description                                                                  |
|----------------------------|--------------|------------------------------------------------------------------------------|
| <b>DeviceDiscovery.Get</b> | (none)       | Returns the device network discovery function status and broadcast interval. |

| Command | Parameter(s) | Description                                                                                                                                                                                              |
|---------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |              | <ul style="list-style-type: none"> <li>• <b>discoveryenable</b>: [true, false] — true means discovery is active.</li> <li>• <b>interval</b>: [2–300] Broadcast period in seconds. Default: 2.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "DeviceDiscovery.Get"}
```

## DeviceDiscovery.Set

| Command                    | Parameter(s)                 | Description                                                                                                                                                                                          |
|----------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DeviceDiscovery.Set</b> | discoveryenable,<br>interval | Enables or disables the device network discovery broadcast function. <ul style="list-style-type: none"> <li>• <b>discoveryenable</b>: [true, false].</li> <li>• <b>interval</b>: [2–300].</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "DeviceDiscovery.Set", "params": {"discoveryenable": false, "interval": 2}}
```

**Reply:**

```
{"id": "User1", "result": {"discoveryenable": false, "interval": 2}, "jsonrpc": "2.0"}
```

## Network Commands

### Network.Get

| Command            | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Network.Get</b> | (none)       | Returns the current network configuration including IP mode, address, subnet, gateway, MAC, and DNS settings. <ul style="list-style-type: none"> <li>• <b>ip_mode</b>: [static, dhcp].</li> <li>• <b>ipaddr</b>: IP address (e.g., 192.168.1.100).</li> <li>• <b>netmask</b>: Subnet mask (e.g., 255.255.255.0).</li> <li>• <b>gateway</b>: Gateway address.</li> <li>• <b>mac</b>: MAC address.</li> <li>• <b>dns_mode</b>: [manual, auto].</li> <li>• <b>dns_preferred</b>: Primary DNS server.</li> <li>• <b>dns_alternative</b>: Secondary DNS server.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "Network.Get"}
```

### Network.Set

| Command            | Parameter(s)                                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Network.Set</b> | ip_mode, ipaddr, netmask, gateway, dns{dns_mode, dns_preferred, dns_alternative} | Configures the device network parameters. <ul style="list-style-type: none"> <li>• <b>ip_mode</b>: [static, dhcp] — Required.</li> <li>• <b>ipaddr</b>: IP address. Required when ip_mode is static.</li> <li>• <b>netmask</b>: Subnet mask. Required when ip_mode is static.</li> <li>• <b>gateway</b>: Gateway. Required when ip_mode is static.</li> <li>• <b>dns_mode</b>: [manual, auto] — Required.</li> <li>• <b>dns_preferred</b>: Primary DNS. Required when dns_mode is manual.</li> <li>• <b>dns_alternative</b>: Secondary DNS. Required when dns_mode is manual.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "Network.Set", "params": { "ip_mode": "T", "ipaddr": "X.X.X.X", "netmask": "Y.Y.Y.Y", "gateway": "Z.Z.Z.Z", "dns": { "dns_mode": "U", "dns_preferred": "W.W.W.W", "dns_alternative": "V.V.V.V" } } }
```

## NetworkHostname.Get

| Command                    | Parameter(s) | Description                                                                                                                               |
|----------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NetworkHostname.Get</b> | (none)       | Returns the current device hostname. <ul style="list-style-type: none"> <li>• <b>hostname</b>: Current device hostname string.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "NetworkHostname.Get" }
```

**Reply:**

```
{ "id": "User1", "result": { "hostname": "ATL" }, "jsonrpc": "2.0" }
```

## NetworkHostname.Set

| Command                    | Parameter(s) | Description                                                                                                                                                               |
|----------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NetworkHostname.Set</b> | hostname     | Sets the device hostname. <ul style="list-style-type: none"> <li>• <b>hostname</b>: New hostname. Must be 1–63 characters; letters, numbers, and hyphens only.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "NetworkHostname.Set", "params": { "hostname": "ATL" } }
```

**Reply:**

```
{ "id": "X", "result": { "hostname": "ATL" }, "jsonrpc": "2.0" }
```

## Security Commands

### LoginPassword.Set

| Command                  | Parameter(s)                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LoginPassword.Set</b> | communication,<br>oldpassword,<br>newpassword, confirm | <p>Changes the login password for the specified communication method.</p> <ul style="list-style-type: none"> <li>• <b>communication:</b> [http] — specifies which link password to change.</li> <li>• <b>oldpassword:</b> Current password. Length 6–16 characters; letters or numbers only.</li> <li>• <b>newpassword:</b> New password. Length 6–16 characters; letters or numbers only.</li> <li>• <b>confirm:</b> Password confirmation. Must match newpassword.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "LoginPassword.Set", "params": { "communication": "http", "oldpassword": "admin", "newpassword": "admin1", "confirm": "admin1" } }
```

**Reply:**

```
{ "id": "User1", "result": { "communication": "http", "oldpassword": "admin", "newpassword": "admin1", "confirm": "admin1" }, "jsonrpc": "2.0" }
```

## Verify8021X.Get

| Command                | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Verify8021X.Get</b> | (none)       | <p>Returns the current 802.1X authentication settings.</p> <ul style="list-style-type: none"> <li>• <b>8021xenable:</b> [true, false] — whether 802.1X is enabled.</li> <li>• <b>verifymode:</b> [0, 1] — 0: EAP-MSCHAPV2; 1: EAP-TLS.</li> <li>• <b>mschapv2user:</b> Username for EAP-MSCHAPV2 mode.</li> <li>• <b>mschapv2password:</b> Password for EAP-MSCHAPV2 mode.</li> <li>• <b>tlskeypassword:</b> Cipher key for EAP-TLS mode.</li> <li>• <b>certificate:</b> [true, false] — true indicates a certificate upload is required.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "Verify8021X.Get" }
```

## Verify8021X.Set

| Command                | Parameter(s)                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Verify8021X.Set</b> | 8021xen, verifymode,<br>mschapv2user,<br>mschapv2password,<br>tlskeypassword,<br>certificate | <p>Configures 802.1X network authentication parameters.</p> <ul style="list-style-type: none"> <li>• <b>8021xen:</b> [true, false] — enable or disable 802.1X. Required.</li> <li>• <b>verifymode:</b> [0, 1] — 0: EAP-MSCHAPV2; 1: EAP-TLS. Required.</li> <li>• <b>mschapv2password:</b> Required when verifymode is 0 (EAP-MSCHAPV2).</li> <li>• <b>tlskeypassword:</b> Required when verifymode is 1 (EAP-TLS).</li> </ul> |

| Command | Parameter(s) | Description                                                                                       |
|---------|--------------|---------------------------------------------------------------------------------------------------|
|         |              | <ul style="list-style-type: none"> <li>• <b>certificate:</b> [true, false] — Required.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "Verify8021XSetResults", "method": "Verify8021X.Set", "params": { "8021xen": true, "verifymode": 0, "certificate": false, "mschapv2user": "steve", "mschapv2password": "testing" } }
```

**Reply:**

```
{ "id": "Verify8021XSetResults", "jsonrpc": "2.0", "result": { "8021xen": true, "certificate": false, "mschapv2password": "testing", "mschapv2user": "steve", "verifymode": 0 } }
```

## AuthEnabled.Set

| Command                | Parameter(s)                   | Description                                                                                                                                                                                                                                                                       |
|------------------------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AuthEnabled.Set</b> | websocket_auth,<br>telnet_auth | Enables or disables authentication for WebSocket and Telnet connections. <ul style="list-style-type: none"> <li>• <b>websocket_auth:</b> [true, false] — enable WebSocket authentication.</li> <li>• <b>telnet_auth:</b> [true, false] — enable Telnet authentication.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AuthEnabled.Set", "params": { "websocket_auth": true, "telnet_auth": false } }
```

## AuthEnabled.Get

| Command                | Parameter(s) | Description                                                                            |
|------------------------|--------------|----------------------------------------------------------------------------------------|
| <b>AuthEnabled.Get</b> | (none)       | Returns the current authentication enable status for WebSocket and Telnet connections. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AuthEnabled.Get" }
```

## Auth.Set

| Command         | Parameter(s)       | Description                                                                                                                                                                                                                 |
|-----------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Auth.Set</b> | username, password | Sets the authentication credentials used when authentication is enabled. <ul style="list-style-type: none"> <li>• <b>username:</b> Authentication username.</li> <li>• <b>password:</b> Authentication password.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "Auth.Set", "params": { "set_password": "123456" } }
```

## TelnetEnable.Get

| Command                 | Parameter(s) | Description                                        |
|-------------------------|--------------|----------------------------------------------------|
| <b>TelnetEnable.Get</b> | (none)       | Returns the current Telnet protocol enable status. |

| Command | Parameter(s) | Description                                                                                                    |
|---------|--------------|----------------------------------------------------------------------------------------------------------------|
|         |              | <ul style="list-style-type: none"> <li><b>enable:</b> [true, false] — true means Telnet is enabled.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "TelnetEnable.Get"}
```

## TelnetEnable.Set

| Command                 | Parameter(s) | Description                                                                                                              |
|-------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>TelnetEnable.Set</b> | enable       | Enables or disables the Telnet protocol. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "TelnetEnable.Set", "params": {"telnet_enable": true}}
```

## SshEnable.Get

| Command              | Parameter(s) | Description                                                                                                                                                 |
|----------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SshEnable.Get</b> | (none)       | Returns the current SSH protocol enable status. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false] — true means SSH is enabled.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "SshEnable.Get"}
```

## SshEnable.Set

| Command              | Parameter(s) | Description                                                                                                           |
|----------------------|--------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>SshEnable.Set</b> | enable       | Enables or disables the SSH protocol. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "SshEnable.Set", "params": {"ssh_enable": true}}
```

## Front Panel Commands

### FrontPanelLockButtons.Set

| Command                          | Parameter(s) | Description                                                                                                                                                                  |
|----------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FrontPanelLockButtons.Set</b> | lock         | Enables or disables the front panel button lock function. <ul style="list-style-type: none"> <li><b>lock:</b> [true, false] — true locks the front panel buttons.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "FrontPanelLockButtons.Set", "params": {"lock": true}}
```

**Reply:**

```
{"id": "User1", "result": {"lock": true}, "jsonrpc": "2.0"}
```

## FrontPanelLockButtons.Get

| Command                          | Parameter(s) | Description                                                                                                                       |
|----------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>FrontPanelLockButtons.Get</b> | (none)       | Returns the current front panel button lock status. <ul style="list-style-type: none"> <li><b>lock:</b> [true, false].</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "FrontPanelLockButtons.Get"}
```

## FrontPanelLed.Set

| Command                  | Parameter(s) | Description                                                                                                                                             |
|--------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FrontPanelLed.Set</b> | enable       | Enables or disables the front panel LED lights. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false] — true enables the LEDs.</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "FrontPanelLed.Set", "params": {"led": true}}
```

### Reply:

```
{"id": "User1", "result": {"led": true}, "jsonrpc": "2.0"}
```

## FrontPanelLed.Get

| Command                  | Parameter(s) | Description                                        |
|--------------------------|--------------|----------------------------------------------------|
| <b>FrontPanelLed.Get</b> | (none)       | Returns the current front panel LED enable status. |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "FrontPanelLed.Get"}
```

## FrontPanelBlinkLed.Set

| Command                       | Parameter(s)      | Description                                                                                                                                                                                                        |
|-------------------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FrontPanelBlinkLed.Set</b> | enable, blinkrate | Controls the front panel LED blinking function. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false] — true enables LED blinking.</li> <li><b>blinkrate:</b> Blink rate in milliseconds.</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "FrontPanelBlinkLed.Set"}
```

### Reply:

```
{"id": "User1", "result": {"blinkled": true}, "jsonrpc": "2.0"}
```

## OSD Commands

### OsdText.Get

| Command            | Parameter(s) | Description                                                                                                                                                                                                                                                      |
|--------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OsdText.Get</b> | (none)       | Returns the current OSD text overlay settings. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false] — whether OSD text is displayed.</li> <li><b>text:</b> The OSD text string.</li> <li><b>position:</b> OSD text position on screen.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"OsdText.Get","params":{"osdindex":"O"}}
```

## OsdText.Set

| Command            | Parameter(s)           | Description                                                                                                                                                                                                                   |
|--------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OsdText.Set</b> | enable, text, position | Configures the OSD text overlay. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> <li><b>text:</b> Text to display on screen.</li> <li><b>position:</b> Screen position for the OSD text.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"OsdText.Set","params":{"osdindex":0,"osdtextenabled":true,"displaytext":{"content":"text","fontcolor":"255.255.255","backcolor":"255.255.255","fonttransparency":255,"backtransparency":1,"startpostion":{"startx":10,"starty":1},"fontsize":{"fonth":1},"displayscolleffects":{"iterations":0,"speed":0,"direction":0,"enable":true}}}}
```

**Reply:**

```
{"id":"X","result":{"osdindex":0,"osdtextenabled":true,"displaytext":{"content":"text","fontcolor":"255.255.255","backcolor":"255.255.255","fonttransparency":255,"backtransparency":1,"startpostion":{"startx":10,"starty":1},"fontsize":{"fonth":1},"displayscolleffects":{"iterations":0,"speed":0,"direction":0,"enable":true}}},"jsonrpc":"2.0"}
```

## OsdLogo.Get

| Command            | Parameter(s) | Description                                                      |
|--------------------|--------------|------------------------------------------------------------------|
| <b>OsdLogo.Get</b> | (none)       | Returns the current OSD logo overlay enable status and position. |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"OsdLogo.Get"}
```

**Reply:**

```
{"id":"User1","result":{"osdlogoenabled":true,"displaylogo":{"backtransparency":"128","startpostion":{"startx":"10","starty":"10"},"logohight":"20"}},"jsonrpc":"2.0"}
```

## OsdLogo.Set

| Command            | Parameter(s)     | Description                                                                                                                                                    |
|--------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OsdLogo.Set</b> | enable, position | Configures the OSD logo overlay. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> <li><b>position:</b> Position on screen.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"OsdLogo.Set","params":{"osdlogoenabled":true,"displaylogo":{"backtransparency":128,"startpostion":{"startx":10,"starty":10},"logohight":20}}}
```

**Reply:**

```
{
  "id": "User1",
  "result": {
    "osdlogoenabled": true,
    "displaylogo": {
      "backtransparency": 128,
      "startposition": {
        "startx": 10,
        "starty": 10
      },
      "logohight": 20
    }
  },
  "jsonrpc": "2.0"
}
```

## Thumbnail Commands

### ThumbnailStatus.Get

| Command             | Parameter(s) | Description                                                                                                                                            |
|---------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| ThumbnailStatus.Get | (none)       | Returns the current thumbnail (preview stream) function enable status. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

#### Request:

```
{
  "jsonrpc": "2.0",
  "id": "X",
  "method": "ThumbnailStatus.Get"
}
```

#### Reply:

```
{
  "id": "X",
  "result": {
    "thumbnailenable": "Y"
  },
  "jsonrpc": "2.0"
}
```

### ThumbnailStatus.Set

| Command             | Parameter(s) | Description                                                                                                                                  |
|---------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| ThumbnailStatus.Set | enable       | Enables or disables the thumbnail (preview stream) function. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

#### Request:

```
{
  "jsonrpc": "2.0",
  "id": "User1",
  "method": "ThumbnailStatus.Set",
  "params": {
    "thumbnailenable": true
  }
}
```

#### Reply:

```
{
  "id": "User1",
  "result": {
    "thumbnailenable": true
  },
  "jsonrpc": "2.0"
}
```

## HDCP Scrambling Commands

### Scrambling.Set

| Command        | Parameter(s) | Description                                                                                                                                                                    |
|----------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scrambling.Set | mode         | Sets the HDCP scrambling mode for the device. <ul style="list-style-type: none"> <li><b>mode:</b> [normal, open, prohibitive] — normal is default.</li> </ul> Default: normal. |

#### Request:

```
{
  "jsonrpc": "2.0",
  "id": "X",
  "method": "Scrambling.Set",
  "params": {
    "mode": "M"
  }
}
```

#### Reply:

```
{
  "id": "X",
  "jsonrpc": "2.0",
  "result": {
    "mode": "M"
  }
}
```

## Scrambling.Get

| Command               | Parameter(s) | Description                                                                                                                           |
|-----------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Scrambling.Get</b> | (none)       | Returns the current HDCP scrambling mode. <ul style="list-style-type: none"> <li><b>mode:</b> [normal, open, prohibitive].</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "Scrambling.Get"}
```

### Reply:

```
{"id": "X", "jsonrpc": "2.0", "result": {"mode": "M"}}
```

## Network Port Status Commands

**NOTE:** The encoder (HW-LUMA-E4211) uses AllEthernetLanStatus.Get and reports ethernet and audiolan ports. The decoder (HW-LUMA-D4111) uses AllEthernetStatus.Get and reports ethernet and lan ports.

## AllEthernetLanStatus.Get

| Command                         | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AllEthernetLanStatus.Get</b> | (none)       | Encoder only. Returns status information for the ethernet and audiolan (Dante) network ports. <ul style="list-style-type: none"> <li><b>lanid:</b> [ethernet, audiolan] — ethernet is the main port; audiolan is used for Dante audio.</li> <li><b>link:</b> [true, false] — connection status.</li> <li><b>speed:</b> Network port speed (e.g., 1000M).</li> <li><b>mac:</b> MAC address.</li> <li><b>lldp:</b> LLDP information string.</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "AllEthernetLanStatus.Get"}
```

## AllEthernetStatus.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AllEthernetStatus.Get</b> | (none)       | Decoder only. Returns status information for the ethernet and lan network ports. <ul style="list-style-type: none"> <li><b>lanid:</b> [ethernet, lan] — ethernet is the main port; lan is the secondary serial network port.</li> <li><b>link:</b> [true, false] — connection status.</li> <li><b>speed:</b> Network port speed (e.g., 1000M).</li> <li><b>mac:</b> MAC address.</li> <li><b>lldp:</b> LLDP information string.</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "AllEthernetStatus.Get"}
```

## Serial Port Commands

### SerialTransParam.Get

| Command                     | Parameter(s) | Description                                                                                         |
|-----------------------------|--------------|-----------------------------------------------------------------------------------------------------|
| <b>SerialTransParam.Get</b> | (none)       | Returns the current serial port communication parameters (baud rate, data bits, stop bits, parity). |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "SerialTransParam.Get"}
```

### SerialControlParams.Get

| Command                        | Parameter(s) | Description                                                       |
|--------------------------------|--------------|-------------------------------------------------------------------|
| <b>SerialControlParams.Get</b> | (none)       | Returns the current serial port control configuration parameters. |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "SerialControlParams.Get"}
```

### SerialControlParams.Reset

| Command                          | Parameter(s) | Description                                                      |
|----------------------------------|--------------|------------------------------------------------------------------|
| <b>SerialControlParams.Reset</b> | (none)       | Restores the serial port control parameters to factory defaults. |

### SerialControlMode.Set

| Command                      | Parameter(s) | Description                                                                                                                                                            |
|------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SerialControlMode.Set</b> | mode         | Sets the serial port control mode. <ul style="list-style-type: none"> <li><b>mode:</b> [pass-through, proxy, command] — defines how serial data is handled.</li> </ul> |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "SerialControlMode.Set", "params": {"commandmode": true, "proxymode": true}}
```

Reply:

```
{"id": "X", "jsonrpc": "2.0", "result": {"commandmode": true, "proxymode": true}}
```

### SerialControlProxy.Set

| Command                       | Parameter(s) | Description                                                                                                    |
|-------------------------------|--------------|----------------------------------------------------------------------------------------------------------------|
| <b>SerialControlProxy.Set</b> | proxyparams  | Configures the serial port proxy mode parameters, defining how serial commands are forwarded over the network. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "SerialControlProxy.Set", "params": { "proxyport": 10000 } }
```

**Reply:**

```
{ "id": "User1", "result": { "proxyport": 10000 }, "jsonrpc": "2.0" }
```

## ControlSerialCommand.Set

| Command                         | Parameter(s) | Description                                                                              |
|---------------------------------|--------------|------------------------------------------------------------------------------------------|
| <b>ControlSerialCommand.Set</b> | command      | Sends a serial control command through the device's serial port in command-control mode. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "ControlSerialCommand.Set", "params": { "option": "add", "commandparam": [ { "commandid": "1", "commandtype": "vol+", "commandname": "VOL+", "commandvalue": "00000", "valuehex": false }, { "commandid": "2", "commandtype": "custom", "commandname": "UP", "commandvalue": "111111", "valuehex": false } ] } }
```

## ControlSerialParam.Set

| Command                       | Parameter(s) | Description                                                      |
|-------------------------------|--------------|------------------------------------------------------------------|
| <b>ControlSerialParam.Set</b> | params       | Sets the serial control parameters used in command-control mode. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "ControlSerialParam.Set", "params": { "baudrate": 15200, "parity": "none", "databit": 8, "stopbit": 1 } }
```

**Reply:**

```
{ "id": "User1", "result": { "baudrate": 115200, "parity": "none", "databit": 8, "stopbit": 1 }, "jsonrpc": "2.0" }
```

## IR Control Commands

### IrControlParams.Get

| Command                    | Parameter(s) | Description                                                         |
|----------------------------|--------------|---------------------------------------------------------------------|
| <b>IrControlParams.Get</b> | (none)       | Returns the current infrared (IR) control configuration parameters. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "IrControlParams.Get" }
```

### IrControlParams.Reset

| Command                      | Parameter(s) | Description                                             |
|------------------------------|--------------|---------------------------------------------------------|
| <b>IrControlParams.Reset</b> | (none)       | Restores the IR control parameters to factory defaults. |

### IrControlMode.Set

| Command                  | Parameter(s) | Description                                                                                                                                                             |
|--------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>IrControlMode.Set</b> | mode         | Sets the IR control operating mode. <ul style="list-style-type: none"> <li><b>mode:</b> [pass-through, proxy, command] — defines how IR signals are handled.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "IrControlMode.Set", "params": { "commandmode": true, "proxymode": true } }
```

**Reply:**

```
{ "id": "X", "jsonrpc": "2.0", "result": { "commandmode": true, "proxymode": true } }
```

## IrControlProxy.Set

| Command                   | Parameter(s) | Description                                                                                       |
|---------------------------|--------------|---------------------------------------------------------------------------------------------------|
| <b>IrControlProxy.Set</b> | proxyparams  | Configures the IR proxy mode parameters, defining how IR commands are forwarded over the network. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "IrControlProxy.Set", "params": { "proxyport": "P" } }
```

**Reply:**

```
{ "id": "X", "result": { "proxyport": "P" }, "jsonrpc": "2.0" }
```

## ControlIrCommand.Set

| Command                     | Parameter(s) | Description                                                               |
|-----------------------------|--------------|---------------------------------------------------------------------------|
| <b>ControlIrCommand.Set</b> | command      | Sends an IR command through the device's IR port in command-control mode. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "ControlIrCommand.Set", "params": { "option": "add", "commandparam": [ { "commandid": "1", "commandtype": "vol+", "commandname": "VOL+", "commandvalue": "0000 0073 0000 001e", "valueformat": 1 } ] } }
```

## Encoder Commands (HW-LUMA-E4211)

The following commands apply exclusively to the HW-LUMA-E4211 LumaStream Encoder.

### System Commands (Encoder)

#### SystemFuncPowerStatus.Set

| Command                          | Parameter(s)    | Description                                                                                                                                                                                                 |
|----------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemFuncPowerStatus.Set</b> | videointerlaced | Sets the state of the video interlaced input function. <ul style="list-style-type: none"><li>• <b>videointerlaced</b>: [true, false] — true enables interlaced input; false places it in standby.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "SystemFuncPowerStatus.Set", "params": {"videointerlaced": "Y"}}
```

#### SystemFuncPowerStatus.Get

| Command                          | Parameter(s) | Description                                                                                                                                                |
|----------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemFuncPowerStatus.Get</b> | (none)       | Returns the current state of the video interlaced input function. <ul style="list-style-type: none"><li>• <b>videointerlaced</b>: [true, false].</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "SystemFuncPowerStatus.Get"}
```

#### SystemFuncDantePowerStatus.Set

| Command                               | Parameter(s) | Description                                                                                                                                                                                 |
|---------------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemFuncDantePowerStatus.Set</b> | dante        | Enables or disables the Dante audio function on the device. <ul style="list-style-type: none"><li>• <b>dante</b>: [true, false] — true enables Dante; false places it in standby.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "SystemFuncDantePowerStatus.Set", "params": {"dante": true}}
```

Reply:

```
{"id": "User1", "result": {"dante": true}, "jsonrpc": "2.0"}
```

#### SystemFuncDantePowerStatus.Get

| Command                               | Parameter(s) | Description                                                                                                                                  |
|---------------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemFuncDantePowerStatus.Get</b> | (none)       | Returns the current state of the device Dante audio function. <ul style="list-style-type: none"><li>• <b>dante</b>: [true, false].</li></ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "SystemFuncDantePowerStatus.Get"}
```

**Reply:**

```
{"id": "X", "result": {"dante": "Z"}, "jsonrpc": "2.0"}
```

## PowerConsumption.Get

| Command                     | Parameter(s) | Description                                                                                                                                                                      |
|-----------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PowerConsumption.Get</b> | (none)       | Returns the current overall power consumption of the device. <ul style="list-style-type: none"> <li><b>powerconsumption:</b> Power value as a string (e.g., "12.6W").</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "PowerConsumption.Get"}
```

**Reply:**

```
{"id": "X", "result": {"powerconsumption": "Y"}, "jsonrpc": "2.0"}
```

## SystemLcdBrigtness.Set

| Command                       | Parameter(s)  | Description                                                                                                                                                                                       |
|-------------------------------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemLcdBrigtness.Set</b> | lcdbrightness | Adjusts the LCD screen brightness level on the encoder. <ul style="list-style-type: none"> <li><b>lcdbrightness:</b> [0 to 1] — brightness level. 0 = minimum, 1 = maximum brightness.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "SystemLcdBrigtness.Set", "params": {"lcdbrightness": 1}}
```

**Reply:**

```
{"id": "User1", "result": {"lcdbrightness": 1}, "jsonrpc": "2.0"}
```

## SystemLcdBrigtness.Get

| Command                       | Parameter(s) | Description                                                                                                                                                   |
|-------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SystemLcdBrigtness.Get</b> | (none)       | Returns the current LCD screen brightness level. <ul style="list-style-type: none"> <li><b>lcdbrightness:</b> [0 to 1] — current brightness level.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "SystemLcdBrigtness.Get"}
```

**Reply:**

```
{"id": "User1", "result": {"lcdbrightness": 1}, "jsonrpc": "2.0"}
```

## DanteInfo.Get

| Command              | Parameter(s) | Description                                                                                                                                                                                                       |
|----------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DanteInfo.Get</b> | (none)       | Returns Dante network information for the device. <ul style="list-style-type: none"> <li><b>name:</b> Dante device name.</li> <li><b>ip:</b> Dante IP address.</li> <li><b>mac:</b> Dante MAC address.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "DanteInfo.Get"}
```

**Reply:**

```
{"id": "User1", "result": {"name": "DNT-TEST", "ip": "192.168.1.123", "mac": "AA:BB:CC:DD:EE:FF"}, "jsonrpc": "2.0"}
```

## NetworkSpeed.Get

| Command                 | Parameter(s) | Description                                                                                                                                        |
|-------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NetworkSpeed.Get</b> | (none)       | Returns the current network port speed setting. <ul style="list-style-type: none"><li>• <b>speed:</b> Network speed (e.g., 100M, 1000M).</li></ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "NetworkSpeed.Get"}
```

**Reply:**

```
{"id": "User1", "result": {"netspeed": "1000M"}, "jsonrpc": "2.0"}
```

## NetworkSpeed.Set

| Command                 | Parameter(s) | Description                                                                                                 |
|-------------------------|--------------|-------------------------------------------------------------------------------------------------------------|
| <b>NetworkSpeed.Set</b> | speed        | Sets the network port speed. <ul style="list-style-type: none"><li>• <b>speed:</b> [100M, 1000M].</li></ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "NetworkSpeed.Set", "params": {"netspeed": "1000M"}}
```

**Reply:**

```
{"id": "User1", "result": {"netspeed": "1000M"}, "jsonrpc": "2.0"}
```

## Dscp.Get

| Command         | Parameter(s) | Description                                                                                     |
|-----------------|--------------|-------------------------------------------------------------------------------------------------|
| <b>Dscp.Get</b> | (none)       | Returns the current DSCP (Differentiated Services Code Point) settings for the network traffic. |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "Dscp.Get"}
```

## Dscp.Set

| Command         | Parameter(s) | Description                                                                         |
|-----------------|--------------|-------------------------------------------------------------------------------------|
| <b>Dscp.Set</b> | dscp params  | Configures the DSCP quality-of-service settings for network traffic prioritization. |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "Dscp.Set", "params": {"video": "S", "audio": "W"}}
```

**Reply:**

```
{"id": "X", "result": {"video": "S", "audio": "W"}, "jsonrpc": "2.0"}
```

## HDCP & EDID Commands (Encoder)

### HdcpAnnouncement.Get

| Command                     | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HdcpAnnouncement.Get</b> | (none)       | Returns the HDCP compatibility version of the HDMI input. The HW-LUMA-E4211 supports one HDMI input (in1). <ul style="list-style-type: none"> <li><b>source:</b> [in1] — HDMI input channel number.</li> <li><b>hdcpversion:</b> [NONE, HDCP1.4, HDCP2.2] — NONE means HDCP is not supported.</li> </ul> <b>Note:</b> When HDCP compatibility is not supported (hdcpannouncement = false), no hdcpversion is returned. |

#### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "HdcpAnnouncement.Get"}
```

### HdcpAnnouncement.Set

| Command                     | Parameter(s)        | Description                                                                                                                                                                            |
|-----------------------------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HdcpAnnouncement.Set</b> | source, hdcpversion | Sets the HDCP compatibility version for the HDMI input. <ul style="list-style-type: none"> <li><b>source:</b> [in1].</li> <li><b>hdcpversion:</b> [NONE, HDCP1.4, HDCP2.2].</li> </ul> |

#### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "HdcpAnnouncement.Set", "params": {"source": "in1", "hdcpversion": "HDCP1.4"}}
```

#### Reply:

```
{"id": "User1", "result": {"source": "in1", "hdcpversion": "HDCP1.4"}, "jsonrpc": "2.0"}
```

### DisplayHdcpStatus.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                         |
|------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DisplayHdcpStatus.Get</b> | (none)       | Returns the HDCP support capability of the HDMI loop-out output sink. <ul style="list-style-type: none"> <li><b>hdmiout:</b> [out1] — HDMI output channel number.</li> <li><b>hdcpversion:</b> [NONE, HDCP1.4, HDCP2.2].</li> </ul> |

#### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "DisplayHdcpStatus.Get"}
```

### EdidInput.Get

| Command              | Parameter(s) | Description                                          |
|----------------------|--------------|------------------------------------------------------|
| <b>EdidInput.Get</b> | (none)       | Returns the current EDID setting for the HDMI input. |

#### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "EdidInput.Get"}
```

#### Reply:

```
{"id": "X", "result": {"source": "Y", "edidmode": "Z"}, "jsonrpc": "2.0"}
```

## EdidInput.Set

| Command              | Parameter(s) | Description                                                                                          |
|----------------------|--------------|------------------------------------------------------------------------------------------------------|
| <b>EdidInput.Set</b> | edid params  | Sets the EDID for the HDMI input, controlling the video/audio capabilities advertised to the source. |

### Request:

```
{ "jsonrpc": "2.0", "id": "User1", "method": "EdidInput.Set", "params": { "source": "hdmiin", "edidmode": 3 } }
```

## EdidInputBuildFile.Set

| Command                       | Parameter(s)     | Description                                            |
|-------------------------------|------------------|--------------------------------------------------------|
| <b>EdidInputBuildFile.Set</b> | edid file params | Builds and sets a custom EDID file for the HDMI input. |

### Request:

```
{ "jsonrpc": "2.0", "id": "X", "method": "EdidInputBuildFile.Set", "params": { "buildmethod": "B", "edid": "Z", "edidname": "D" } }
```

## EdidMem1File.Get

| Command                 | Parameter(s) | Description                                    |
|-------------------------|--------------|------------------------------------------------|
| <b>EdidMem1File.Get</b> | (none)       | Returns the EDID data stored in memory slot 1. |

### Request:

```
{ "jsonrpc": "2.0", "id": "User1", "method": "EdidMem1File.Get" }
```

### Reply:

```
{ "id": "User1", "result": { "edid": "00FFFFFFXX", "edidname": "2K_48khz" }, "jsonrpc": "2.0" }
```

## Video Commands (Encoder)

### VideoInputStatus.Get

| Command                     | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoInputStatus.Get</b> | (none)       | <p>Returns current HDMI input video information including connection state, signal validity, format, and HDCP type.</p> <ul style="list-style-type: none"> <li>• <b>active:</b> [yes, no] — yes means HDMI input is connected.</li> <li>• <b>signal:</b> [valid, invalid] — indicates whether the signal is valid.</li> <li>• <b>videoformat:</b> Format string: timing;color;bitdepth;scan (e.g., 1920x1080,60;rgb;8bit;progressive).</li> <li>• <b>hdcpl:</b> [NONE, HDCP1.4, HDCP2.2].</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"VideoInputStatus.Get"}
```

**Reply:**

```
{"id":"User1","result":{"active":"yes","signal":"vaid","videoformat":"1920x1080,60;rgb;8bit;progressive","hdcp":"NONE"},"jsonrpc":"2.0"}
```

## VideoInputMute.Get

| Command                   | Parameter(s) | Description                                                                                                                                                                                                                                        |
|---------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoInputMute.Get</b> | (none)       | Returns the current video input mute state (whether encoding of the HDMI input video is blocked). <ul style="list-style-type: none"> <li><b>videomute:</b> [true, false] — true means the input video is muted (blocked from encoding).</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoInputMute.Get"}
```

**Reply:**

```
{"id":"X","result":{"videomute":"Y"},"jsonrpc":"2.0"}
```

## VideoInputMute.Set

| Command                   | Parameter(s) | Description                                                                                                                                                                       |
|---------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoInputMute.Set</b> | videomute    | Enables or disables the video input mute function. <ul style="list-style-type: none"> <li><b>videomute:</b> [true, false] — true blocks the input video from encoding.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoInputMute.Set","params":{"videomute":"Y"}}
```

**Reply:**

```
{"id":"X","result":{"videomute":"Y"},"jsonrpc":"2.0"}
```

## MainStreamVideoEncode.Get

| Command                          | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MainStreamVideoEncode.Get</b> | (none)       | Returns the current main stream video encoding parameters. <ul style="list-style-type: none"> <li><b>chn:</b> [0] — channel number (currently only channel 0).</li> <li><b>encodetype:</b> [h264, h265] — encoding type.</li> <li><b>rctype:</b> [cbr, vbr] — rate control type. cbr = fixed bitrate; vbr = variable bitrate.</li> <li><b>framerate:</b> [0, 15, 25, 30, 50, 60] — frame rate. 0 = follow source.</li> <li><b>gop:</b> [1–300] — group of pictures interval.</li> <li><b>bitrate:</b> [128–60000] — bitrate in kbps.</li> <li><b>resolution:</b> Output resolution (e.g., 1920x1080).</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"MainStreamVideoEncode.Get"}
```

**Reply:**

```
{"id":"User1","result":{"chn":0,"encodetype":"h265","rctype":"vbr","framerate":"60","gop":"60","bitrate":"1024","resolution":"1920x1080"},"jsonrpc":"2.0"}
```

## MainStreamVideoEncode.Set

| Command                          | Parameter(s)                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MainStreamVideoEncode.Set</b> | chn, encodetype, rctype, framerate, gop, bitrate, resolution | Sets the main stream video encoding parameters. <ul style="list-style-type: none"> <li>• <b>chn</b>: [0] — Required.</li> <li>• <b>encodetype</b>: [h264, h265] — encoding type.</li> <li>• <b>rctype</b>: [cbr, vbr].</li> <li>• <b>framerate</b>: [0, 15, 25, 30, 50, 60].</li> <li>• <b>gop</b>: [1–300].</li> <li>• <b>bitrate</b>: [128–60000] in kbps.</li> <li>• <b>resolution</b>: e.g., 1920x1080.</li> </ul> |

### Request:

```
{ "jsonrpc": "2.0", "id": "User1", "method": "MainStreamVideoEncode.Set", "params": { "chn": 0, "encodetype": "h264", "rctype": "vbr", "framerate": 60, "gop": 60, "bitrate": 1024, "resolution": "1920x1080" } }
```

### Reply:

```
{ "id": "User1", "result": { "chn": 0, "encodetype": "h264", "rctype": "vbr", "framerate": 60, "gop": 60, "bitrate": 1024, "resolution": "1920x1080" }, "jsonrpc": "2.0" }
```

## MainStreamVideoQp.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                                                                                          |
|------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MainStreamVideoQp.Get</b> | (none)       | Returns the current QP (quantization parameter) values for the main video stream. <ul style="list-style-type: none"> <li>• <b>chn</b>: [0] — channel number.</li> <li>• <b>minqp</b>: [0–51] — lower value = better image quality.</li> <li>• <b>maxqp</b>: [0–51] — must be &gt;= minqp.</li> </ul> |

### Request:

```
{ "jsonrpc": "2.0", "id": "X", "method": "MainStreamVideoQp.Get" }
```

## MainStreamVideoQp.Set

| Command                      | Parameter(s)      | Description                                                                                                                                                                                                                                       |
|------------------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MainStreamVideoQp.Set</b> | chn, minqp, maxqp | Sets the QP (quantization parameter) values for the main video stream image quality. <ul style="list-style-type: none"> <li>• <b>chn</b>: [0].</li> <li>• <b>minqp</b>: [0–51].</li> <li>• <b>maxqp</b>: [0–51] — must be &gt;= minqp.</li> </ul> |

### Request:

```
{ "jsonrpc": "2.0", "id": "User1", "method": "MainStreamVideoQp.Set", "params": { "chn": 0, "minqp": 0, "maxqp": 51 } }
```

### Reply:

```
{ "id": "User1", "result": { "chn": 0, "minqp": 0, "maxqp": 51 }, "jsonrpc": "2.0" }
```

## ProfileSelection.Get

| Command                     | Parameter(s) | Description                                                                                 |
|-----------------------------|--------------|---------------------------------------------------------------------------------------------|
| <b>ProfileSelection.Get</b> | (none)       | Returns the current profile mode selection and associated encoding parameter configuration. |

Request:

```
{"jsonrpc":"2.0","id":"X","method":"ProfileSelection.Get"}
```

## ProfileSelection.Set

| Command                     | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ProfileSelection.Set</b> | profilemode  | <p>Sets the encoding profile mode. Each mode applies a predefined set of video/audio encoding defaults.</p> <ul style="list-style-type: none"> <li><b>1:</b> Standard (default) — H.265, VBR, 16 Mbps.</li> <li><b>3:</b> AT-OME-CS31 Compatible — H.265, VBR, 6 Mbps, AAC audio, 100M network.</li> <li><b>4:</b> Custom — fully user-configurable.</li> <li><b>5:</b> RTMP — H.264, CBR, 6.8 Mbps, 30fps, AAC audio.</li> </ul> <p><b>Note:</b> When setting a profile, configure video/audio encoding parameters via MainStreamVideoEncode.Set, AudioEncode.Set, etc. Profile 3 enforces H.265 encoding and cannot be changed.</p> |

Request:

```
{"jsonrpc":"2.0","id":"X","method":"ProfileSelection.Set","params":{"profilemode":1}}
```

## ImageDisplay.Get

| Command                 | Parameter(s) | Description                                                                   |
|-------------------------|--------------|-------------------------------------------------------------------------------|
| <b>ImageDisplay.Get</b> | (none)       | Returns the current image basemap (background image) display policy settings. |

Request:

```
{"jsonrpc":"2.0","id":"X","method":"ImageDisplay.Get"}
```

Reply:

```
{"id":"X","result":{"display":"Y"},"jsonrpc":"2.0"}
```

## ImageDisplay.Set

| Command                 | Parameter(s)   | Description                                                            |
|-------------------------|----------------|------------------------------------------------------------------------|
| <b>ImageDisplay.Set</b> | display params | Sets the image basemap display policy when no video source is present. |

## ImageReset.Set

| Command               | Parameter(s) | Description                                                                                                                                                    |
|-----------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ImageReset.Set</b> | (none)       | <p>Restores the image basemap to factory default.</p> <ul style="list-style-type: none"> <li><b>result:</b> [true, false] — true indicates success.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "ImageReset.Set"}
```

**Reply:**

```
{"id": "X", "result": "Y", "jsonrpc": "2.0"}
```

## Video Streaming Commands (Encoder)

### VideoMultiIpStream.Get

| Command                       | Parameter(s) | Description                                                                                                  |
|-------------------------------|--------------|--------------------------------------------------------------------------------------------------------------|
| <b>VideoMultiIpStream.Get</b> | (none)       | Returns the current multicast video stream configuration including stream name, multicast address, and port. |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "VideoMultiIpStream.Get"}
```

### VideoMultiIpStream.Set

| Command                       | Parameter(s)         | Description                                                                                                    |
|-------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------|
| <b>VideoMultiIpStream.Set</b> | stream config params | Configures the multicast video stream parameters including stream name, multicast IP address, and port number. |

**Request:**

```
{"jsonrpc": "2.0", "id": "VideoMultiIpStreamSetResults", "method": "VideoMultiIpStream.Set", "params": {"transenable": true, "streamname": "AT-LUMA-E4211-B2C12C", "usetype": "userdefine", "multiip": {"portnum": 1000, "ipaddr": "239.80.222.213"}}
```

**Reply:**

```
{"id": "VideoMultiIpStreamSetResults", "jsonrpc": "2.0", "result": {"multiip": {"ipaddr": "239.80.222.213", "portnum": 1000}, "streamname": "AT-LUMA-E4211-B2C12C", "transenable": true, "usetype": "userdefine"}}
```

### StreamTtl.Get

| Command              | Parameter(s) | Description                                                                                                                                                                         |
|----------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>StreamTtl.Get</b> | (none)       | Returns the current multicast stream TTL (time-to-live) value. <ul style="list-style-type: none"> <li><b>tll:</b> TTL value (number of hops before packet is discarded).</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "StreamTtl.Get"}
```

### StreamTtl.Set

| Command              | Parameter(s) | Description                          |
|----------------------|--------------|--------------------------------------|
| <b>StreamTtl.Set</b> | tll          | Sets the multicast stream TTL value. |

| Command | Parameter(s) | Description                                                              |
|---------|--------------|--------------------------------------------------------------------------|
|         |              | <ul style="list-style-type: none"> <li><b>ttl:</b> TTL value.</li> </ul> |

### DanteStreamLanSelection.Get

| Command                            | Parameter(s) | Description                                                             |
|------------------------------------|--------------|-------------------------------------------------------------------------|
| <b>DanteStreamLanSelection.Get</b> | (none)       | Returns which LAN port is selected for Dante audio stream transmission. |

#### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"DanteStreamLanSelection.Get"}
```

#### Reply:

```
{"id":"User1","result":{"lanid":"audiolan"},"jsonrpc":"2.0"}
```

### DanteStreamLanSelection.Set

| Command                            | Parameter(s)  | Description                                                      |
|------------------------------------|---------------|------------------------------------------------------------------|
| <b>DanteStreamLanSelection.Set</b> | lan selection | Sets which LAN port is used for Dante audio stream transmission. |

#### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"DanteStreamLanSelection.Set","params":{"lanid":"audiolan"}}
```

#### Reply:

```
{"id":"User1","result":{"lanid":"audiolan"},"jsonrpc":"2.0"}
```

## RTMP Commands (Encoder)

**NOTE:** RTMP commands are only active when ProfileSelection is set to mode 5 (RTMP).

### Rtmp.Get

| Command         | Parameter(s) | Description                                                                                                      |
|-----------------|--------------|------------------------------------------------------------------------------------------------------------------|
| <b>Rtmp.Get</b> | (none)       | Returns the current RTMP configuration parameters including server URL, stream key, and authentication settings. |

#### Request:

```
{"jsonrpc":"2.0","id":"X","method":"Rtmp.Get"}
```

### Rtmp.Set

| Command         | Parameter(s)       | Description                                                                                                                                                                                        |
|-----------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Rtmp.Set</b> | rtmp config params | Configures the RTMP streaming parameters including server URL, stream key, and authentication credentials.<br><b>Note:</b> The password field is specified as "authpassword" in the params object. |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "Rtmp.Set", "params": { "enable": true, "url": "rtmp://192.168.1.100/live", "streamname": "stream", "advanced": true, "port": 1935, "username": "admin", "authpassword": "admin" } }
```

**Reply:**

```
{ "id": "User1", "result": { "enable": true, "url": "rtmp://192.168.1.100/live", "streamname": "stream", "advanced": true, "port": 1935, "username": "admin", "authpassword": "admin" }, "jsonrpc": "2.0" }
```

## RtmpRunStatus.Get

| Command                  | Parameter(s) | Description                                                                                                                                                 |
|--------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RtmpRunStatus.Get</b> | (none)       | Returns the current RTMP streaming run status.<br><ul style="list-style-type: none"> <li><b>runstatus:</b> [notavailable, ready, active, error].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "RtmpRunStatus.Get" }
```

**Reply:**

```
{ "id": "X", "result": { "runstatus": "X", "error": { "code": "Y", "message": "Z" }, "jsonrpc": "2.0" }
```

## Preview Stream Commands (Encoder)

### PreviewStreamEnable.Get

| Command                        | Parameter(s) | Description                                                                                                                                   |
|--------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PreviewStreamEnable.Get</b> | (none)       | Returns whether the preview (thumbnail) stream is enabled.<br><ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "PreviewStreamEnable.Get" }
```

### PreviewStreamEnable.Set

| Command                        | Parameter(s) | Description                                                                                                                |
|--------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>PreviewStreamEnable.Set</b> | enable       | Enables or disables the preview stream.<br><ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "PreviewStreamEnable.Set", "params": { "previewenabled": "Y" } }
```

**Reply:**

```
{ "id": "X", "result": { "previewenabled": "Y", "jsonrpc": "2.0" }
```

## PreviewClientCnt.Get

| Command                     | Parameter(s) | Description                                                                                                                                                                                        |
|-----------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PreviewClientCnt.Get</b> | (none)       | Returns the number of clients currently connected to the preview stream. Maximum is 5. <ul style="list-style-type: none"> <li><b>result:</b> Number of connected preview clients (0–5).</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "PreviewClientCnt.Get"}
```

### Reply:

```
{"id": "X", "result": {"count": "Y"}, "jsonrpc": "2.0"}
```

## Audio Commands (Encoder)

### AudioInputFormat.Get

| Command                     | Parameter(s) | Description                                                                                                                                                                                                                                                                                                     |
|-----------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInputFormat.Get</b> | (none)       | Returns audio input information including format, sample rate, bit depth, and channel count. <ul style="list-style-type: none"> <li><b>source:</b> [hdmiin, analogaudioin] — input channel.</li> <li><b>audioformat:</b> Format string: type;samplerate;bitdepth;channels (e.g., PCM;48kHz;16bit;2).</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "AudioInputFormat.Get"}
```

### Reply:

```
{"id": "User1", "result": {"source": "hdmiin", "audioformat": "PCM;48kHz;16bit;2"}, "jsonrpc": "2.0"}
```

### AnalogAudioInputStatus.Get

| Command                           | Parameter(s) | Description                                                                                                                                                                              |
|-----------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AnalogAudioInputStatus.Get</b> | (none)       | Returns the analog audio input signal presence status. <ul style="list-style-type: none"> <li><b>result:</b> [active, negative] — active means analog audio input is present.</li> </ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "User1", "method": "AnalogAudioInputStatus.Get"}
```

### Reply:

```
{"id": "User1", "result": "active", "jsonrpc": "2.0"}
```

### AudioInSelection.Get

| Command                     | Parameter(s) | Description                                                                                                                                       |
|-----------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInSelection.Get</b> | (none)       | Returns the currently selected audio input source. <ul style="list-style-type: none"> <li><b>audiosource:</b> [hdmiin, analogaudioin].</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"AudioInSelection.Get"}
```

## AudioInSelection.Set

| Command                     | Parameter(s) | Description                                                                                                                                                                                       |
|-----------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInSelection.Set</b> | audiosource  | Sets the audio input source. <ul style="list-style-type: none"> <li><b>audiosource:</b> [hdmiin, analogaudioin] — hdmiin selects HDMI audio; analogaudioin selects analog audio input.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"AudioInSelection.Set","params":{"audiosource":"hdmi"}}
```

**Reply:**

```
{"id":"User1","result":{"audiosource":"hdmi"},"jsonrpc":"2.0"}
```

## AudioOutputVol.Get

| Command                   | Parameter(s) | Description                                                                                                                                                                            |
|---------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutputVol.Get</b> | (none)       | Returns the current analog audio output volume level. <ul style="list-style-type: none"> <li><b>out:</b> [analogaudioout].</li> <li><b>volume:</b> [-80 to 0] volume in dB.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"AudioOutputVol.Get"}
```

**Reply:**

```
{"id":"User1","result":{"analogaudioout":"Y","volume":"Z"},"jsonrpc":"2.0"}
```

## AudioOutputVol.Set

| Command                   | Parameter(s) | Description                                                                                                                                                          |
|---------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutputVol.Set</b> | out, volume  | Sets the analog audio output volume level. <ul style="list-style-type: none"> <li><b>out:</b> [analogaudioout].</li> <li><b>volume:</b> [-80 to 0] in dB.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"AudioOutputVol.Set","params":{"out":"Y","volume":"Z"}}
```

**Reply:**

```
{"id":"User1","result":{"out":"Y","volume":"Z"},"jsonrpc":"2.0"}
```

## AudioInputMute.Get

| Command                   | Parameter(s) | Description                                                                                                                                           |
|---------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInputMute.Get</b> | (none)       | Returns the current audio input mute state. <ul style="list-style-type: none"> <li><b>mute:</b> [true, false] — true means audio is muted.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"AudioInputMute.Get"}
```

**Reply:**

```
{"id":"User1","result":{"mute":false},"jsonrpc":"2.0"}
```

## AudioInputMute.Set

| Command                   | Parameter(s) | Description                                                                                                             |
|---------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInputMute.Set</b> | mute         | Enables or disables the audio input mute. <ul style="list-style-type: none"> <li><b>mute:</b> [true, false].</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"AudioInputMute.Set","params":{"mute":true}}
```

### Reply:

```
{"id":"User1","result":{"mute":true},"jsonrpc":"2.0"}
```

## AudioInputDownmixing.Get

| Command                         | Parameter(s) | Description                                                                                                                                               |
|---------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInputDownmixing.Get</b> | (none)       | Returns whether LPCM audio input downmixing to mono is enabled. <ul style="list-style-type: none"> <li><b>downmixingenable:</b> [true, false].</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"X","method":"AudioInputDownmixing.Get"}
```

## AudioInputDownmixing.Set

| Command                         | Parameter(s)     | Description                                                                                                                                                                          |
|---------------------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioInputDownmixing.Set</b> | downmixingenable | Enables or disables LPCM audio input downmixing to mono. <ul style="list-style-type: none"> <li><b>downmixingenable:</b> [true, false] — true enables downmixing to mono.</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"X","method":"AudioInputDownmixing.Set","params":{"downmixingenable":"Y"}}
```

### Reply:

```
{"id":"X","result":{"downmixingenable":"Y"},"jsonrpc":"2.0"}
```

## AudioEncode.Get

| Command                | Parameter(s) | Description                                                                                                                                                                                                                                                                 |
|------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioEncode.Get</b> | (none)       | Returns the current audio encoding parameters. <ul style="list-style-type: none"> <li><b>codingalgorithm:</b> [lpcm, mpeg4aac_lc] — audio encoding type.</li> <li><b>bitrate:</b> Bitrate in bps. Applicable when codingalgorithm is mpeg4aac_lc (e.g., 240000).</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"AudioEncode.Get"}
```

### Reply:

```
{"id":"User1","result":{"codingalgorithm":"lpcm"},"jsonrpc":"2.0"}
```

## AudioEncode.Set

| Command                | Parameter(s)                | Description                                                                                                                                                                                                                |
|------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioEncode.Set</b> | codingalgorithm,<br>bitrate | Sets the audio encoding parameters. <ul style="list-style-type: none"><li>• <b>codingalgorithm</b>: [lpcm, mpeg4aac_lc].</li><li>• <b>bitrate</b>: Bitrate in bps. Required when codingalgorithm is mpeg4aac_lc.</li></ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "AudioEncode.Set", "params": {"codingalgorithm": "Y", "bitrate": "T"}}
```

## AES Encryption Commands (Encoder)

### AesEncryptEnabled.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                                   |
|------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesEncryptEnabled.Get</b> | (none)       | Returns whether AES stream encryption is enabled for video and audio. <ul style="list-style-type: none"><li>• <b>audio</b>: [true, false] — audio encryption state.</li><li>• <b>video</b>: [true, false] — video encryption state.</li></ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "AesEncryptEnabled.Get"}
```

### Reply:

```
{"id": "X", "result": {"video": "Y", "audio": "Z"}, "jsonrpc": "2.0"}
```

### AesEncryptEnabled.Set

| Command                      | Parameter(s) | Description                                                                                                                                                                                  |
|------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesEncryptEnabled.Set</b> | audio, video | Enables or disables AES encryption for the video and/or audio streams. <ul style="list-style-type: none"><li>• <b>audio</b>: [true, false].</li><li>• <b>video</b>: [true, false].</li></ul> |

### Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "AesEncryptEnabled.Set", "params": {"video": "Y", "audio": "Z"}}
```

### Reply:

```
{"id": "X", "result": {"video": "Y", "audio": "Z"}, "jsonrpc": "2.0"}
```

### AesEncryptPassword.Get

| Command                       | Parameter(s) | Description                                                               |
|-------------------------------|--------------|---------------------------------------------------------------------------|
| <b>AesEncryptPassword.Get</b> | (none)       | Returns the current AES encryption passwords for video and audio streams. |

| Command | Parameter(s) | Description                                                                                                                                                                                                                              |
|---------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |              | <ul style="list-style-type: none"> <li>• <b>video:</b> Video AES encryption password. Returns "" if system default is in use.</li> <li>• <b>audio:</b> Audio AES encryption password. Returns "" if system default is in use.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "AesEncryptPassword.Get"}
```

## AesEncryptPassword.Set

| Command                       | Parameter(s) | Description                                                                                                                                                                                                                                                                                              |
|-------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesEncryptPassword.Set</b> | video, audio | Sets the AES encryption passwords for video and/or audio streams. <ul style="list-style-type: none"> <li>• <b>video:</b> Password for video stream. 1–16 characters, letters and numbers only.</li> <li>• <b>audio:</b> Password for audio stream. 1–16 characters, letters and numbers only.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "User1", "method": "AesEncryptPassword.Set", "params": {"video": "123456789", "audio": "123456789"}}
```

**Reply:**

```
{"id": "User1", "result": {"video": "123456789", "audio": "123456789"}, "jsonrpc": "2.0"}
```

## AesEncryptPassword.Reset

| Command                         | Parameter(s) | Description                                                                                                                                                                        |
|---------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesEncryptPassword.Reset</b> | reset        | Restores AES encryption passwords to factory defaults. <ul style="list-style-type: none"> <li>• <b>reset:</b> [video, audio, all] — specify which password(s) to reset.</li> </ul> |

**Request:**

```
{"jsonrpc": "2.0", "id": "X", "method": "AesEncryptPassword.Reset", "params": {"reset": "Y"}}
```

**Reply:**

```
{"id": "X", "result": {"reset": "Y"}, "jsonrpc": "2.0"}
```

## AudioLAN & Dante Streaming Commands (Encoder)

### AudiolanEnabled.Get

| Command                    | Parameter(s) | Description                                                                                                                                                                                  |
|----------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudiolanEnabled.Get</b> | (none)       | Returns the current AudioLAN (Dante) network port enable status. <ul style="list-style-type: none"> <li>• <b>enable:</b> [true, false] — true means the AudioLAN port is enabled.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"AudiolanEnabled.Get"}
```

**Reply:**

```
{"id":"User1","result":{"enable":false},"jsonrpc":"2.0"}
```

## AudiolanEnabled.Set

| Command                    | Parameter(s) | Description                                                                                                                            |
|----------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudiolanEnabled.Set</b> | enable       | Enables or disables the AudioLAN (Dante) network port. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"AudiolanEnabled.Set","params":{"enable":"Y"}}
```

## AudioMultiIpStream.Get

| Command                       | Parameter(s) | Description                                                                                                  |
|-------------------------------|--------------|--------------------------------------------------------------------------------------------------------------|
| <b>AudioMultiIpStream.Get</b> | (none)       | Returns the current audio multicast stream configuration including stream name, multicast address, and port. |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"AudioMultiIpStream.Get"}
```

## AudioMultiIpStream.Set

| Command                       | Parameter(s)         | Description                                       |
|-------------------------------|----------------------|---------------------------------------------------|
| <b>AudioMultiIpStream.Set</b> | stream config params | Configures the audio multicast stream parameters. |

**Request:**

```
{"jsonrpc":"2.0","id":"AudioMultiIpStreamSetResults","method":"AudioMultiIpStream.Set","params":{"transenable":true,"streamname":"AT-LUMA-E4211-B2C12C","usetype":"default","followvideoenable":false,"multiip":{"portnum":1000,"ipaddr":"239.80.193.44"}}}
```

**Reply:**

```
{"id":"AudioMultiIpStreamSetResults","jsonrpc":"2.0","result":{"followvideoenable":false,"multiip":{"ipaddr":"239.80.193.44","portnum":1000},"streamname":"AT-LUMA-E4211-B2C12C","transenable":true,"usetype":"default"}}
```

## Decoder Commands (HW-LUMA-D4111)

The following commands apply exclusively to the HW-LUMA-D4111 LumaStream Decoder.

### HDCP & CEC Commands (Decoder)

#### OutHdcpAnnouncement.Get

| Command                        | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OutHdcpAnnouncement.Get</b> | (none)       | Returns the current HDMI output HDCP compatibility setting. <ul style="list-style-type: none"><li>• <b>hdcpversion</b>: [INPUT, HDCP1.4, HDCP2.2] — INPUT means follow the TX source HDCP version. Default: INPUT.</li><li>• <b>AUTO</b>: For testing only — follows sink capability.</li><li>• <b>NONE</b>: For testing only — disables HDCP.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "OutHdcpAnnouncement.Get"}
```

Reply:

```
{"id": "X", "result": {"hdcpversion": "T"}, "jsonrpc": "2.0"}
```

#### OutHdcpAnnouncement.Set

| Command                        | Parameter(s) | Description                                                                                                                                                       |
|--------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>OutHdcpAnnouncement.Set</b> | hdcpversion  | Sets the HDMI output HDCP compatibility version. <ul style="list-style-type: none"><li>• <b>hdcpversion</b>: [INPUT, HDCP1.4, HDCP2.2]. Default: INPUT.</li></ul> |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "OutHdcpAnnouncement.Set", "params": {"hdcpversion": "T"}}
```

Reply:

```
{"id": "X", "result": {"hdcpversion": "T"}, "jsonrpc": "2.0"}
```

#### OutHdcpAnnouncementTest.Set

| Command                            | Parameter(s) | Description                                                          |
|------------------------------------|--------------|----------------------------------------------------------------------|
| <b>OutHdcpAnnouncementTest.Set</b> | hdcpversion  | Sets the HDMI output HDCP version in test mode (for diagnostic use). |

Request:

```
{"jsonrpc": "2.0", "id": "X", "method": "OutHdcpAnnouncementTest.Set", "params": {"hdcpversion": "T"}}
```

Reply:

```
{"id": "X", "result": {"hdcpversion": "T"}, "jsonrpc": "2.0"}
```

#### VideoHDMIOut5V.Get

| Command                   | Parameter(s) | Description                                                                                                                                                                                   |
|---------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoHDMIOut5V.Get</b> | (none)       | Returns the current HDMI output 5V power disabled state. <ul style="list-style-type: none"> <li><b>HDMIOut5V:</b> [true, false] — true means the HDMI output 5V disable is active.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoHDMIOut5V.Get" }
```

**Reply:**

```
{ "id": "User1", "result": { "HDMIOut5V": true }, "jsonrpc": "2.0" }
```

## VideoHDMIOut5V.Set

| Command                   | Parameter(s) | Description                                                                                                                                                                                     |
|---------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoHDMIOut5V.Set</b> | HDMIOut5V    | Sets the HDMI output 5V power disabled state (passive mode). <ul style="list-style-type: none"> <li><b>HDMIOut5V:</b> [true, false] — true disables the HDMI output 5V power supply.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "VideoHDMIOut5V.Set", "params": { "HDMIOut5V": "Y" } }
```

## CecCommand.Get

| Command               | Parameter(s) | Description                                                                                                                                                                                                                                                                             |
|-----------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CecCommand.Get</b> | (none)       | Returns the CEC command codes currently configured on the device. <ul style="list-style-type: none"> <li><b>commandtype:</b> [0, 1, 2, 3] — 0: one-touch play; 1: standby; 2: volume up; 3: volume down.</li> <li><b>commandvalue:</b> CEC command hex code (e.g., "40 04").</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "CecCommand.Get" }
```

## CecCommand.Set

| Command               | Parameter(s)              | Description                                                                                                                                                                                       |
|-----------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CecCommand.Set</b> | commandtype, commandvalue | Sets a CEC command code for the specified command type. <ul style="list-style-type: none"> <li><b>commandtype:</b> [0, 1, 2, 3].</li> <li><b>commandvalue:</b> CEC command hex string.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "CecCommand.Set", "params": { "commandtype": 0, "commandvalue": "40 04" } }
```

**Reply:**

```
{ "id": "X", "result": { "commandtype": 0, "commandvalue": "40 04" }, "jsonrpc": "2.0" }
```

## CecCommandControl.Set

| Command                      | Parameter(s) | Description                                                                                                                                                             |
|------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CecCommandControl.Set</b> | commandtype  | Executes a stored CEC command immediately. <ul style="list-style-type: none"> <li><b>commandtype:</b> [0, 1, 2, 3] — executes the corresponding CEC command.</li> </ul> |

### Request:

```
{ "jsonrpc": "2.0", "id": "CecCommandSetResults", "method": "CecCommand.Set", "params": { "commandtype": 0, "commandvalue": "40 04" } }
```

### Reply:

```
{ "id": "CecCommandSetResults", "result": { "commandtype": 0, "commandvalue": "40 04" }, "jsonrpc": "2.0" }
```

## Video Output Commands (Decoder)

### VideoOutputStatus.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutputStatus.Get</b> | (none)       | Returns the current HDMI output video status including connection state, format, and HDCP type. <ul style="list-style-type: none"> <li><b>active:</b> [yes, no] — yes means the output is connected to a display.</li> <li><b>videoformat:</b> Format string: timing;color;bitdepth (e.g., 1920x1080,60;rgb;8bit).</li> <li><b>hdcpc:</b> [NONE, HDCP1.4, HDCP2.2].</li> </ul> |

### Request:

```
{ "jsonrpc": "2.0", "id": "User1", "method": "VideoOutputStatus.Get" }
```

### Reply:

```
{ "id": "User1", "result": { "active": "yes", "videoformat": "1920x1080,60;rgb;8bit", "hdcpc": "HDCP1.4" }, "jsonrpc": "2.0" }
```

### NoVideoSource.Get

| Command                  | Parameter(s) | Description                                                                                                                                                                                                                                                                                            |
|--------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NoVideoSource.Get</b> | (none)       | Returns the current display mode setting for when no video source is available. <ul style="list-style-type: none"> <li><b>novideodisplay:</b> [0–5] — 0: show image basemap; 1: black screen; 2: CEC shutdown; 3: disable output; 4: go to another stream; 5: serial port shutdown command.</li> </ul> |

### Request:

```
{ "jsonrpc": "2.0", "id": "User1", "method": "NoVideoSource.Get" }
```

### Reply:

```
{ "id": "User1", "jsonrpc": "2.0", "result": { "novideodisplay": 0 } }
```

### NoVideoSource.Set

| Command                  | Parameter(s)                                                                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NoVideoSource.Set</b> | novideodisplay,<br>[nexttvvideoaddr{selecttype,<br>streamname,<br>multiaddr{ipaddr,<br>portnumber}}] | Sets the display mode when no video source is active. <ul style="list-style-type: none"> <li>• <b>novideodisplay</b>: [0–5].</li> <li>• <b>selecttype</b>: [0, 1] — used when novideodisplay = 4. 0: stream name pairing; 1: multicast address pairing.</li> <li>• <b>streamname</b>: Stream name to switch to when novideodisplay = 4.</li> <li>• <b>ipaddr / portnumber</b>: Multicast address and port when selecttype = 1.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"NoVideoSource.Set","params":{"novideodisplay":0}}
```

**Reply:**

```
{"id":"X","result":{"displaytype":"Y"},"jsonrpc":"2.0"}
```

## VideoOutColorspace.Get

| Command                       | Parameter(s) | Description                                                                                                                                                                           |
|-------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutColorspace.Get</b> | (none)       | Returns the current HDMI video output color space setting. <ul style="list-style-type: none"> <li>• <b>colorspace</b>: [auto, rgb, yuv444, yuv422, yuv420]. Default: auto.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"VideoOutColorspace.Get"}
```

**Reply:**

```
{"id":"User1","result":{"colorspace":"auto"},"jsonrpc":"2.0"}
```

## VideoOutColorspace.Set

| Command                       | Parameter(s) | Description                                                                                                                                                        |
|-------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutColorspace.Set</b> | colorspace   | Sets the HDMI video output color space. <ul style="list-style-type: none"> <li>• <b>colorspace</b>: [auto, rgb, yuv444, yuv422, yuv420]. Default: auto.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoOutColorspace.Set","params":{"colorspace":"Y"}}
```

## VideoOutSupportedTimingList.Get

| Command                                | Parameter(s) | Description                                                                                                                                                                                                                                     |
|----------------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutSupportedTimingList.Get</b> | (none)       | Returns the list of video output timings supported by the decoder chip and display sink. <ul style="list-style-type: none"> <li>• <b>result</b>: Array of timing strings (e.g., ["3840x2160@60","3840x2160@50","1920x1080@60",...]).</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoOutSupportedTimingList.Get"}
```

## VideoOutTiming.Get

| Command                   | Parameter(s) | Description                                                                                                                                                                                                                              |
|---------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutTiming.Get</b> | (none)       | Returns the current video output timing setting. <ul style="list-style-type: none"> <li><b>timing:</b> [followinput, auto, 3840x2160@60, ...] — followinput follows the source; auto selects best for display. Default: auto.</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"VideoOutTiming.Get"}
```

### Reply:

```
{"id":"User1","result":{"timing":"auto"},"jsonrpc":"2.0"}
```

## VideoOutTiming.Set

| Command                   | Parameter(s) | Description                                                                                                                                                             |
|---------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutTiming.Set</b> | timing       | Sets the video output timing. <ul style="list-style-type: none"> <li><b>timing:</b> [followinput, auto, or a specific resolution@fps string]. Default: auto.</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"X","method":"VideoOutTiming.Set","params":{"timing":"Y"}}
```

## LowLatencyMode.Get

| Command                   | Parameter(s) | Description                                                                                                                              |
|---------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LowLatencyMode.Get</b> | (none)       | Returns the current video low latency mode enable state. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false].</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"LowLatencyMode.Get"}
```

### Reply:

```
{"id":"X","result":{"lowlatencyenable":true},"jsonrpc":"2.0"}
```

## LowLatencyMode.Set

| Command                   | Parameter(s) | Description                                                                                                                                                            |
|---------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LowLatencyMode.Set</b> | enable       | Enables or disables the low latency video output mode. <ul style="list-style-type: none"> <li><b>enable:</b> [true, false] — true enables low latency mode.</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"LowLatencyMode.Set","params":{"lowlatencyenable":false}}
```

### Reply:

```
{"id":"User1","result":{"lowlatencyenable":false},"jsonrpc":"2.0"}
```

## VideoOutColorRange.Get

| Command                       | Parameter(s) | Description                                                                                                                                       |
|-------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutColorRange.Get</b> | (none)       | Returns the current video output color range setting. <ul style="list-style-type: none"> <li><b>colorrange:</b> [auto, full, limited].</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoOutColorRange.Get"}
```

**Reply:**

```
{"id":"X","result":{"colorrange":"limited"},"jsonrpc":"2.0"}
```

## VideoOutColorRange.Set

| Command                       | Parameter(s) | Description                                                                                                                    |
|-------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoOutColorRange.Set</b> | colorrange   | Sets the video output color range. <ul style="list-style-type: none"> <li><b>colorrange:</b> [auto, full, limited].</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoOutColorRange.Set","params":{"colorrange":"full"}}
```

**Reply:**

```
{"id":"X","result":{"colorrange":"full"},"jsonrpc":"2.0"}
```

## ImageReset.Set

| Command               | Parameter(s) | Description                                                                                                                    |
|-----------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>ImageReset.Set</b> | (none)       | Restores the image basemap to factory default. <ul style="list-style-type: none"> <li><b>result:</b> [true, false].</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"ImageReset.Set"}
```

**Reply:**

```
{"id":"X","result":"Y","jsonrpc":"2.0"}
```

## DisplayEdid.Get

| Command                | Parameter(s) | Description                                                     |
|------------------------|--------------|-----------------------------------------------------------------|
| <b>DisplayEdid.Get</b> | (none)       | Returns the EDID data reported by the connected display (sink). |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"DisplayEdid.Get"}
```

## Video Stream Selection Commands (Decoder)

### VideoStreamNameList.Get

| Command                        | Parameter(s) | Description                                                                                                                                                                                                                      |
|--------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoStreamNameList.Get</b> | (none)       | Returns the list of available video source stream names that can be paired for multicast decoding. <ul style="list-style-type: none"> <li><b>result:</b> Array of stream name strings (e.g., ["stream1", "stream2"]).</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "VideoStreamNameList.Get" }
```

## SelectVideoStream.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SelectVideoStream.Get</b> | (none)       | Returns the current multicast video stream pairing configuration. <ul style="list-style-type: none"> <li><b>selecttype:</b> [0, 1] — 0: stream name pairing; 1: multicast address pairing.</li> <li><b>streamname:</b> Paired stream name.</li> <li><b>ipaddr:</b> Multicast address (e.g., 226.223.2.26). 0.0.0.0 if not set.</li> <li><b>portnumber:</b> Multicast port number.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "SelectVideoStream.Get" }
```

## SelectVideoStream.Set

| Command                      | Parameter(s)                                                   | Description                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SelectVideoStream.Set</b> | selecttype,<br>streamname,<br>multiaddr{ipaddr,<br>portnumber} | Sets the multicast video stream to decode. <ul style="list-style-type: none"> <li><b>selecttype:</b> [0, 1] — 0: stream name; 1: multicast address.</li> <li><b>streamname:</b> 1–24 characters; letters, numbers, and hyphens only.</li> <li><b>ipaddr:</b> Multicast IP address (e.g., 226.223.2.26).</li> <li><b>portnumber:</b> Multicast port number (int16).</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "SelectVideoStream.Set", "params": { "selecttype": 0, "streamname": "ATL", "multiaddr": { "ipaddr": "192.168.1.10", "portnumber": 62000 } } }
```

**Reply:**

```
{ "id": "User1", "result": { "streamname": "ATL", "multiaddr": { "ipaddr": "192.168.1.10", "portnumber": "62000" } }, "jsonrpc": "2.0" }
```

## Audio Output Commands (Decoder)

### AnalogAudioOutputVol.Set

| Command                         | Parameter(s) | Description                                                                                                                                                          |
|---------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AnalogAudioOutputVol.Set</b> | out, volume  | Sets the analog audio output volume level. <ul style="list-style-type: none"> <li><b>out:</b> [analogaudioout].</li> <li><b>volume:</b> [-80 to 0] in dB.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "AudioOutputVolSetResults", "method": "AudioOutputVol.Set", "params": { "out": "analogaudioout", "volume": -29 } }
```

**Reply:**

```
{ "id": "AudioOutputVolSetResults", "result": { "audioout": "analogaudioout", "volume": -29 }, "jsonrpc": "2.0" }
```

## AudioOutputMute.Get

| Command                    | Parameter(s) | Description                                                                                                                                                   |
|----------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutputMute.Get</b> | (none)       | Returns the current audio output mute state. <ul style="list-style-type: none"> <li><b>mute:</b> [true, false] — true means audio output is muted.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AudioOutputMute.Get", "params": { "audioout": "hdmi" } }
```

**Reply:**

```
{ "id": "X", "jsonrpc": "2.0", "result": { "audioout": "hdmi", "mute": false } }
```

## AudioOutputMute.Set

| Command                    | Parameter(s) | Description                                                                                                              |
|----------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutputMute.Set</b> | mute         | Enables or disables the audio output mute. <ul style="list-style-type: none"> <li><b>mute:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AudioOutputMute.Set", "params": { "audioout": "Z", "mute": "W" } }
```

## AudioOutputDownmixing.Get

| Command                          | Parameter(s) | Description                                                                                                                                           |
|----------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutputDownmixing.Get</b> | (none)       | Returns whether audio output downmixing to mono is enabled. <ul style="list-style-type: none"> <li><b>downmixingenable:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AudioOutputDownmixing.Get" }
```

**Reply:**

```
{ "id": "X", "result": { "downmixingenable": "Y" }, "jsonrpc": "2.0" }
```

## AudioOutputDownmixing.Set

| Command                          | Parameter(s)     | Description                                                                                                                                    |
|----------------------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutputDownmixing.Set</b> | downmixingenable | Enables or disables audio output downmixing to mono. <ul style="list-style-type: none"> <li><b>downmixingenable:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AudioOutputDownmixing.Set", "params": { "downmixingenable": "Y" } }
```

**Reply:**

```
{ "id": "X", "result": { "downmixingenable": "Y" }, "jsonrpc": "2.0" }
```

## AudioOutDelay.Get

| Command                  | Parameter(s) | Description                                                                                                                                                              |
|--------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutDelay.Get</b> | (none)       | Returns the current audio output delay (for lip sync correction). <ul style="list-style-type: none"> <li><b>audiooutdelay:</b> [0–500] delay in milliseconds.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AudioOutDelay.Get" }
```

## AudioOutDelay.Set

| Command                  | Parameter(s)  | Description                                                                                                                                           |
|--------------------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioOutDelay.Set</b> | audiooutdelay | Sets the audio output delay for lip sync correction. <ul style="list-style-type: none"> <li><b>audiooutdelay:</b> [0–500] in milliseconds.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AudioOutDelay.Set", "params": { "audiooutdelay": 180 } }
```

**Reply:**

```
{ "id": "User1", "result": { "audiooutdelay": 180 }, "jsonrpc": "2.0" }
```

## Audio Stream Selection Commands (Decoder)

### AudioStreamNameList.Get

| Command                        | Parameter(s) | Description                                                                                                                                                                                       |
|--------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AudioStreamNameList.Get</b> | (none)       | Returns the list of available audio source stream names that can be paired for multicast decoding. <ul style="list-style-type: none"> <li><b>result:</b> Array of stream name strings.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AudioStreamNameList.Get" }
```

### SelectAudioStream.Get

| Command                      | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SelectAudioStream.Get</b> | (none)       | Returns the current multicast audio stream pairing configuration. <ul style="list-style-type: none"> <li>• <b>followvideoenable</b>: [true, false] — true means audio follows the video source stream.</li> <li>• <b>selecttype</b>: [0, 1].</li> <li>• <b>streamname</b>: Paired audio stream name.</li> <li>• <b>ipaddr / portnumber</b>: Multicast address and port.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"SelectAudioStream.Get"}
```

## SelectAudioStream.Set

| Command                      | Parameter(s)                                                             | Description                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SelectAudioStream.Set</b> | followvideoenable, selecttype, streamname, multiaddr{ipaddr, portnumber} | Sets the multicast audio stream to decode. <ul style="list-style-type: none"> <li>• <b>followvideoenable</b>: [true, false] — Required. True to follow the video source audio.</li> <li>• <b>selecttype</b>: [0, 1] — 0: stream name; 1: multicast address.</li> <li>• <b>streamname</b>: 1–24 characters; letters, numbers, and hyphens only.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"User1","method":"SelectAudioStream.Set","params":{"followvideoenable":false,"selecttype":0,"streamname":"ATLAUDIO"}}
```

**Reply:**

```
{"id":"User1","result":{"followvideoenable":false,"selecttype":0,"streamname":"ATLAUDIO"},"jsonrpc":"2.0"}
```

## AES Decryption Commands (Decoder)

### AesDecryptPassword.Get

| Command                       | Parameter(s) | Description                                                                                                                                                                                                                                                                                                        |
|-------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesDecryptPassword.Get</b> | (none)       | Returns the current AES decryption passwords for video and audio streams. <ul style="list-style-type: none"> <li>• <b>video</b>: Video AES decryption password. Returns "" if system default is in use.</li> <li>• <b>audio</b>: Audio AES decryption password. Returns "" if system default is in use.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"AesDecryptPassword.Get"}
```

### AesDecryptPassword.Set

| Command                       | Parameter(s) | Description                                                                                                                                                                                                                                                                                                                            |
|-------------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesDecryptPassword.Set</b> | video, audio | Sets the AES decryption passwords for video and/or audio streams. <ul style="list-style-type: none"> <li>• <b>video:</b> AES decryption password for video stream. 1–16 characters, letters and numbers only.</li> <li>• <b>audio:</b> AES decryption password for audio stream. 1–16 characters, letters and numbers only.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "AesDecryptPassword.Set", "params": { "video": "123456789", "audio": "123456789" } }
```

**Reply:**

```
{ "id": "User1", "result": { "video": "123456789", "audio": "123456789" }, "jsonrpc": "2.0" }
```

## AesDecryptPassword.Reset

| Command                         | Parameter(s) | Description                                                                                                                                   |
|---------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AesDecryptPassword.Reset</b> | reset        | Restores AES decryption passwords to factory defaults. <ul style="list-style-type: none"> <li>• <b>reset:</b> [video, audio, all].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "AesDecryptPassword.Reset", "params": { "reset": "Y" } }
```

**Reply:**

```
{ "id": "X", "result": { "reset": "Y" }, "jsonrpc": "2.0" }
```

## Network & Source Control Commands (Decoder)

### LanEnabled.Get

| Command               | Parameter(s) | Description                                                                                                                   |
|-----------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>LanEnabled.Get</b> | (none)       | Returns the current LAN port enable status. <ul style="list-style-type: none"> <li>• <b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "User1", "method": "LanEnabled.Get" }
```

**Reply:**

```
{ "id": "User1", "result": { "ethenable": false }, "jsonrpc": "2.0" }
```

### LanEnabled.Set

| Command               | Parameter(s) | Description                                                                                                         |
|-----------------------|--------------|---------------------------------------------------------------------------------------------------------------------|
| <b>LanEnabled.Set</b> | enable       | Enables or disables the LAN port. <ul style="list-style-type: none"> <li>• <b>enable:</b> [true, false].</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "LanEnabledSetResults", "method": "LanEnabled.Set", "params": { "ethenable": false } }
```

**Reply:**

```
{"id":"LanEnabledSetResults","result":{"ethenable":false},"jsonrpc":"2.0"}
```

## SwitchVideoSource.Set

| Command                      | Parameter(s) | Description                                                                                                                                                                                                              |
|------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>SwitchVideoSource.Set</b> | direction    | Switches the video source up or down through the available source list. <ul style="list-style-type: none"> <li><b>direction:</b> [up, down] — up moves to the previous source; down moves to the next source.</li> </ul> |

### Request:

```
{"jsonrpc":"2.0","id":"X","method":"SwitchVideoSource.Set","params":{"direction":"up"}}
```

### Reply:

```
{"id":"X","jsonrpc":"2.0","result":{"direction":"up"}}
```

## Video Wall Commands (Decoder)

### StandardVideoWallInfo.Get

| Command                          | Parameter(s) | Description                                                                                                             |
|----------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>StandardVideoWallInfo.Get</b> | (none)       | Returns the current standard video wall configuration parameters including wall layout, position, and display settings. |

### Request:

```
{"jsonrpc":"2.0","id":"User1","method":"StandardVideoWallInfo.Get"}
```

### StandardVideoWallInfo.Set

| Command                          | Parameter(s)             | Description                                                                                                                                                                                                                                                                                  |
|----------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>StandardVideoWallInfo.Set</b> | video wall config params | Configures the standard video wall parameters including wall layout dimensions, the decoder's position in the wall, and display mode settings. <p><b>Note:</b> For specific parameter details including wmenable, refer to the video wall configuration section of the product's web UI.</p> |

### Request:

```
{"jsonrpc":"2.0","id":"StandardVideoWallInfoSetResults","method":"StandardVideoWallInfo.Set","params":{"vmenable":true,"vwmatrix":"2_2_1_1"}}
```

### VideoWallBezel.Set

| Command                   | Parameter(s)                        | Description                                                                       |
|---------------------------|-------------------------------------|-----------------------------------------------------------------------------------|
| <b>VideoWallBezel.Set</b> | totalwidth, totalheight, topborder, | Sets the video wall bezel compensation parameters. All values are in millimeters. |

| Command | Parameter(s)                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | bottomborder,<br>leftborder, rightborder | <ul style="list-style-type: none"> <li>• <b>totalwidth:</b> Total physical width of the TV frame in mm (100–10000).</li> <li>• <b>totalheight:</b> Total physical height of the TV frame in mm (100–10000).</li> <li>• <b>topborder:</b> Top bezel width in mm.</li> <li>• <b>bottomborder:</b> Bottom bezel width in mm.</li> <li>• <b>leftborder:</b> Left bezel width in mm.</li> <li>• <b>rightborder:</b> Right bezel width in mm.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "VideoWallBezel.Set", "params": { "totalwidth": 320, "totalheight": 260, "topborder": 0, "bottomborder": 0, "leftborder": 0, "rightborder": 20 } }
```

**Reply:**

```
{ "id": "X", "jsonrpc": "2.0", "result": { "bottomborder": 0, "leftborder": 0, "rightborder": 20, "topborder": 0, "totalheight": 260, "totalwidth": 320 } }
```

## VideoWallBezel.Get

| Command                   | Parameter(s) | Description                                                                                                                                                                                                                                                                |
|---------------------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoWallBezel.Get</b> | (none)       | Returns the current video wall bezel compensation configuration. <ul style="list-style-type: none"> <li>• <b>totalwidth, totalheight:</b> Total frame dimensions in mm.</li> <li>• <b>topborder, bottomborder, leftborder, rightborder:</b> Bezel widths in mm.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "VideoWallBezel.Get" }
```

**Reply:**

```
{ "id": "X", "jsonrpc": "2.0", "result": { "bottomborder": 0, "leftborder": 0, "rightborder": 20, "topborder": 0, "totalheight": 260, "totalwidth": 320 } }
```

## VideoWallDisplayPref.Set

| Command                         | Parameter(s) | Description                                                                                                                                                                                                 |
|---------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoWallDisplayPref.Set</b> | rotate       | Sets the image rotation angle for the video wall display. <ul style="list-style-type: none"> <li>• <b>rotate:</b> [0, 90, 180, 270] — rotation in degrees. Only these four values are supported.</li> </ul> |

**Request:**

```
{ "jsonrpc": "2.0", "id": "X", "method": "VideoWallDisplayPref.Set", "params": { "rotate": 90 } }
```

**Reply:**

```
{ "id": "X", "jsonrpc": "2.0", "result": { "rotate": 90 } }
```

## VideoWallDisplayPref.Get

| Command                         | Parameter(s) | Description                                       |
|---------------------------------|--------------|---------------------------------------------------|
| <b>VideoWallDisplayPref.Get</b> | (none)       | Returns the current image rotation angle setting. |

| Command | Parameter(s) | Description                                                                                                             |
|---------|--------------|-------------------------------------------------------------------------------------------------------------------------|
|         |              | <ul style="list-style-type: none"> <li><b>rotate:</b> [0, 90, 180, 270] — current rotation angle in degrees.</li> </ul> |

**Request:**

```
{"jsonrpc":"2.0","id":"X","method":"VideoWallDisplayPref.Get"}
```

**Reply:**

```
{"id":"X","jsonrpc":"2.0","result":{"rotate":90}}
```

## Events & Notifications

Events are unsolicited messages sent from the device to connected clients when the device state changes. The "id" field in the event message identifies the event type. Events do not require a params object; they are push notifications only.

### Encoder Events (HW-LUMA-E4211)

| Command                            | Parameter(s) | Description                                                                                                                                                                                                                                                  |
|------------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoInputStatusOnUpdate</b>    | (event)      | Reported when the video input status changes. Contains active, signal, videoformat, and hdcp fields. Same structure as VideoInputStatus.Get response.                                                                                                        |
| <b>AudioInputFormatOnUpdate</b>    | (event)      | Reported when the audio input format changes. Contains source and audioformat fields. Same structure as AudioInputFormat.Get response.                                                                                                                       |
| <b>LanStatusOnUpdate</b>           | (event)      | Reported when network port connection status changes. Contains lanid, link, speed, mac, and lldp fields.                                                                                                                                                     |
| <b>RtmpRunStatusOnUpdate</b>       | (event)      | Reported when RTMP streaming status changes. Contains runstatus field [notavailable, ready, active, error]. When status is error, code and message fields are also included.<br><b>Note:</b> Only active when ProfileSelection is set to RTMP mode (mode 5). |
| <b>PreviewClientCntOnUpdate</b>    | (event)      | Reported when the number of preview stream client connections changes. Result contains the client count (max 5).                                                                                                                                             |
| <b>AnalogAudioInStatusOnUpdate</b> | (event)      | Reported when the analog audio input presence status changes. Result is [active, negative].<br><b>Note:</b> Only reported when audio input source is set to analog audio input.                                                                              |
| <b>DanteInfoOnUpdate</b>           | (event)      | Reported when Dante network information changes. Contains name, ip, and mac fields.                                                                                                                                                                          |

### Decoder Events (HW-LUMA-D4111)

| Command                       | Parameter(s) | Description                                                                                              |
|-------------------------------|--------------|----------------------------------------------------------------------------------------------------------|
| <b>LanStatusOnUpdate</b>      | (event)      | Reported when network port connection status changes. Contains lanid, link, speed, mac, and lldp fields. |
| <b>HdmiOutputInfoOnUpdate</b> | (event)      | Reported when the HDMI output video information changes. Contains active, videoformat, and hdcp fields.  |

| Command                      | Parameter(s)      | Description                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>VideoWallSwitchSource</b> | (multicast event) | <p>A multicast event broadcast to all decoders on a video wall for synchronized source switching. Sent to 226.2.0.1:4322 in JSON-RPC format. Contains videosource and audiosource configuration, along with a videowalldevlist array of target decoder addresses.</p> <p><b>Note:</b> This is a video wall controller broadcast message, not a standard WebSocket/Telnet API command.</p> |

## Version History

---

This document combines API information from both the HW-LUMA-E4211 encoder (source document version 4.3) and the HW-LUMA-D4111 decoder (source document version 2.9).

### Atlona Document History

| Version | Date       | Notes                                                                                                                             |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 1.0     | April 2026 | Combined encoder and decoder API reference document created from manufacturer source documents v4.3 (encoder) and v2.9 (decoder). |