

RS-232 and Network Command Catalog

For JSON RPC/Pulse Based Projectors

For UDM

End User

Reference guide

2.2

2023-01-24

Table of contents

Pulse API	3
Introduction	3
Connecting to Pulse services	3
Quick start guide	4
Object and method naming	5
Type support	5
Parameters	6
Authentication	6
Service API	7
Methods	7
Properties	8
Signals	11
Notifications	13
Introspection API	14
Object changed signal	17
File endpoints	18
Programmers guide	20
Best practice	20
Basic operation	20
Sources	23
Illumination	29
Picture settings	33
Warping with grid files	36
Blending with images	37
Black level adjustment with images	39
Environment information	40
ECO mode	43
Important note about the API documentation	43
Properties	44
Methods	304
Signals	408
Files	417
Property index	424
Method index	436
Signal index	439
File index	440

Pulse API

This document describes the application programmers interface to **Pulse** projectors.

How to connect to the projector, the communication protocol and a programmers guide is presented in the following section.

Introduction

The facade API is based on the **JSON-RPC 2.0** protocol and provides access to **Pulse** services to clients. The services can be accessed through the network using the TCP/IP protocol, or using a RS232 serial cable.

Connecting to **Pulse** services

Network

If the projector is on a network, TCP/IP can be used to connect to **Pulse** services. The service is available on port number 9090.

Discovering a projector on the local network

The projector can be discovered by using the simple search and discovery protocol (SSDP). To discover only the projectors on the network, use the following search:

"urn:schemas-upnp-org:device:projector:1"

Serial port

A serial cable can be connected to the projector in order to access the **Pulse** services.

Connect the projector and host using a standard serial cable with 9-pin female to the host, and 9-pin male to the projector. Pin 2 connects to pin 2, pin 3 connects to pin 3 and pin 5 connects to pin 5.

RS232 Communication Parameters

Parameter	Value
Baud rate	19200
Parity	None
Data bits	8
Stop bits	1
Flow control	None

Quick start guide

The following sections are examples of frequently used commands, just to get you started. This assumes that the connection is set up as described in the previous chapter. The type of connection is not important. The same commands are available for all connection types.

Power on projector

Request

```
{
  "jsonrpc": "2.0",
  "method": "system.poweron"
}
```

Power off projector

Request

```
{
  "jsonrpc": "2.0",
  "method": "system.poweroff"
}
```

Select DisplayPort 1 as input source

Request

```
{
  "jsonrpc": "2.0",
  "method": "property.set",
  "params": {
    "property": "image.window.main.source",
    "value": "DisplayPort 1"
  }
}
```

Select HDMI as input source

Request

```
{
  "jsonrpc": "2.0",
  "method": "property.set",
  "params": {
    "property": "image.window.main.source",
    "value": "HDMI"
  }
}
```

Object and method naming

Objects and members are named using dot notation in lowercase format (JavaScript-like notation). Members are either a method, property, signal or object. A typical method part of an invocation will then look like:

```
method: "foo.echo"
```

If there are more than one object of a "kind", it may be modeled and notated like:

```
tempctrl.fans
tempctrl.fans.mainfan
tempctrl.fans.lampblower
```

In the example above it is possible to get all fans by introspecting the object represented by `tempctrl.fans`. Example: accessing the `rpm` property of the `mainfan`:

```
tempctrl.fans.mainfan.rpm
```

Type support

- **Basic types**
 - string (e.g. "hello")
 - integer (e.g 114)
 - float (e.g 3.141592653589793)
 - boolean (e.g true)
- **Container types**
 - array (e.g ["hello", "world"])
 - object (e.g {"name": "Johnny", "age": 30, "children": ["Agnes", "Tim"]})
 - dictionary with string key (e.g gold medals in 2018 Winter Olympics Pyeongchang {"Norway": 13, "Germany": 13})

Parameters

All parameters are passed by name, but the position or order of the parameters doesn't matter.

Thus:

```
{
  "jsonrpc": "2.0",
  "method": "property.set",
  "params": {
    "property": "image.window.main.source",
    "value": "DisplayPort 1"
  }
}
```

is exactly the same as:

```
{
  "method": "property.set",
  "jsonrpc": "2.0",
  "params": {
    "value": "DisplayPort 1",
    "property": "image.window.main.source"
  }
}
```

Authentication

A client session must start with an authentication request containing a secret pass code. The purpose of the authentication protocol is to set the user access level. Authentication is only necessary when a higher level than normal end user is required. For normal end user access the authentication can be skipped. To authenticate with the server use the following type of request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "authenticate", "params": { "code": 98765 }, "id": 1 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 1 }</pre>

Service API

Property	Type	Required	Comments
jsonrpc	string	yes	2.0
method	string	yes	see below
params			see below
id	string number	no	Request identifier
error	object	yes, if error	Error object - see JSON-RPC 2.0

Methods

Method invocation API

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "ledctrl.blink", "params": { "led": "systemstatus", "color": "red", "period": 42 }, "id": 3 }</pre>	<pre>{ "jsonrpc": "2.0", "result": 0, "id": 3 }</pre>

Properties

API for setting and getting property values

Set value of a property

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "objectname.propertyname", "value": 100 }, "id": 3 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 3 }</pre>

It is best practice to wait for the confirmation of the `property.set` before setting the same property again. Continuously setting the same property without waiting for confirmation may flood the server with unnecessary request and may reduce performance.

Read the value of a property

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "objectname.propertyname" }, "id": 4 }</pre>	<pre>{ "jsonrpc": "2.0", "result": 100, "id": 4 }</pre>

Read values of multiple properties

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": ["image.brightness", "image.contrast"] }, "id": 5 }</pre>	<pre>{ "jsonrpc": "2.0", "result": { "image.brightness": 0, "image.contrast": 1 }, "id": 5 }</pre>
--	--

Observe changes on one property

For change notifications, see [Notifications](#)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "property.subscribe", "params": { "property": "image.brightness" }, "id": 6 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 6 }</pre>
--	--

Observe changes on multiple properties

For change notifications, see [Notifications](#)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "property.subscribe", "params": { "property": ["image.brightness", "image.contrast"] }, "id": 7 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 7 }</pre>
--	--

Stop observing one property

For change notifications, see [Notifications](#)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "property.unsubscribe", "params": { "property": "image.brightness" }, "id": 8 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 8 }</pre>
--	--

Stop observing multiple properties

For change notifications, see [Notifications](#)

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.unsubscribe", "params": { "property": ["image.brightness", "image.contrast"] }, "id": 9 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 9 }</pre>

Signals

Subscribe to a signal

For change notifications, see [Notifications](#)

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "signal.subscribe", "params": { "signal": "modelupdated" }, "id": 10 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 10 }</pre>

Subscribe to multiple signals

For change notifications, see [Notifications](#)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "signal.subscribe", "params": { "signal": ["modelupdated", "image.processing.warp.gridchanged"] }, "id": 11 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 11 }</pre>
--	---

Unsubscribe from a signal

For change notifications, see [Notifications](#)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "signal.unsubscribe", "params": { "signal": "modelupdated" }, "id": 12 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 12 }</pre>
---	---

Unsubscribe from multiple signals

For change notifications, see [Notifications](#)

Request

Response

```
{
  "jsonrpc": "2.0",
  "method": "signal.unsubscribe",
  "params": {
    "signal": [
      "modelupdated",
      "image.processing.warp.gridchanged"
    ]
  },
  "id": 13
}
```

```
{
  "jsonrpc": "2.0",
  "result": true,
  "id": 13
}
```

Notifications

The client has to implement the notification API to retrieve notifications. Notification messages will not have an `id` and no response message must be returned.

Properties

The client must implement the `property.changed` function which receives an array of property/value pairs.

Notification

```
{
  "jsonrpc": "2.0",
  "method": "property.changed",
  "params": {
    "property": [
      {
        "objectname.propertyname": 100
      },
      {
        "otherobject.otherproperty": "bar"
      }
    ]
  }
}
```

Signals

The client must implement the `signal.callback` function which receives an array of signal/argument-list pairs.

Notification

```
{
  "jsonrpc": "2.0",
  "method": "signal.callback",
  "params": {
    "signal": [
      {
        "objectname.signalname": {
          "arg1": 100,
          "arg2": "cat"
        }
      },
      {
        "otherobject.othersignal": {
          "foo": "bar"
        }
      }
    ]
  }
}
```

Introspection API

Read metadata method

Metadata of available objects (methods, properties, signals) can be read out. The data is restricted by the client's authenticated access level. A typical usage for metadata is to set up OSD menus, etc. The format of the metadata is described here: [API introspection data format](#).

Introspection API (recursive)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "introspect", "params": { "object": "foo", "recursive": true, "id": 1 } }</pre>	<pre>{ "jsonrpc": "2.0", "result": { "object": "foo", "methods": [{ "name": "echo" }], "more": "... see API introspection data " }, "id": 1 }</pre>
--	---

- or -

```
{
  "jsonrpc": "2.0",
  "method": "introspect",
  "params": [
    "foo",
    true
  ],
  "id": 1
}
```

Introspection API (non recursive)

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "introspect", "params": { "object": "motors", "recursive": false, "id": 2 } }</pre>	<pre>{ "jsonrpc": "2.0", "result": { "name": "motors", "objects": [{ "name": "motors.motor1" }, { "name": "motors.motor2" }, { "name": "motors.motor3" }] }, "id": 2 }</pre>
--	--

- or -

```
{
  "jsonrpc": "2.0",
  "method": "introspect",
  "params": [
    "motors",
    false
  ],
  "id": 2
}
```

Property	Type	Required	Comments
jsonrpc	string	yes	2.0
method	string	yes	introspect
params	object {"object": string}	no (default = "")	"object": name of object to introspect (dot notation allowed), default/empty will introspect everything. The object and string notations are equivalent.

Property	Type	Required	Comments
	<code>{"recursive": bool}</code>	no (default=true)	recursive": if false then only object names are listed (one level). This is convenient if you want to list collections of objects
id	string number	no	Request identifier
result	object	yes	The full format of the result is described here API Introspection data format
error	object	if error	Error object - see JSON-RPC 2.0

Object changed signal

The introspect API provides a signal that triggers when new objects arrive, or when objects are removed. The name of the signal is: `modelupdated`. (See section [Signals](#) on how to subscribe and unsubscribe to signals.)

Subscribe to the model updated signal

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "signal.subscribe", "params": { "signal": "modelupdated" }, "id": 2 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 2 }</pre>

Callback method on client side

Notification

```
{
  "jsonrpc": "2.0",
  "method": "signal.callback",
  "params": {
    "signal": [
      {
        "introspect.objectchanged": {
          "object": "motors.motor1",
          "newobject": true
        }
      }
    ]
  }
}
```

Argument	Type	Decription
object	string	Name of object
isnew	bool	true: object is new, false: object is lost

File endpoints

Some objects provide endpoints for uploading or downloading various data types. For example, a warp grid can be uploaded to the warp engine.The objects that provide end points are found in the documentation below under the **File endpoints** headings.

To download a file from the projector, you must enter its URL. The URL is constructed from the following parts:

- **http://**
- Address of the projector. E.g: **192.168.1.100**
- **/api**
- File endpoint. E.g: **/image/processing/warp/file/transfer**

This will give you a URL that looks like this: **http://192.168.100/api/image/processing/warp/file/transfer**

Entering this URL in a browser will trigger a download from the projector and save the current warp grid to your download folder.

You can also use the **curl** program to do the same. E.g: **curl -O -J http://api/image/processing/warp/file/transfer**

Note that not all endpoints supports downloading the current file. In those cases you need to specify which file to download. E.g: `http://192.168.1.100/api/image/processing/warp/file/transfer/warpgrid.xml`

To upload a file to the projector, use the `curl` program, or some other tool that supports HTTP upload. To upload a warp grid to the projector from your local drive, enter the following command:

```
curl -F file=@warpgrid.xml http://192.168.1.100/api/image/processing/warp/file/transfer
```

You can also specify `-X POST` to the command, but in this case that's implied.

Programmers guide

This section describes common tasks for controlling the projector. For example, selecting input source and adjusting image properties.

Best practice

This chapter describes some basic guidelines and best practice when communicating with the projector.

Availability of information

Properties, methods and signals are not necessarily available at all times. This is because the projector has different states. One of these states is **standby**. In this state, not all the projector services are running. This means that properties and methods that are available when the projector is completely on, are not necessarily available when in standby. For example, the shutter, which is available when in **ready** or **on**, is not available in **standby**.

Tip! Do not poll properties or call methods that are not available.

Polling properties, typically by calling `property.get` at regular intervals, with a property that's currently unavailable, just creates overhead in the system. Doing this will log an internal error and also return an error to the caller.

Setting property values

Sometimes, setting a property value can actually trigger big operations in the projector, causing the projector to be busy for some time. For example, setting a color coordinate of the **RealColor P7** properties may take a while for the projector to process.

Tip! Always wait for the confirmation before setting the property value again.

When calling `property.set` always specify an `id`. Doing so means that the projector will respond with the result of the call. Not doing so, means that you will not be notified when the projector is finished with processing the request. When the projector response is received, it is safe to again set the property value. Not waiting for the result, and quickly calling multiple `property.set` may lead to unexpected results.

Basic operation

This chapter describes basic operation such as powering on/off the projector.

Projector state

To get the current operation state from the projector, use the following command.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "system.state" }, "id": 1 }</pre>	<pre>{ "jsonrpc": "2.0", "result": "on", "id": 1 }</pre>

The resulting value will be one of the following:

- "boot" - the projector is booting up
- "eco" - the projector is in ECO/power save mode
- "standby" - the projector is in standby mode
- "ready" - the projector is in ready mode
- "conditioning" - the projector is warming up
- "on" - the projector is on
- "deconditioning" - the projector is cooling down

To be notified when the state changes, a subscription must be requested, as shown in the following example.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.subscribe", "params": { "property": "system.state" }, "id": 2 }</pre>	<pre>{ "jsonrpc": "2.0", "result": true, "id": 2 }</pre>

When ever there is a change in the state, the server will notify the client as shown in the next example.

Notification

```
{
  "jsonrpc": "2.0",
  "method": "property.changed",
  "params": {
    "property": [
      {
        "system.state": "ready"
      }
    ]
  }
}
```

Power on

To power on the projector, issue the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "system.poweron", "params": { "property": "system.state" }, "id": 3 }</pre>	<pre>{ "jsonrpc": "2.0", "result": null, "id": 3 }</pre>

Notice that the **result** is **null**. This is not an error, it's just that the method does not return a proper result. If there was an error, the response would contain an **error** member that contains information about the error that occurred. Also notice that this method does not need any arguments, so the **params** member does not need to be present. Nothing bad happens if the **params** member **is** present. It will just be ignored.

If the projector already is on, or if it's in transition between states, nothing will happen. Therefore, it's good practice to verify that the projector state is either **standby** or **ready** before issuing the power on command.

Power off

To power on the projector, issue the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "system.poweroff", "params": { "property": "system.state" }, "id": 4 }</pre>	<pre>{ "jsonrpc": "2.0", "result": null, "id": 4 }</pre>

If the projector already is off, or if it's in transition between states, nothing will happen. Therefore, it's good practice to verify that the projector state is **on** before issuing the power off command.

Sources

The source input management is made up windows, sources and connectors. Each window has a source attached, and each source is made up of one or more connectors.

Active source

To get the name of the currently active source, use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "image.window.main.source" }, "id": 0 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 0, "result": "DisplayPort 1" }</pre>

List available sources

To get a list of available sources, use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "image.source.list", "id": 1 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 1, "result": ["DVI 1", "DVI 2", "DisplayPort 1", "DisplayPort 2", "Dual DVI", "Dual DisplayPort", "Dual Head DVI", "Dual Head DisplayPort", "HDBaseT", "HDMI", "SDI"] }</pre>

The response contains a list of all the available source names. The list contents will vary depending on the projector model.

Set the active source

To set the active source, first get the list of available sources as shown in the previous section. Then use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.window.main.source", "value": "DisplayPort 1" }, "id": 2 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 2, "result": true }</pre>

Connectors

The connectors are the physical input connectors of the projector. Available connectors depend on the projector model.

To list all the available connectors, use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "image.connector.list", "id": 3 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 3, "result": ["DVI 1", "DVI 2", "DisplayPort 1", "DisplayPort 2", "HDBaseT", "HDMI", "SDI"] }</pre>

To list the connectors used by a specific source, the source **object** name must be acquired. This can be done by translating the source name, or by introspection.

The easiest way is to translate the source name. Given the source name of **DisplayPort 1**, remove all non word characters and convert to all lower case characters. This can be done quite easy by using regular expressions. Example in **JavaScript** shown below.

```
const sourceName = 'DisplayPort 1';
const objectName = sourceName.replace(/\W/g, '').toLowerCase();
/* objectName is now 'displayport1' */
```

Now that we have the source object name, we can call the method that lists all the connectors used by this source.

Request

```
{
  "jsonrpc": "2.0",
  "method": "image.source.displayport1.listconnectors",
  "id": 4
}
```

Response

```
{
  "jsonrpc": "2.0",
  "id": 4,
  "result": [
    {
      "gridposition": {
        "row": 0,
        "column": 0,
        "plane": 0
      },
      "name": "DisplayPort 1"
    }
  ]
}
```

The result of the method is an array of connector information. This information contains the connector name and the grid position of the connector, which is useful when multiple connectors are used.

Source signal

Given a connector name of `DisplayPort 1`, this translated to a connector name of `displayport1`. We can then get signal information by using the following request.

Request

```
{
  "jsonrpc": "2.0",
  "method": "property.get",
  "params": {
    "property":
      "image.connector.displayport1.detectedsignal"
  },
  "id": 5
}
```

Response

```
{
  "jsonrpc": "2.0",
  "id": 5,
  "result": {
    "active": true,
    "name": "2560x1600 @ 50.10Hz",
    "vertical_total": 1638,
    "horizontal_total": 2720,
    "vertical_resolution": 1600,
    "horizontal_resolution": 2560,
    "vertical_sync_width": 6,
    "vertical_front_porch": 3,
    "vertical_back_porch": 29,
    "horizontal_sync_width": 32,
    "horizontal_front_porch": 48,
    "horizontal_back_porch": 80,
    "horizontal_frequency":
      82068.11653672549,
    "vertical_frequency":
      50.102710556641114,
    "pixel_rate": 223222961,
    "scan": "Progressive",
    "bits_per_component": 10,
    "color_space": "RGB",
    "signal_range": "0-255",
    "chroma_sampling": "4:4:4",
    "gamma_type": "POWER"
  }
}
```

Source and signal updates

To get updates when a new source is selected, or the signal on a connector change, you must listen for changes on a number of properties.

Subscribe to the source property of the window

Request

Response

```
{
  "jsonrpc": "2.0",
  "method": "property.subscribe",
  "params": {
    "property": "image.window.main.source"
  },
  "id": 6
}
```

```
{
  "jsonrpc": "2.0",
  "id": 6,
  "result": true
}
```

This will generate a property change notification when ever there is a change in the active source. E.g when the user switches from **DisplayPort 1** to **DisplayPort 2**.

The **JSON RPC** client must therefore implement the property change listener in order to process the notification.

The **JSON RPC** server will send notifications as shown below.

Notification

```
{
  "jsonrpc": "2.0",
  "method": "property.changed",
  "params": {
    "property": [
      {
        "image.window.main.source": ""
      }
    ]
  }
}
```

Notification

```
{
  "jsonrpc": "2.0",
  "method": "property.changed",
  "params": {
    "property": [
      {
        "image.window.main.source": "DisplayPort 2"
      }
    ]
  }
}
```

Note that two notifications are delivered in this case. First, when the previously selected source is deselected. Next, when the new source is selected.

Connector signal updates

To get updates when there is a change in the signal on sources connectors, each connector of the source must have a listener for the signal detection property of the connector.

The recommended way to do this is to reflect the source and connector structure of the server in the client application.

That means:

- Store all the source names and object names
 - Call `image.source.list`
 - Translate the source names into source object names as shown in the beginning of the chapter
- For each source object
 - Call `image.source.[name].listconnectors`
 - Translate the connector names to connector object names
 - Maintain a list of connectors per source
- For each connector object
 - Subscribe to `image.connector.[name].detectedsignal`

When the notifications are received, match up the connector names with the connectors used by the active source and show the new information to the user.

Keep in mind that notifications are only sent when there is an actual change in a value. Simply subscribing to a property does not get the current value. To get the current value use the `property.get` method and specify which property you are interested in.

Illumination

This section describes how to set the lamp/LED/laser power of the projector.

Illumination state

To read the state of the illumination use the following commands.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "illumination.state" }, "id": 0 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 0, "result": "Off" }</pre>

The result will be either **On** or **Off**. To receive notifications when the state changes, you need to subscribe. The following commands will request change notifications for the illumination state.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.subscribe", "params": { "property": "illumination.state" }, "id": 1 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 1, "result": true }</pre>

The client needs to implement the property change listener to receive notifications.

When a change happens, the server will send the following notifications to the client.

Notification
<pre>{ "jsonrpc": "2.0", "method": "property.changed", "params": { "property": [{ "illumination.state": "On" }] } }</pre>

Illumination sources

Different projectors will have different types of illumination sources. Some will have lasers, others will have LEDs, or a combination of both, and yet others may have xenon or UHP lamps.

To query the available sources, you must do an **introspection** of the `illumination` object. The following command shows an example of that.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "introspect", "params": { "property": "illumination.sources", "recursive": false }, "id": 2 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 2, "result": { "objects": [{ "name": "illuminations.sources.laser" }] } }</pre>

From the response, we can see that this projector has a laser illumination source. Each source has properties that give you information about the minimum, maximum and the current illumination power level.

The current power level can be read or written, the minimum and maxium power level are read only, but they are dynamic values and may change depending on the setting of the projector. The lens type and lens position may also affect the power levels.

To read the current power level, use the following commands.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "illumination.sources.laser.power" }, "id": 3 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 3, "result": 30 }</pre>

To be notified of changes in the power level, you must subscribe to property changes.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.subscribe", "params": { "property": ["illumination.sources.laser.power"] }, "id": 4 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 4, "result": true }</pre>

To set the value of the laser power, use the following command.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "illumination.sources.laser.power", "value": 40 }, "id": 5 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 5, "result": true }</pre>

After the confirmation of the value, a change notification is also sent.

Notification

```
{
  "jsonrpc": "2.0",
  "method": "property.changed",
  "params": {
    "property": [
      {
        "illumination.sources.laser.power": 40
      }
    ]
  }
}
```

To get the minimum and maximum power levels, use the following commands.

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "illumination.sources.laser.power" }, "id": 5 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 5, "result": 100 }</pre>
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "illumination.sources.laser.minpower" }, "id": 6 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 6, "result": 0 }</pre>

Picture settings

The image service has properties for controlling the image appearance. Like brightness, contrast, saturation and gamma.

This section describes how to set basic picture settings.

Brightness

Before setting any values it is a good idea to get some information about the properties. This is done with introspection.

Individual properties can not be introspected, so we must ask the owner of the property. In this case it is the image service.

To perform introspection of the image service, issue the following request.

Request

Response

```
{
  "jsonrpc": "2.0",
  "method":
  "introspect",
  "params": {
    "object": "image",
    "recursive": false
  },
  "id": 6
}
```

```
{
  "jsonrpc": "2.0",
  "id": 6,
  "result": {
    "name": "image",
    "properties": [
      {
        "name": "brightness",
        "type": {
          "base": "float",
          "min": -1,
          "max": 1,
          "step-size": 1,
          "precision": 0.01
        },
        "access": "READ_WRITE",
        "description": "Image brightness/offset. The value is normalized, 0 is
default, 1 is 100% offset."
      },
      {
        "more": "...trimmed for brevity"
      }
    ],
    "objects": [
      {
        "more": "...trimmed for brevity"
      }
    ]
  }
}
```

The result contains information about all the properties, objects and signals belonging to the image service.

Here we are only interested in the **brightness** property, so the rest of the response has been trimmed.

We can see that the **brightness** property has a **type** which is **float**. It also has some constraints that define the minimum and maximum values for the property. This information can be used to prevent sending out of range values to the server, and also give hints to the user when e.g using a slider to control the brightness value.

The **type** also has a **step-size**. This is a hint that can be used when incrementing or decrementing the value in a GUI application. The **step-size** is used as a factor to the **precision**. E.g for brightness we have a **precision** of **0.01** and a **step-size** of **1**. That means that when ever the user wants to increment the value, we can add (**step-size** x **precision**) to the current value.

To get the current brightness value, use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.get", "params": { "property": "image.brightness" }, "id": 7 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 7, "result": 0 }</pre>

To receive notification when the brightness value change, use the following subscription.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.subscribe", "params": { "property": ["image.brightness"] }, "id": 8 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 8, "result": true }</pre>

The set the brightness value, use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.brightness", "value": 0.15 }, "id": 9 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 9, "result": true }</pre>

In addition to the confirmation from the `property.set` method, a notification is also sent.

Notification

```
{
  "jsonrpc": "2.0",
  "method": "property.changed",
  "params": {
    "property": [
      {
        "image.brightness": 0.15
      }
    ]
  }
}
```

The set the other picture settings follow the same procedure as for brightness.

Warping with grid files

To globally enable warp, use the following command.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.warp.enable", "value": true }, "id": 10 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 10, "result": true }</pre>

Upload warp file

Send the warp grid file through HTTP `POST`. The following example is using `curl` to upload a file to the projector at address `192.168.1.100`.

```
 curl -X POST -F file=@warp.xml http://192.168.1.100/api/image/processing/warp/file/transfer
```

Note that `-X POST` can be omitted since this is implied when using `-F`.

To select/activate the uploaded file, use the following command.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.warp.file.selected", "value": "warp.xml" }, "id": 11 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 11, "result": true }</pre>

Finally, the grid file warping must be enabled. To do this, use the following command.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.warp.file.enable", "value": true }, "id": 12 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 12, "result": true }</pre>

Warp file format

The warp file format is the same as on the [MCM500/400](#).

Blending with images

In a multi channel setup, blending is required in order to get a seamless transition between the channels.

For this, blend edges can be configured, or one can use pre generated blend masks that are uploaded to the projector.

Uploading a blend mask

Blend masks are grayscale images in either 8 bit or 16 bit pixel resolution. The size of the blend mask must match the resolution of the blend layer of the projector. The following table shows the possible sizes.

Projector resolution	Mask resolution
WUXGA	1920 x 1200

Projector resolution

Mask resolution

WQXGA	1280 x 800
WQXGA+ (2716 x 1600)	1358 x 800
4K	1280 x 800
4K Cinemascope	1280 x 540

To upload a blend mask, use the following shell command.

```
 curl -X POST -F file=@mask.png http://192.168.1.100/api/image/processing/blend/file/transfer
```

In the example above we have a blend mask in the **PNG** format called **mask.png** and the projector has the **IP** address of **192.168.1.100**. To select the blend file that was just uploaded, us the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.blend.file.selected", "value": "mask.png" }, "id": 13 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 13, "result": true }</pre>

To enable the blend mask use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.blend.file.enable", "value": true }, "id": 14 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 14, "result": true }</pre>

Supported image formats

The supported image formats are:

- PNG - up to 16 bit
- JPEG
- TIFF

The interface only supports gray scale images, but will accept color images. It will then only use the blue channel. This is to support gray scale images saved as RGB images, we don't support the using the full colour information.

Black level adjustment with images

In a multi channel setup, blending is required in order to get a seamless transition between the channels. To compensate for the extra light in the blended regions, the black level can be increased in the non-blended regions to get a uniform black level across all channels. Black levels can be configured by using the basic controls to specify linear edges, or one can use pre generated images that are uploaded to the projector.

Uploading a black level mask

Black level masks are grayscale images in either 8 bit or 16 bit pixel resolution. The size of the black level mask must match the resolution of the black level layer of the projector. The following table shows the possible sizes.

Projector resolution	Mask resolution
WUXGA	1920 x 1200
WQXGA	1280 x 800
WQXGA+ (2716 x 1600)	1358 x 800
4K	1280 x 800
4K Cinemascope	1280 x 540

To upload a black level mask, use the following shell command.

 curl -X POST -F file=@blacklevel.png http://192.168.1.100/api/image/processing/blacklevel/file/transfer

In the example above we have a black level mask in the **PNG** format called **blacklevel.png** and the projector has the **IP** address of **192.168.1.100**.

To select the black level file that was just uploaded, use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.blacklevel.file.selected", "value": "blacklevel.png" }, "id": 15 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 15, "result": true }</pre>

To enable the black level mask use the following request.

Request	Response
<pre>{ "jsonrpc": "2.0", "method": "property.set", "params": { "property": "image.processing.blacklevel.file.enable", "value": true }, "id": 16 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 16, "result": true }</pre>

Supported image formats

The supported image formats are:

- PNG - up to 16 bit
- JPEG
- TIFF

The interface only supports gray scale, but will accept colour images. It will then only use the blue channel. This is to support gray scale images saved as RGB images. We don't support using the full color information.

Environment information

The environment service manages a lot of information in order to keep running the projector at the optimal operating conditions. Among the information available are fan speeds, temperatures and voltages. This section describes an easy method to get information from the environment service.

Temperatures

To get a snapshot or current reading of all the available temperature sensors, use the following request.

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "environment.getcontrolblocks", "params": { "type": "Sensor", "valuetype": "Temperature" }, "id": 18 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 18, "result": { "environment.laser.board0.bank0.temperature": 35.3, "environment.laser.board0.bank1.temperature": 34.8, "environment.laser.board0.bank2.temperature": 35.3, "environment.laser.board0.heatsink0.temperature": 35, "environment.laser.board0.heatsink1.temperature": 37.6, "environment.laser.board0.heatsink2.temperature": 40.4, "environment.laser.board1.bank0.temperature": 36.6, "environment.laser.board1.bank1.temperature": 36.2, "environment.laser.board1.bank2.temperature": 36.4, "environment.laser.board1.heatsink0.temperature": 34.7, "environment.laser.board1.heatsink1.temperature": 34.9, "environment.laser.board1.heatsink2.temperature": 36.5, "environment.temperature.cyclon5": 47.6, "environment.temperature.imx6": 40.1, "environment.temperature.inlet": 25.5, "environment.temperature.mainboard": 40.4, "environment.temperature.mainpower": 37.6, "environment.temperature.outlet": 29.4, "environment.temperature.scalerfpga": 52.8 } }</pre>
---	---

The result is in the form of a dictionary, where the key is the name of the sensor and the value is the temperature reading.

Fan speeds

To get fan speeds, use the following request.

Request

Response

<pre>{ "jsonrpc": "2.0", "method": "environment.getcontrolblocks", "params": { "type": "Sensor", "valuetype": "Speed" }, "id": 19 }</pre>	<pre>{ "jsonrpc": "2.0", "id": 19, "result": { "environment.fan.ar1.tacho": 1800, "environment.fan.ar2.tacho": 1850, "environment.fan.ar3.tacho": 1750, "environment.fan.ar4.tacho": 1800, "environment.fan.ar5.tacho": 1800, "environment.fan.driver.tacho": 2300, "environment.fan.optics.tacho": 2600, "environment.fan.pcb.tacho": 1400, "environment.fan.phosphorleft.tacho": 3850, "environment.fan.phosphorright.tacho": 3800, "environment.fan.psu.tacho": 1450 } }</pre>
---	---

The result is in the form of a dictionary, where the key is the name of the sensor, a fan in this case, and the value is the fan speed reading.

Other environment info

Other environment info is also available and follows the same patterns as for temperatures and fan speeds. Use the `environment.getcontrolblocks` method with different sensor types and different sensor value types to get the desired data.

The applicable sensor types are:

- Sensor
- Filter
- Controller
- Actuator
- Alarm
- GenericBlock

The applicable sensor value types are:

- Temperature
 - Speed
 - PWM
 - Voltage
 - Current
 - Power
 - Altitude
 - Pressure
 - Humidity
- ADC
 - Coordinate
 - Peltier
 - Waveform
 - Average
 - Delay
 - Difference
 - Interpolation
 - Limit
- Median
 - Noise
 - Weighting
 - Comparison
 - Threshold
 - Formula
 - Driver
 - PID
 - Mode
- Simulation
 - State
 - Pump
 - Resistance
 - Constant
 - Manual
 - Range
 - Any

ECO mode

On projectors that have **ECO** mode, special handling is required to wake up the projector. To wake up a projector that is in **ECO** mode:

- Send a wake on LAN request supplying the projectors HW (MAC) address
- Use the power button on the remote control
- Use the power button on the keypad
- Send a special command on the RS232 serial port

In the last case, waking up using the serial port, send the following ASCII characters:

```
:POWER1\r
```

Important note about the API documentation

Parts of the API are dynamic, other parts depend on peripherals or other factors. This means that the documentation shown here may not be complete with respect to a specific projector with a specific configuration. For example, if a lens is mounted that does not have motorized zoom, that part of the API will not be available, even if it's shown here. Another example is DMX. In its basic mode, only 2 channels are present. Setting it in extended mode will expose more channels, which may not be shown in this document.

The best way to know the exact API of your projector is to do an introspection as described in the previous chapters

Properties

Alphabetical list of all properties

dmx.artnet

Artnet enabled or not

MODELS

 All

Access: RW

Name	Type
artnet	bool

dmx.artnetnet

Artnet net selection

MODELS

 All

Access: RW

Name	Type
artnetnet	int
Constraints	
Minimum	0
Maximum	255
Step size	1
Precision	1

dmx.artnetuniverse

Artnet universe selection

MODELS

 All

Access: RW

Name	Type
artnetuniverse	int
Constraints	
Minimum	0
Maximum	255
Step size	1
Precision	1

dmx.mode

Current mode

MODELS

 All

Access: RW

Name	Type
mode	string

dmx.monitor.channel01.function

Description for the dmx channel

MODELS

 All

Access: R

Name	Type
function	string

dmx.monitor.channel01.offset

Offset of the channel.

MODELS

 All

Access: R

Name	Type						
offset	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

dmx.monitor.channel01.value

Current dmx value for the channel

MODELS

 All

Access: R

Name	Type						
value	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

dmx.monitor.channel02.function

Description for the dmx channel

MODELS

 All

Access: R

Name	Type
function	string

dmx.monitor.channel02.offset

Offset of the channel.

MODELS All

Access: R

Name	Type						
offset	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

dmx.monitor.channel02.value

Current dmx value for the channel

MODELS All

Access: R

Name	Type						
value	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

dmx.monitor.connectionstate.active

true indicates that a dmx (if artnet setting is deactivated) or artnet package (if artnet setting is active) was received in the last 10 seconds.

MODELS All

Access: R

Name	Type
active	bool

dmx.shutdown

Shutdown enabled or not

MODELS All

Access: RW

Name	Type
------	------

shutdown bool

dmx.shutdowntimeout

Time out for shutdown in minutes.

MODELS All

Access: RW

Name	Type
shutdowntimeout	int
Constraints	
Minimum	1
Maximum	60
Step size	1
Precision	1

dmx.startchannel

The dmx start channel [1..512].

MODELS All

Access: RW

Name	Type
startchannel	int
Constraints	
Minimum	1
Maximum	512
Step size	1
Precision	1

environment.alarmstate

Alarm state

MODELS All

Access: R

Name	Type
alarmstate	enum
Values	
"Error"	
"Alert"	
"Warning"	
"Ok"	

firmware.component.color-sensor-boot.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.color-sensor-boot.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.color-sensor-boot.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.color-sensor-boot.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.color-sensor-boot.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.color-sensor-boot.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.color-sensor-boot.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.color-sensor-run.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.color-sensor-run.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.color-sensor-run.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.color-sensor-run.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.color-sensor-run.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.color-sensor-run.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.color-sensor-run.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type				
status	enum				
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr></table>	Values	"Inactive"	"Active"	"Error"
Values					
"Inactive"					
"Active"					
"Error"					

"Finished"

firmware.component.cornet-fpga.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.cornet-fpga.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.cornet-fpga.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.cornet-fpga.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.cornet-fpga.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type		
progress	{ }		
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type		

base	int
hints	{ }

Name	Type
------	------

firmware.component.cornet-fpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.cornet-fpga.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.formatter-blue-wqxca.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.formatter-blue-wqxca.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.formatter-blue-wqxca.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.formatter-blue-wqxca.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.formatter-blue-wqxca.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.formatter-blue-wqxca.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.formatter-blue-wqxca.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum

Values
"Inactive"
"Active"
"Error"
"Finished"

firmware.component.formatter-green-wqxca.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.formatter-green-wqxca.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.formatter-green-wqxca.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.formatter-green-wqxca.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.formatter-green-wqxca.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.formatter-green-wqxca.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.formatter-green-wqxca.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.formatter-red-wqxca.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.formatter-red-wqxca.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.formatter-red-wqpga.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.formatter-red-wqpga.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.formatter-red-wqpga.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

firmware.component.formatter-red-wqpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.formatter-red-wqpga.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.intel-i210.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.intel-i210.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.intel-i210.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.intel-i210.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.intel-i210.progress

The progress of the current firmware upgrade

MODELS

All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.intel-i210.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

All

Access: R

Name	Type
requiredversion	string

firmware.component.intel-i210.status

The status of the current firmware upgrade

MODELS

All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.l1-vh-input-fpga.actualversion

The version of the currently installed firmware.

MODELS

All

Access: R

Name	Type
actualversion	string

firmware.component.l1-vh-input-fpga.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.l1-vh-input-fpga.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.l1-vh-input-fpga.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.l1-vh-input-fpga.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.l1-vh-input-fpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.l1-vh-input-fpga.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.l1-vh-input-hd-base-t-a.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.l1-vh-input-hd-base-t-a.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.l1-vh-input-hd-base-t-a.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.l1-vh-input-hd-base-t-a.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
------	------

mismatch bool

firmware.component.l1-vh-input-hd-base-t-a.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.l1-vh-input-hd-base-t-a.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.l1-vh-input-hd-base-t-a.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.l1-vh-input-hd-base-t-b.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.l1-vh-input-hd-base-t-b.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.l1-vh-input-hd-base-t-b.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.l1-vh-input-hd-base-t-b.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.l1-vh-input-hd-base-t-b.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

firmware.component.l1-vh-input-hd-base-t-b.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.l1-vh-input-hd-base-t-b.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.l1-vh-input-power-sequencer.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.l1-vh-input-power-sequencer.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.l1-vh-input-power-sequencer.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.l1-vh-input-power-sequencer.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.l1-vh-input-power-sequencer.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.l1-vh-input-power-sequencer.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.l1-vh-input-power-sequencer.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.laser-control-board-boot.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.laser-control-board-boot.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-control-board-boot.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-control-board-boot.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-control-board-boot.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.laser-control-board-boot.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-control-board-boot.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.laser-control-board-fpga.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.laser-control-board-fpga.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.laser-control-board-fpga.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.laser-control-board-fpga.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-control-board-fpga.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

firmware.component.laser-control-board-fpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-control-board-fpga.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
	Values
	"Inactive"
	"Active"
	"Error"
	"Finished"

firmware.component.laser-control-board-run.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.laser-control-board-run.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.laser-control-board-run.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.laser-control-board-run.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-control-board-run.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.laser-control-board-run.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-control-board-run.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.laser-driver-module-1-hdr.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-hdr.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-hdr.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-hdr.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-hdr.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.laser-driver-module-1-hdr.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-hdr.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	

"Finished"

firmware.component.laser-driver-module-1-primary.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-primary.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-primary.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-primary.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-primary.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type		
progress	{ }		
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type		

```
base    int
hints   { }
```

Name	Type
------	------

firmware.component.laser-driver-module-1-primary.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-primary.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.laser-driver-module-1-secondary.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-secondary.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-secondary.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-secondary.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-secondary.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.laser-driver-module-1-secondary.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-secondary.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum

Values
"Inactive"
"Active"
"Error"
"Finished"

firmware.component.laser-driver-module-1-slot-1.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-slot-1.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-slot-1.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-slot-1.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-slot-1.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.laser-driver-module-1-slot-1.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-slot-1.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.laser-driver-module-1-slot-2.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-slot-2.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-slot-2.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-slot-2.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-slot-2.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

firmware.component.laser-driver-module-1-slot-2.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-slot-2.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.laser-driver-module-1-slot-3.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-slot-3.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-slot-3.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-slot-3.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-slot-3.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.laser-driver-module-1-slot-3.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-slot-3.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.laser-driver-module-1-slot-6.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-slot-6.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-slot-6.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-slot-6.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-driver-module-1-slot-6.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.laser-driver-module-1-slot-6.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-driver-module-1-slot-6.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.laser-driver-module-1-slot-7.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.laser-driver-module-1-slot-7.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-driver-module-1-slot-7.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-driver-module-1-slot-7.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
------	------

`mismatch` `bool`

`firmware.component.laser-driver-module-1-slot-7.progress`

The progress of the current firmware upgrade

MODELS *All*

Access: R

Name	Type
progress	{ }
Name Type	
base	int
hints	{ }
Name Type	

`firmware.component.laser-driver-module-1-slot-7.requiredversion`

The firmware version this component is expected to have by the current firmware package.

MODELS *All*

Access: R

Name	Type
requiredversion	string

`firmware.component.laser-driver-module-1-slot-7.status`

The status of the current firmware upgrade

MODELS *All*

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

`firmware.component.laser-power-module-1-slot-8.actualversion`

The version of the currently installed firmware.

MODELS *All*

Access: R

Name	Type
actualversion	string

firmware.component.laser-power-module-1-slot-8.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.laser-power-module-1-slot-8.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.laser-power-module-1-slot-8.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.laser-power-module-1-slot-8.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

firmware.component.laser-power-module-1-slot-8.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.laser-power-module-1-slot-8.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.mainboard-power-sequencer.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.mainboard-power-sequencer.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.mainboard-power-sequencer.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.mainboard-power-sequencer.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.mainboard-power-sequencer.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.mainboard-power-sequencer.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.mainboard-power-sequencer.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.peltier-control-board-boot.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.peltier-control-board-boot.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.peltier-control-board-boot.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.peltier-control-board-boot.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.peltier-control-board-boot.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.peltier-control-board-boot.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.peltier-control-board-boot.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.peltier-control-board-run.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.peltier-control-board-run.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.peltier-control-board-run.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.peltier-control-board-run.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.peltier-control-board-run.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

firmware.component.peltier-control-board-run.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.peltier-control-board-run.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.peri-fpga.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.peri-fpga.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.peri-fpga.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.peri-fpga.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.peri-fpga.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.peri-fpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.peri-fpga.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.power-distribution-board-boot-2.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-boot-2.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-boot-2.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-boot-2.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-boot-2.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.power-distribution-board-boot-2.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-boot-2.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type				
status	enum				
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr></table>	Values	"Inactive"	"Active"	"Error"
Values					
"Inactive"					
"Active"					
"Error"					

"Finished"

firmware.component.power-distribution-board-fcb-fpga.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-fcb-fpga.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-fcb-fpga.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-fcb-fpga.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-fcb-fpga.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type		
progress	{ }		
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type		

```
base    int
hints   { }
```

Name	Type
------	------

firmware.component.power-distribution-board-fcb-fpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-fcb-fpga.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.power-distribution-board-lsr-fantable-ac.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-lsr-fantable-ac.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-lsr-fantable-ac.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-lsr-fantable-ac.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-lsr-fantable-ac.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.power-distribution-board-lsr-fantable-ac.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-lsr-fantable-ac.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum

Values
"Inactive"
"Active"
"Error"
"Finished"

firmware.component.power-distribution-board-lsr-fantable-on.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-lsr-fantable-on.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-lsr-fantable-on.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-lsr-fantable-on.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-lsr-fantable-on.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.power-distribution-board-lsr-fantable-on.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-lsr-fantable-on.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.power-distribution-board-lsr-fantable-sb.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-lsr-fantable-sb.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-lsr-fantable-sb.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-lsr-fantable-sb.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-lsr-fantable-sb.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

firmware.component.power-distribution-board-lsr-fantable-sb.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-lsr-fantable-sb.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.power-distribution-board-lsr-fantable-tc.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-lsr-fantable-tc.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-lsr-fantable-tc.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-lsr-fantable-tc.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-lsr-fantable-tc.progress

The progress of the current firmware upgrade

MODELS

All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.power-distribution-board-lsr-fantable-tc.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-lsr-fantable-tc.status

The status of the current firmware upgrade

MODELS

All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.power-distribution-board-mcb-fpga.actualversion

The version of the currently installed firmware.

MODELS

All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-mcb-fpga.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-mcb-fpga.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-mcb-fpga.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
mismatch	bool

firmware.component.power-distribution-board-mcb-fpga.progress

The progress of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.power-distribution-board-mcb-fpga.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

 All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-mcb-fpga.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.power-distribution-board-run-2.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.power-distribution-board-run-2.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.power-distribution-board-run-2.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.power-distribution-board-run-2.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
------	------

mismatch bool

firmware.component.power-distribution-board-run-2.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.power-distribution-board-run-2.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.power-distribution-board-run-2.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.status

The status of the current component firmware upgrade

MODELS All

Access: R

Name	Type
status	enum

Values
"Inactive"
"Active"
"Error"
"Finished"

firmware.component.switch-mode-power-supply.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.switch-mode-power-supply.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.switch-mode-power-supply.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.switch-mode-power-supply.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.switch-mode-power-supply.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.switch-mode-power-supply.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.switch-mode-power-supply.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.u-boot-img.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.u-boot-img.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.u-boot-img.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.u-boot-img.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.u-boot-img.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

firmware.component.u-boot-img.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.u-boot-img.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.u-boot-spl.actualversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
actualversion	string

firmware.component.u-boot-spl.displayname

The user-friendly name of the firmware component.

MODELS All

Access: R

Name	Type
displayname	string

firmware.component.u-boot-spl.error

The error message in case of an upgrade error.

MODELS All

Access: R

Name	Type
error	string

firmware.component.u-boot-spl.mismatch

Indicate whether there is a version mismatch

MODELS All

Access: R

Name	Type
mismatch	bool

firmware.component.u-boot-spl.progress

The progress of the current firmware upgrade

MODELS

All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.u-boot-spl.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

All

Access: R

Name	Type
requiredversion	string

firmware.component.u-boot-spl.status

The status of the current firmware upgrade

MODELS

All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.component.wobulator-boot.actualversion

The version of the currently installed firmware.

MODELS

All

Access: R

Name	Type
actualversion	string

firmware.component.wobulator-boot.displayname

The user-friendly name of the firmware component.

MODELS

All

Access: R

Name	Type
displayname	string

firmware.component.wobulator-boot.error

The error message in case of an upgrade error.

MODELS

All

Access: R

Name	Type
error	string

firmware.component.wobulator-boot.mismatch

Indicate whether there is a version mismatch

MODELS

All

Access: R

Name	Type
mismatch	bool

firmware.component.wobulator-boot.progress

The progress of the current firmware upgrade

MODELS

All

Access: R

Name	Type
progress	{ }
Name	Type
base	int
hints	{ }
Name	Type

firmware.component.wobulator-boot.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS

All

Access: R

Name	Type
requiredversion	string

firmware.component.wobulator-boot.status

The status of the current firmware upgrade

MODELS

 All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

firmware.component.wobulator-run.actualversion

The version of the currently installed firmware.

MODELS

 All

Access: R

Name	Type
actualversion	string

firmware.component.wobulator-run.displayname

The user-friendly name of the firmware component.

MODELS

 All

Access: R

Name	Type
displayname	string

firmware.component.wobulator-run.error

The error message in case of an upgrade error.

MODELS

 All

Access: R

Name	Type
error	string

firmware.component.wobulator-run.mismatch

Indicate whether there is a version mismatch

MODELS

 All

Access: R

Name	Type
------	------

mismatch bool

firmware.component.wobulator-run.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type						
progress	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

firmware.component.wobulator-run.requiredversion

The firmware version this component is expected to have by the current firmware package.

MODELS All

Access: R

Name	Type
requiredversion	string

firmware.component.wobulator-run.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type					
status	enum					
	<table><tr><th>Values</th></tr><tr><td>"Inactive"</td></tr><tr><td>"Active"</td></tr><tr><td>"Error"</td></tr><tr><td>"Finished"</td></tr></table>	Values	"Inactive"	"Active"	"Error"	"Finished"
Values						
"Inactive"						
"Active"						
"Error"						
"Finished"						

firmware.error

The error of the last firmware upgrade (only set if status is error)

MODELS All

Access: R

Name	Type
error	string

firmware.eula.requiretermagreement

Indiates whether or not the user is required to accept the term agreement.

MODELS All

Access: R

Name	Type
requiretermagreement	bool

firmware.eula.termagreement

The user agreement with the firmware EULA.

MODELS All

Access: RW

Name	Type
termagreement	bool

firmware.firmwareversion

The version of the currently installed firmware.

MODELS All

Access: R

Name	Type
firmwareversion	string

firmware.progress

The progress of the current firmware upgrade

MODELS All

Access: R

Name	Type
progress	{ }
base	int
hints	{ }
Name	Type

firmware.status

The status of the current firmware upgrade

MODELS All

Access: R

Name	Type
status	enum
Values	
"Inactive"	
"Active"	
"Error"	
"Finished"	

gsm.data.status

The current status of the mobile connection.

MODELS All

Access: R

Name	Type
status	enum
Values	
"FAILED"	
"UNKNOWN"	
"INITIALIZING"	
"LOCKED"	
"DISABLED"	
"DISABLING"	
"ENABLING"	
"ENABLED"	
"SEARCHING"	
"REGISTERED"	
"DISCONNECTING"	
"CONNECTING"	
"CONNECTED"	

gsm.iccid

The ICCID that identifies the SIM.

MODELS All

Access: R

Name	Type
iccid	string

gsm.imei

The IMEI that identifies the GSM module.

MODELS All

Access: R

Name	Type
imei	string

gsm.pinstate

The current state of PIN

MODELS

All

Access: R

Name	Type
pinstate	enum
Values	
"Accepted"	
"Failed"	
"Locked"	
"Unknown"	

gsm.signalstrength

The signal strength in percent.

MODELS

All

Access: R

Name	Type
signalstrength	{ }
NameType	
base	int
hints	{ }
NameType	

illumination.clo.allowed

Indicates which features are preventing CLO from being enabled

MODELS

All

Access: R

Name	Type
allowed	{ }
NameType	
allowed	bool
reasons	[string]

illumination.clo.availability

Shows the current availability.

MODELS

All

Access: R

Name	Type
availability	enum
Values	
"Available"	
"SensorUnavailable"	
"PendingWarmup"	
"PendingShutdown"	
"Unavailable"	
"Unknown"	

illumination.clo.enable

True if constant light output is enabled, false otherwise

MODELS All

Access: RW

Name	Type
enable	bool

illumination.clo.scale

The percentage to scale the setpoint by.

MODELS All

Access: RW

Name	Type
scale	{ }
Name Type	
base	float
hints	{ }
Name Type	

illumination.clo.setpoint

The target luminosity of the light source

MODELS All

Access: RW

Name	Type
setpoint	{ }
Name Type	
base	float
hints	{ }
Name Type	

illumination.clo.state

State of the CLO

MODELS All

Access: R

Name	Type
state	enum
Values	
"Idle"	
"Ok"	
"TooDim"	
"TooBright"	
"Pending"	
"Error"	

illumination.laser.opticalalignment

Request optical alignment mode.

MODELS All

Access: R

Name	Type
opticalalignment	bool

illumination.laser.timcuring.cureavailable

Description not provided

MODELS All

Access: R

Name	Type
cureavailable	bool

illumination.laser.timcuring.state

State of the current or past curing execution expressed in the format of a date, a status and a list of failed plates

MODELS All

Access: R

Name	Type
state	{ }
NameType	
date	string
status	enum

Values	
"Inactive"	
"Preparing"	
"Active"	
"Cooling"	
"Success"	
"Failure"	
failures	[{ }]
	{ }
Name	Type
base	int
hints	{ }
Name	
Type	

illumination.laser.whitepoint.tracking.cancellable

Indicates whether the calibration can be cancelled

MODELS All

Access: R

Name	Type
cancellable	bool

illumination.laser.whitepoint.tracking.progress

Progress of the current calibration process (0..100)%

MODELS All

Access: R

Name	Type
progress	{ }
Name	
Type	
base	int
hints	{ }
Name	
Type	

illumination.laser.whitepoint.tracking.result

Result of the last calibration process

MODELS All

Access: R

Name	Type
result	{ }
Name	
Type	

date	string
state	enum
Values	
"Idle"	
"Busy"	
"Success"	
"Error"	

illumination.laser.whitepoint.tracking.state

State of the current calibration process

MODELS All

Access: R

Name	Type
state	enum
Values	
"Idle"	
"Busy"	
"Success"	
"Error"	

illumination.sources.laser.actualpower

actual power in percent with limits

MODELS All

Access: R

Name	Type
actualpower	{ }
Name	Type
base	float
hints	{ }
Name	Type

illumination.sources.laser.controlledbyclo

Describes whether CLO is controlling this source.

MODELS All

Access: R

Name	Type
controlledbyclo	bool

illumination.sources.laser.ispowerlimited

Whether power is currently limited.

MODELS

 All

Access: R

Name	Type
ispowerlimited	bool

illumination.sources.laser.maxpower

maximum power in percent

MODELS

 All

Access: R

Name	Type						
maxpower	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

illumination.sources.laser.minpower

minimum power in percent

MODELS

 All

Access: R

Name	Type						
minpower	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

illumination.sources.laser.power

target power in percent

MODELS

 All

Access: RW

Name	Type						
power	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

illumination.sources.laser.powerlimitreason

If power is limited, gives the reason

MODELS All

Access: R

Name	Type
powerlimitreason	string

illumination.sources.laser.status

Description not provided

MODELS All

Access: R

Name	Type
status	enum
Values	
"Off"	
"Ignite"	
"WarmingUp"	
"On"	
"Cooling"	
"Malfunctioning"	
"NotPresent"	

illumination.state

The state of light

MODELS All

Access: R

Name	Type
state	enum
Values	
"On"	
"Off"	

image.blackcontentdetection.dimminginterval

DEPRECATED: Interval in milliseconds during which the power dimming is done upon black content detection.

MODELS All

Access: RW

Name	Type
------	------

dimminginterval	int
Constraints	
Minimum	0
Maximum	3000
Step size	10
Precision	1

image.blackcontentdetection.enable

DEPRECATED: Enable/disable black content detection.

MODELS

 All

Access: RW

Name	Type
enable	bool

image.blackcontentdetection.sampleinterval

DEPRECATED: Sample interval in milliseconds for black content detection.

MODELS

 All

Access: RW

Name	Type
sampleinterval	int
Constraints	
Minimum	500
Maximum	3000
Step size	10
Precision	1

image.blackcontentdetection.state

DEPRECATED: State true/false means black content was/was not detected.

MODELS

 All

Access: R

Name	Type
state	bool

image.blackcontentdetection.threshold

DEPRECATED: The offset seen from the signal range lowest value, which specifies the black level threshold for the detector.

MODELS

 All

Access: RW

Name	Type
threshold	int
Constraints	
Minimum	0
Maximum	255
Step size	1
Precision	1

image.brightness

Image brightness/offset. The value is normalized, 0 is default, 1 is 100% offset.

MODELS All

Access: RW

Name	Type
brightness	float
Constraints	
Minimum	-1
Maximum	1
Step size	1
Precision	0.01

image.capture.corrected

Indicates whether or not the captured images are corrected or not.

MODELS All

Access: R

Name	Type
corrected	bool

image.capture.enable

Enable/Disable capturemode, which facilitates capturing the projected image

MODELS All

Access: RW

Name	Type
enable	bool

image.color.p7.custom.bluegain

Desired blue gain value

MODELS All

Access: RW

Name	Type
bluegain	{ }
Name	Type
base	float
hints	{ }
Name	Type

image.color.p7.custom.bluelum

Desired blue luminanace

MODELS All

Access: RW

Name	Type
bluelum	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.bluex

Desired blue x in xy-coordinates

MODELS All

Access: RW

Name	Type
bluex	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.bluey

Desired blue y in xy-coordinates

MODELS All

Access: RW

Name	Type
bluey	float
Constraints	
Minimum	0.0001

Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.cmyreadable

true if secondaries should be shown (OSD)

MODELS All

Access: R

Name	Type
cmyreadable	bool

image.color.p7.custom.cmywritable

true if secondaries are Writable

MODELS All

Access: R

Name	Type
cmywritable	bool

image.color.p7.custom.cyangain

Desired cyan gain value

MODELS All

Access: RW

Name	Type						
cyangain	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.color.p7.custom.cyanlum

Desired cyan luminanace

MODELS All

Access: RW

Name	Type						
cyanlum	float						
<table><tr><th colspan="2">Constraints</th></tr><tr><td>Minimum</td><td>0.0001</td></tr><tr><td>Maximum</td><td>1</td></tr></table>		Constraints		Minimum	0.0001	Maximum	1
Constraints							
Minimum	0.0001						
Maximum	1						

Step size	0.0001
Precision	1

image.color.p7.custom.cyanx

Desired cyan x in xy-coordinates

MODELS

 All

Access: RW

Name	Type
cyanx	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.cyany

Desired cyan y in xy-coordinates

MODELS

 All

Access: RW

Name	Type
cyany	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.gainsavailable

true when gains are available

MODELS

 All

Access: R

Name	Type
gainsavailable	bool

image.color.p7.custom.greengain

Desired green gain value

MODELS

 All

Access: RW

Name	Type
greengain	{ }
Name	Type
base	float
hints	{ }
Name	Type

image.color.p7.custom.greenlum

Desired green luminanace

MODELS All

Access: RW

Name	Type
greenlum	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.greenx

Desired green x in xy-coordinates

MODELS All

Access: RW

Name	Type
greenx	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.greeny

Desired green y in xy-coordinates

MODELS All

Access: RW

Name	Type
greeny	float
Constraints	
Minimum	0.0001

Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.luminancesavailable

true if luminances are available

MODELS All

Access: R

Name	Type
luminancesavailable	bool

image.color.p7.custom.magentagain

Desired magenta gain value

MODELS All

Access: RW

Name	Type						
magentagain	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.color.p7.custom.magentalum

Desired magenta luminanace

MODELS All

Access: RW

Name	Type										
magentalum	float										
<table><tr><th colspan="2">Constraints</th></tr><tr><td>Minimum</td><td>0.0001</td></tr><tr><td>Maximum</td><td>1</td></tr><tr><td>Step size</td><td>0.0001</td></tr><tr><td>Precision</td><td>1</td></tr></table>		Constraints		Minimum	0.0001	Maximum	1	Step size	0.0001	Precision	1
Constraints											
Minimum	0.0001										
Maximum	1										
Step size	0.0001										
Precision	1										

image.color.p7.custom.magentax

Desired magenta x in xy-coordinates

MODELS All

Access: RW

Name	Type
magentax	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.magentay

Desired magenta y in xy-coordinates

MODELS

All

Access: RW

Name	Type
magentay	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.mode

Description not provided

MODELS

All

Access: RW

Name	Type
mode	string

image.color.p7.custom.modes

Description not provided

MODELS

All

Access: R

Name	Type				
modes	[{ }]				
{ }					
<table><tr><th>Name</th><th>Type</th></tr><tr><td>group</td><td>enum</td></tr></table>		Name	Type	group	enum
Name	Type				
group	enum				
Values					
"Custom"					
"Preset"					

`modes [ string ]`

image.color.p7.custom.redgain

Desired red gain value

MODELS All

Access: RW

Name	Type
redgain	{ }
Name	Type
base	float
hints	{ }
Name	Type

image.color.p7.custom.redlum

Desired red luminanace

MODELS All

Access: RW

Name	Type
redlum	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.redx

Desired red x in xy-coordinates

MODELS All

Access: RW

Name	Type
redx	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.redy

Desired red y in xy-coordinates

MODELS

All

Access: RW

Name	Type
redy	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.rgbcmygainsavailable

true when R,G,B,C,M,Y gains are available

MODELS

All

Access: R

Name	Type
rgbcmygainsavailable	bool

image.color.p7.custom.rgbwritable

true if RGB are Writable (not in WHITE mode)

MODELS

All

Access: R

Name	Type
rgbwritable	bool

image.color.p7.custom.target

Target color coordinates for the current preset

MODELS

All

Access: R

Name	Type
target	{ }
Name Type red { }	
Name Type x { }	
Name Type base float Name Type hints { }	
Name Type	

	y	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
			Name	Type						
			base	float						
			hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									
green	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }		
			Name	Type						
			x	{ }						
			<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type									
base	float									
hints	{ }									
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									
	y	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
			Name	Type						
			base	float						
			hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									
blue	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }		
			Name	Type						
			x	{ }						
			<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type									
base	float									
hints	{ }									
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									
	y	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
			Name	Type						
			base	float						
			hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									
white	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }		
			Name	Type						
			x	{ }						
			<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type									
base	float									
hints	{ }									
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									
	y	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
			Name	Type						
			base	float						
			hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type							
Name	Type									

image.color.p7.custom.whitegain

Desired white gain value

MODELS

 All

Access: RW

Name	Type
whitegain	{ }
Name	Type
base	float
hints	{ }
Name	Type

image.color.p7.custom.whitegainavailable

true when white gain is available

MODELS

 All

Access: R

Name	Type
whitegainavailable	bool

image.color.p7.custom.whitelum

Desired white luminanace

MODELS

 All

Access: RW

Name	Type
whitelum	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.whitemode

Description not provided

MODELS

 All

Access: RW

Name	Type
whitemode	enum
Values	
"Coordinates"	
"Temperature"	

image.color.p7.custom.whitetemperature

Desired white point temperature

MODELS All

Access: RW

Name	Type
whitetemperature	int
Constraints	
Minimum	3200
Maximum	13000
Step size	100
Precision	1

image.color.p7.custom.whitetemperatureavailable

true if White temperature is available

MODELS All

Access: R

Name	Type
whitetemperatureavailable	bool

image.color.p7.custom.whitewritable

true if White is Writable

MODELS All

Access: R

Name	Type
whitewritable	bool

image.color.p7.custom.whitex

Desired white x in xy-coordinates

MODELS All

Access: RW

Name	Type
whitex	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.whitey

Desired white y in xy-coordinates

MODELS All

Access: RW

Name	Type
whitey	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.yellowgain

Desired yellow gain value

MODELS All

Access: RW

Name	Type
yellowgain	{ }
NameType	
base	float
hints	{ }
NameType	

image.color.p7.custom.yellowlum

Desired yellow luminanace

MODELS All

Access: RW

Name	Type
yellowlum	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.yellowx

Desired yellow x in xy-coordinates

MODELS All

Access: RW

Name	Type
yellowx	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.custom.yellowy

Desired yellow y in xy-coordinates

MODELS All

Access: RW

Name	Type
yellowy	float
Constraints	
Minimum	0.0001
Maximum	1
Step size	0.0001
Precision	1

image.color.p7.native.c1

Native C1 x in xy-coordinates

MODELS All

Access: R

Name	Type
c1	{ }
Name Type	
x	{ }
Name Type	
base	float
hints	{ }
Name Type	
y	{ }
Name Type	
base	float
hints	{ }
Name Type	
lum	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.color.p7.native.c1available

Description not provided

MODELS All

Access: R

Name	Type
c1available	bool

image.color.p7.native.c2

Native C2 x in xy-coordinates

MODELS All

Access: R

Name	Type																																															
c2	{ }																																															
<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }																																											
Name	Type																																															
x	{ }																																															
<table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>y</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lum</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table></td></tr></table>				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type	y	{ }	<table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lum</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type	lum	{ }	<table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																																								
Name	Type																																															
base	float																																															
hints	{ }																																															
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type																																													
Name	Type																																															
y	{ }																																															
<table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lum</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type	lum	{ }	<table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type																	
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																																								
Name	Type																																															
base	float																																															
hints	{ }																																															
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type																																													
Name	Type																																															
lum	{ }																																															
<table><tr><td></td><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type																																		
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																																								
Name	Type																																															
base	float																																															
hints	{ }																																															
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type																																													
Name	Type																																															

image.color.p7.native.c2available

Description not provided

MODELS All

Access: R

Name	Type
c2available	bool

image.color.p7.native.list

list available native sets

MODELS

All

Access: R

Name	Type
list	[string]

image.color.p7.native.normal.c1

Native C1 x in xy-coordinates

MODELS

All

Access: R

Name	Type																																				
c1	{ }																																				
<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>y</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>lum</td><td>{ }</td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		Name	Type	x	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }	y	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }	lum	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
Name	Type																																				
x	{ }																																				
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }																														
Name	Type																																				
base	float																																				
hints	{ }																																				
y	{ }																																				
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }																														
Name	Type																																				
base	float																																				
hints	{ }																																				
lum	{ }																																				
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }																														
Name	Type																																				
base	float																																				
hints	{ }																																				
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type																																		
Name	Type																																				

image.color.p7.native.normal.c1available

Description not provided

MODELS

All

Access: R

Name	Type
c1available	bool

image.color.p7.native.normal.c2

Native C2 x in xy-coordinates

MODELS All

Access: R

Name	Type
c2	{ }

Name	Type
x	{ }

Name	Type
base	float
hints	{ }

Name	Type
y	{ }

Name	Type
base	float
hints	{ }

Name	Type
lum	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.color.p7.native.normal.c2available

Description not provided

MODELS All

Access: R

Name	Type
c2available	bool

image.color.p7.native.normal.rgbw

Native red x in xy-coordinates

MODELS All

Access: R

Name	Type
rgbw	{ }

Name	Type
red	{ }

Name Type	
x	{ }
Name Type	
base	float
hints	{ }
Name Type	
y	{ }
Name Type	
base	float
hints	{ }
Name Type	
lum	{ }
Name Type	
base	float
hints	{ }
Name Type	
green	{ }
Name Type	
x	{ }
Name Type	
base	float
hints	{ }
Name Type	
y	{ }
Name Type	
base	float
hints	{ }
Name Type	
lum	{ }
Name Type	
base	float
hints	{ }
Name Type	
blue	{ }
Name Type	
x	{ }
Name Type	
base	float
hints	{ }
Name Type	
y	{ }
Name Type	

		<table><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	base	float	hints	{ }		
base	float							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
lum	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
white	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>x</td><td>{ }</td></tr></table>	Name	Type	x	{ }		
Name	Type							
x	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
y	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
lum	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							

image.color.p7.native.normal.whiteavailable

Description not provided

MODELS All

Access: R

Name	Type
whiteavailable	bool

image.color.p7.native.rgbw

Native red x in xy-coordinates

MODELS All

Access: R

Name	Type
rgbw	{ }
Name	Type

red { }							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }		
Name	Type						
x	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>y</td><td>{ }</td></tr></table>		Name	Type	y	{ }		
Name	Type						
y	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>lum</td><td>{ }</td></tr></table>		Name	Type	lum	{ }		
Name	Type						
lum	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>green</td><td>{ }</td></tr></table>		Name	Type	green	{ }		
Name	Type						
green	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }		
Name	Type						
x	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>y</td><td>{ }</td></tr></table>		Name	Type	y	{ }		
Name	Type						
y	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>lum</td><td>{ }</td></tr></table>		Name	Type	lum	{ }		
Name	Type						
lum	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>blue</td><td>{ }</td></tr></table>		Name	Type	blue	{ }		
Name	Type						
blue	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }		
Name	Type						
x	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>y</td><td>{ }</td></tr></table>		Name	Type	y	{ }		
Name	Type						
y	{ }						

[illegible]

image.color.p7.native.whiteavailable

Description not provided

MODELS All

Access: R

Name	Type
whiteavailable	bool

```
image.color.rgbmode.rgbmode
```

RGB Mode

MODELS All

Access: RW

Name	Type
rgbmode	enum

Values
"Full"
"Red"
"Green"
"Blue"
"RedGreen"
"GreenBlue"
"BlueRed"

image.connector.l1displayport.capabilities

Capabilities.

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values "HDCP" "HDCP2"	

image.connector.l1displayport.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorprimaries	enum
Values "Auto" "Uncorrected" "REC709" "REC2020" "DCI-P3-D65" "DCI-P3-Theater"	

image.connector.l1displayport.colorprimariesavailable

true if Color Primaries is available

MODELS All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1displayport.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	
"REC709"	
"REC601"	
"REC2020"	

image.connector.l1displayport.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS

 All

Access: R

Name	Type								
detectedsignal	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr></table>		Name	Type	active	bool	name	string	vertical_total	{ }
Name	Type								
active	bool								
name	string								
vertical_total	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
horizontal_total	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
vertical_resolution	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
horizontal_resolution	{ }								

vertical_sync_width	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
vertical_front_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
vertical_back_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_sync_width	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_front_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_back_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_frequency	{ }		
		base	float
		hints	{ }
		Name	Type
		base	float
		hints	{ }
		Name	Type
vertical_frequency	{ }		
		base	float
		hints	{ }
		Name	Type
		base	float
		hints	{ }
		Name	Type

		Name	Type
pixel_rate	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type
scan	enum		
		Values	
		"Progressive"	
		"Interlaced"	
bits_per_component	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type
color_space	enum		
		Values	
		"RGB"	
		"REC709"	
		"REC601"	
		"REC2020"	
signal_range	enum		
		Values	
		"0-255"	
		"16-235"	
chroma_sampling	enum		
		Values	
		"4:4:4"	
		"4:2:2"	
		"4:2:0"	
gamma_type	enum		
		Values	
		"POWER"	
		"sRGB"	
		"REC_BT1886"	
		"SIM1"	
		"SMPTE_ST2084"	
		"DICOM_10"	
		"DICOM_60"	
		"DICOM_180"	
		"DICOM_250"	
		"DICOM_300"	

color_primaries	enum	"DICOM_400"	
		Values	
		"REC709"	
		"REC2020"	
		"DCI-P3-D65"	
mastering_luminance	{ }	"DCI-P3-Theater"	
		Name	Type
		base	float
		hints	{ }
max_cll	{ }		
		Name	Type
		base	float
		hints	{ }
max_fall	{ }		
		Name	Type
		base	float
		hints	{ }
content_aspect_ratio	enum		
		Values	
		"5:4"	
		"4:3"	
		"16:10"	
		"16:9"	
		"1.85:1"	
		"1.88:1"	
		"1.9:1"	
		"2.20:1"	
		"2.00:1"	
		"2.35:1"	
		"2.37:1"	
		"2.39:1"	
		"Unknown"	
content_type	enum		
		Values	
		"Graphics"	
		"Photo"	
		"Cinema"	

		"Game"
		"Testpattern"
		"Unknown"
is_stereo	bool	
stereo_mode	enum	
Values		
"None"		
"Sequential"		
"FramePacked"		
"TopBottom"		
"SideBySide"		

image.connector.l1displayport.edid.selected

Selected EDID for connector

MODELS All

Access: RW

Name	Type
selected	string

image.connector.l1displayport.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.connector.l1displayport.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.connector.l1hdbaset1.capabilities

Capabilities.

MODELS

 All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1hdbaset1.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorprimaries	enum
Values	
"Auto"	
"Uncorrected"	
"REC709"	
"REC2020"	
"DCI-P3-D65"	
"DCI-P3-Theater"	

image.connector.l1hdbaset1.colorprimariesavailable

true if Color Primaries is available

MODELS

 All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1hdbaset1.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	

"REC709"

"REC601"

"REC2020"

image.connector.l1hdbaset1.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS All

Access: R

Name	Type																																																														
detectedsignal	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>horizontal_total</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_resolution</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>horizontal_resolution</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_sync_width</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_front_porch</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	active	bool	name	string	vertical_total	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	horizontal_total	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_resolution	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	horizontal_resolution	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_sync_width	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_front_porch	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type																																																														
active	bool																																																														
name	string																																																														
vertical_total	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																								
Name	Type																																																														
base	int																																																														
hints	{ }																																																														
horizontal_total	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																								
Name	Type																																																														
base	int																																																														
hints	{ }																																																														
vertical_resolution	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																								
Name	Type																																																														
base	int																																																														
hints	{ }																																																														
horizontal_resolution	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																								
Name	Type																																																														
base	int																																																														
hints	{ }																																																														
vertical_sync_width	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																								
Name	Type																																																														
base	int																																																														
hints	{ }																																																														
vertical_front_porch	{ }																																																														
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																												
Name	Type																																																														

		base	int		
		hints	{ }		
				Name	Type
vertical_back_porch	{ }				
		Name	Type		
		base	int		
		hints	{ }		
				Name	Type
horizontal_sync_width	{ }				
		Name	Type		
		base	int		
		hints	{ }		
				Name	Type
horizontal_front_porch	{ }				
		Name	Type		
		base	int		
		hints	{ }		
				Name	Type
horizontal_back_porch	{ }				
		Name	Type		
		base	int		
		hints	{ }		
				Name	Type
horizontal_frequency	{ }				
		Name	Type		
		base	float		
		hints	{ }		
				Name	Type
vertical_frequency	{ }				
		Name	Type		
		base	float		
		hints	{ }		
				Name	Type
pixel_rate	{ }				
		Name	Type		
		base	int		
		hints	{ }		
				Name	Type
scan	enum				
		Values			
		"Progressive"			
		"Interlaced"			

bits_per_component	{ }													
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }						
Name	Type													
base	int													
hints	{ }													
color_space	enum	<table><tr><th>Name</th><th>Type</th></tr></table> <table><tr><th>Values</th></tr><tr><td>"RGB"</td></tr><tr><td>"REC709"</td></tr><tr><td>"REC601"</td></tr><tr><td>"REC2020"</td></tr></table>	Name	Type	Values	"RGB"	"REC709"	"REC601"	"REC2020"					
Name	Type													
Values														
"RGB"														
"REC709"														
"REC601"														
"REC2020"														
signal_range	enum	<table><tr><th>Values</th></tr><tr><td>"0-255"</td></tr><tr><td>"16-235"</td></tr></table>	Values	"0-255"	"16-235"									
Values														
"0-255"														
"16-235"														
chroma_sampling	enum	<table><tr><th>Values</th></tr><tr><td>"4:4:4"</td></tr><tr><td>"4:2:2"</td></tr><tr><td>"4:2:0"</td></tr></table>	Values	"4:4:4"	"4:2:2"	"4:2:0"								
Values														
"4:4:4"														
"4:2:2"														
"4:2:0"														
gamma_type	enum	<table><tr><th>Values</th></tr><tr><td>"POWER"</td></tr><tr><td>"sRGB"</td></tr><tr><td>"REC_BT1886"</td></tr><tr><td>"SIM1"</td></tr><tr><td>"SMPTE_ST2084"</td></tr><tr><td>"DICOM_10"</td></tr><tr><td>"DICOM_60"</td></tr><tr><td>"DICOM_180"</td></tr><tr><td>"DICOM_250"</td></tr><tr><td>"DICOM_300"</td></tr><tr><td>"DICOM_400"</td></tr></table>	Values	"POWER"	"sRGB"	"REC_BT1886"	"SIM1"	"SMPTE_ST2084"	"DICOM_10"	"DICOM_60"	"DICOM_180"	"DICOM_250"	"DICOM_300"	"DICOM_400"
Values														
"POWER"														
"sRGB"														
"REC_BT1886"														
"SIM1"														
"SMPTE_ST2084"														
"DICOM_10"														
"DICOM_60"														
"DICOM_180"														
"DICOM_250"														
"DICOM_300"														
"DICOM_400"														
color_primaries	enum	<table><tr><th>Values</th></tr><tr><td>"REC709"</td></tr><tr><td>"REC2020"</td></tr><tr><td>"DCI-P3-D65"</td></tr><tr><td>"DCI-P3-Theater"</td></tr></table>	Values	"REC709"	"REC2020"	"DCI-P3-D65"	"DCI-P3-Theater"							
Values														
"REC709"														
"REC2020"														
"DCI-P3-D65"														
"DCI-P3-Theater"														
mastering_luminance	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr></table>	Name	Type	base	float								
Name	Type													
base	float													

		hints	{ }	
				NameType
max_cll	{ }			
		Name	Type	
		base	float	
		hints	{ }	
				NameType
max_fall	{ }			
		Name	Type	
		base	float	
		hints	{ }	
				NameType
content_aspect_ratio	enum			
		Values		
		"5:4"		
		"4:3"		
		"16:10"		
		"16:9"		
		"1.85:1"		
		"1.88:1"		
		"1.9:1"		
		"2.20:1"		
		"2.00:1"		
		"2.35:1"		
		"2.37:1"		
		"2.39:1"		
		"Unknown"		
content_type	enum			
		Values		
		"Graphics"		
		"Photo"		
		"Cinema"		
		"Game"		
		"Testpattern"		
		"Unknown"		
is_stereo	bool			
stereo_mode	enum			
		Values		
		"None"		
		"Sequential"		
		"FramePacked"		
		"TopBottom"		

"SideBySide"

image.connector.l1hdbaset1.edid.selected

Selected EDID for connector

MODELS All

Access: RW

Name	Type
selected	string

image.connector.l1hdbaset1.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.connector.l1hdbaset1.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.connector.l1hdbaset2.capabilities

Capabilities.

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1hdbaset2.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorprimaries	enum
	Values
	"Auto"
	"Uncorrected"
	"REC709"
	"REC2020"
	"DCI-P3-D65"
	"DCI-P3-Theater"

image.connector.l1hdbaset2.colorprimariesavailable

true if Color Primaries is available

MODELS

 All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1hdbaset2.colorsapce

Override the detected signal color space. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorspace	enum
	Values
	"Auto"
	"RGB"
	"REC709"
	"REC601"
	"REC2020"

image.connector.l1hdbaset2.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS

 All

Access: R

Name	Type								
detectedsignal	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr></table>	Name	Type	active	bool	name	string	vertical_total	{ }
Name	Type								
active	bool								
name	string								
vertical_total	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
horizontal_total	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_resolution	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
horizontal_resolution	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_sync_width	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_front_porch	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_back_porch	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
horizontal_sync_width	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								

horizontal_front_porch	{ }	base	int	Name	Type
		hints	{ }		
horizontal_back_porch	{ }	Name	Type	Name	Type
		base	int		
		hints	{ }		
horizontal_frequency	{ }	Name	Type	Name	Type
		base	float		
		hints	{ }		
vertical_frequency	{ }	Name	Type	Name	Type
		base	float		
		hints	{ }		
pixel_rate	{ }	Name	Type	Name	Type
		base	int		
		hints	{ }		
scan	enum	Values		Name	Type
		"Progressive"			
		"Interlaced"			
bits_per_component	{ }	Name	Type	Name	Type
		base	int		
		hints	{ }		
color_space	enum	Values		Name	Type
		"RGB"			
		"REC709"			
		"REC601"			

signal_range	enum	"REC2020"
		Values
		"0-255"
chroma_sampling	enum	"16-235"
		Values
		"4:4:4"
gamma_type	enum	"4:2:2"
		"4:2:0"
		Values
		"POWER"
		"sRGB"
		"REC_BT1886"
		"SIM1"
		"SMPTE_ST2084"
		"DICOM_10"
		"DICOM_60"
		"DICOM_180"
		"DICOM_250"
color_primaries	enum	"DICOM_300"
		"DICOM_400"
		Values
		"REC709"
		"REC2020"
mastering_luminance	{ }	"DCI-P3-D65"
		"DCI-P3-Theater"
		Name
		Type
max_c11	{ }	base
		float
		hints
		{ }
max_c11	{ }	Name
		Type
		base
		float
max_c11	{ }	hints
		{ }
max_c11	{ }	Name
		Type
		base
		float

		hints	{ }	
				NameType
content_aspect_ratio	enum			
		Values		
		"5:4"		
		"4:3"		
		"16:10"		
		"16:9"		
		"1.85:1"		
		"1.88:1"		
		"1.9:1"		
		"2.20:1"		
		"2.00:1"		
		"2.35:1"		
		"2.37:1"		
		"2.39:1"		
		"Unknown"		
content_type	enum			
		Values		
		"Graphics"		
		"Photo"		
		"Cinema"		
		"Game"		
		"Testpattern"		
		"Unknown"		
is_stereo	bool			
stereo_mode	enum			
		Values		
		"None"		
		"Sequential"		
		"FramePacked"		
		"TopBottom"		
		"SideBySide"		

```
image.connector.l1hdbaset2.edid.selected
```

Selected EDID for connector

MODELS All

Access: RW

Name	Type
selected	string

image.connector.l1hdbaset2.name

Name of the source

MODELS

 All

Access: R

Name	Type
name	string

image.connector.l1hdbaset2.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.connector.l1hdmi.capabilities

Capabilities.

MODELS

 All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1hdmi.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorprimaries	enum
Values	
"Auto"	
"Uncorrected"	
"REC709"	
"REC2020"	

"DCI-P3-D65"

"DCI-P3-Theater"

image.connector.l1hdmi.colorprimariesavailable

true if Color Primaries is available

MODELS All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1hdmi.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	
"REC709"	
"REC601"	
"REC2020"	

image.connector.l1hdmi.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS All

Access: R

Name	Type								
detectedsignal	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr></table>		Name	Type	active	bool	name	string	vertical_total	{ }
Name	Type								
active	bool								
name	string								
vertical_total	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
horizontal_total	{ }								

vertical_resolution	{ }	NameType baseint hints{ }	
		NameType	
horizontal_resolution	{ }	NameType baseint hints{ }	
		NameType	
vertical_sync_width	{ }	NameType baseint hints{ }	
		NameType	
vertical_front_porch	{ }	NameType baseint hints{ }	
		NameType	
vertical_back_porch	{ }	NameType baseint hints{ }	
		NameType	
horizontal_sync_width	{ }	NameType baseint hints{ }	
		NameType	
horizontal_front_porch	{ }	NameType baseint hints{ }	
		NameType	
horizontal_back_porch	{ }	NameType baseint	

horizontal_frequency	{ }	hints	{ }
		Name Type	
		base	float
		hints	{ }
vertical_frequency	{ }	Name Type	
		base	float
		hints	{ }
		Name Type	
pixel_rate	{ }	Name Type	
		base	int
		hints	{ }
		Name Type	
scan	enum	Values	
		"Progressive"	
		"Interlaced"	
bits_per_component	{ }	base	int
		hints	{ }
		Name Type	
color_space	enum	Values	
		"RGB"	
		"REC709"	
		"REC601"	
signal_range	enum	Values	
		"0-255"	
		"16-235"	
chroma_sampling	enum	Values	
		"4:4:4"	
		"4:2:2"	
		"4:2:0"	
gamma_type	enum		

color_primaries	enum	Values	
		"POWER"	
		"sRGB"	
		"REC_BT1886"	
		"SIM1"	
		"SMPTE_ST2084"	
		"DICOM_10"	
		"DICOM_60"	
		"DICOM_180"	
		"DICOM_250"	
		"DICOM_300"	
		"DICOM_400"	
mastering_luminance	{ }	Values	
		"REC709"	
		"REC2020"	
		"DCI-P3-D65"	
		"DCI-P3-Theater"	
max_c11	{ }	NameType	
		basefloat	
		hints{ }	
		NameType	
		basefloat	
max_fall	{ }	NameType	
		basefloat	
		hints{ }	
		NameType	
		basefloat	
content_aspect_ratio	enum	Values	
		"5:4"	
		"4:3"	
		"16:10"	
		"16:9"	
		"1.85:1"	
		"1.88:1"	
		"1.9:1"	

"2.20:1"

"2.00:1"

"2.35:1"

"2.37:1"

"2.39:1"

"Unknown"

content_typeenum

Values
"Graphics"
"Photo"
"Cinema"
"Game"
"Testpattern"
"Unknown"

is_stereobool

stereo_modeenum

Values
"None"
"Sequential"
"FramePacked"
"TopBottom"
"SideBySide"

image.connector.l1hdmi.edid.selected

Selected EDID for connector

MODELSAll

Access: RW

Name	Type
selected	string

image.connector.l1hdmi.name

Name of the source

MODELSAll

Access: R

Name	Type
name	string

image.connector.l1hdmi.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELSAll

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.connector.l1sdia.capabilities

Capabilities.

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1sdia.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorprimaries	enum
Values	
"Auto"	
"Uncorrected"	
"REC709"	
"REC2020"	
"DCI-P3-D65"	
"DCI-P3-Theater"	

image.connector.l1sdia.colorprimariesavailable

true if Color Primaries is available

MODELS All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1sdia.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	
"REC709"	
"REC601"	
"REC2020"	

image.connector.l1sdia.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS

 All

Access: R

Name	Type								
detectedsignal	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr></table>		Name	Type	active	bool	name	string	vertical_total	{ }
Name	Type								
active	bool								
name	string								
vertical_total	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
horizontal_total	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
vertical_resolution	{ }								
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
horizontal_resolution	{ }								

vertical_sync_width	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
vertical_front_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
vertical_back_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_sync_width	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_front_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_back_porch	{ }		
		base	int
		hints	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
horizontal_frequency	{ }		
		base	float
		hints	{ }
		Name	Type
		base	float
		hints	{ }
		Name	Type
vertical_frequency	{ }		
		base	float
		hints	{ }
		Name	Type
		base	float
		hints	{ }
		Name	Type

		Name	Type
pixel_rate	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type
scan	enum		
		Values	
		"Progressive"	
		"Interlaced"	
bits_per_component	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type
color_space	enum		
		Values	
		"RGB"	
		"REC709"	
		"REC601"	
		"REC2020"	
signal_range	enum		
		Values	
		"0-255"	
		"16-235"	
chroma_sampling	enum		
		Values	
		"4:4:4"	
		"4:2:2"	
		"4:2:0"	
gamma_type	enum		
		Values	
		"POWER"	
		"sRGB"	
		"REC_BT1886"	
		"SIM1"	
		"SMPTE_ST2084"	
		"DICOM_10"	
		"DICOM_60"	
		"DICOM_180"	
		"DICOM_250"	
		"DICOM_300"	

color_primaries	enum	"DICOM_400"	
		Values	
		"REC709"	
		"REC2020"	
		"DCI-P3-D65"	
mastering_luminance	{ }	"DCI-P3-Theater"	
		Name	Type
		base	float
		hints	{ }
max_c11	{ }		
		Name	Type
		base	float
		hints	{ }
max_f11	{ }		
		Name	Type
		base	float
		hints	{ }
content_aspect_ratio	enum		
		Values	
		"5:4"	
		"4:3"	
		"16:10"	
		"16:9"	
		"1.85:1"	
		"1.88:1"	
		"1.9:1"	
		"2.20:1"	
		"2.00:1"	
		"2.35:1"	
		"2.37:1"	
		"2.39:1"	
		"Unknown"	
content_type	enum		
		Values	
		"Graphics"	
		"Photo"	
		"Cinema"	

		"Game"
		"Testpattern"
		"Unknown"
is_stereo	bool	
stereo_mode	enum	
Values		
"None"		
"Sequential"		
"FramePacked"		
"TopBottom"		
"SideBySide"		

image.connector.l1sdia.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.connector.l1sdia.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.connector.l1sdib.capabilities

Capabilities.

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1sdib.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorprimaries	enum
Values	
"Auto"	
"Uncorrected"	
"REC709"	
"REC2020"	
"DCI-P3-D65"	
"DCI-P3-Theater"	

image.connector.l1sdib.colorprimariesavailable

true if Color Primaries is available

MODELS All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1sdib.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	
"REC709"	
"REC601"	
"REC2020"	

image.connector.l1sdib.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS All

Access: R

Name	Type								
detectedsignal	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr></table>	Name	Type	active	bool	name	string	vertical_total	{ }
Name	Type								
active	bool								
name	string								
vertical_total	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
horizontal_total	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_resolution	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
horizontal_resolution	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_sync_width	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_front_porch	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
vertical_back_porch	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								
horizontal_sync_width	{ }								

		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
horizontal_front_porch	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
horizontal_back_porch	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
horizontal_frequency	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
vertical_frequency	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
pixel_rate	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
scan	enum							
		<table><tr><th>Values</th></tr><tr><td>"Progressive"</td></tr><tr><td>"Interlaced"</td></tr></table>	Values	"Progressive"	"Interlaced"			
Values								
"Progressive"								
"Interlaced"								
bits_per_component	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
color_space	enum							
		<table><tr><th>Values</th></tr><tr><td>"RGB"</td></tr><tr><td>"REC709"</td></tr></table>	Values	"RGB"	"REC709"			
Values								
"RGB"								
"REC709"								

signal_range	enum	"REC601"	
		"REC2020"	
		Values	
		"0-255"	
chroma_sampling	enum	"16-235"	
		Values	
		"4:4:4"	
		"4:2:2"	
gamma_type	enum	"4:2:0"	
		Values	
		"POWER"	
		"sRGB"	
color_primaries	enum	"REC_BT1886"	
		"SIM1"	
		"SMPTE_ST2084"	
		"DICOM_10"	
		"DICOM_60"	
		"DICOM_180"	
		"DICOM_250"	
		"DICOM_300"	
		"DICOM_400"	
		Values	
mastering_luminance	{ }	"REC709"	
		"REC2020"	
		"DCI-P3-D65"	
		"DCI-P3-Theater"	
max_c11	{ }	Name	Type
		base	float
		hints	{ }
		Name Type	
max_fall	{ }	Name	Type
		base	float
		hints	{ }
		Name Type	
		Name	Type

		base float
		hints { }
		Name Type
content_aspect_ratio	enum	Values "5:4" "4:3" "16:10" "16:9" "1.85:1" "1.88:1" "1.9:1" "2.20:1" "2.00:1" "2.35:1" "2.37:1" "2.39:1" "Unknown"
content_type	enum	Values "Graphics" "Photo" "Cinema" "Game" "Testpattern" "Unknown"
is_stereo	bool	
stereo_mode	enum	Values "None" "Sequential" "FramePacked" "TopBottom" "SideBySide"

image.connector.l1sdib.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.connector.l1sdib.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.connector.l1sdic.capabilities

Capabilities.

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1sdic.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorprimaries	enum
Values	
"Auto"	
"Uncorrected"	
"REC709"	
"REC2020"	
"DCI-P3-D65"	
"DCI-P3-Theater"	

image.connector.l1sdic.colorprimariesavailable

true if Color Primaries is available

MODELS All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1sdic.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	
"REC709"	
"REC601"	
"REC2020"	

image.connector.l1sdic.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS All

Access: R

Name	Type
detectedsignal	{ }
Name Type active bool name string vertical_total { } Name Type base int hints { } Name Type horizontal_total { }	
Name Type base int hints { } Name Type vertical_resolution { }	
Name Type base int	

horizontal_resolution	{ }	hints	{ }		
				Name	Type
vertical_sync_width	{ }				
				Name	Type
vertical_front_porch	{ }				
				Name	Type
vertical_back_porch	{ }				
				Name	Type
horizontal_sync_width	{ }				
				Name	Type
horizontal_front_porch	{ }				
				Name	Type
horizontal_back_porch	{ }				
				Name	Type
horizontal_frequency	{ }				
				Name	Type

vertical_frequency	{ }		
		Name	Type
		base	float
		hints	{ }
		Name	Type
pixel_rate	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type
scan	enum		
		Values	
		"Progressive"	
		"Interlaced"	
bits_per_component	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type
color_space	enum		
		Values	
		"RGB"	
		"REC709"	
signal_range	enum		
		Values	
		"0-255"	
		"16-235"	
chroma_sampling	enum		
		Values	
		"4:4:4"	
		"4:2:2"	
gamma_type	enum		
		Values	
		"POWER"	
		"sRGB"	
		"REC_BT1886"	
		"SIM1"	
		"SMPTE_ST2084"	
		"DICOM_10"	

		"DICOM_60"						
		"DICOM_180"						
		"DICOM_250"						
		"DICOM_300"						
		"DICOM_400"						
color_primaries	enum	Values "REC709" "REC2020" "DCI-P3-D65" "DCI-P3-Theater"						
mastering_luminance	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
max_c11	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
max_fall	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
content_aspect_ratio	enum	Values "5:4" "4:3" "16:10" "16:9" "1.85:1" "1.88:1" "1.9:1" "2.20:1" "2.00:1" "2.35:1" "2.37:1" "2.39:1" "Unknown"						
content_type	enum							

		Values "Graphics" "Photo" "Cinema" "Game" "Testpattern" "Unknown"
is_stereo	bool	
stereo_mode	enum	Values "None" "Sequential" "FramePacked" "TopBottom" "SideBySide"

image.connector.l1sdic.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.connector.l1sdic.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
signalrange	enum
Values "Auto" "0-255" "16-235"	

image.connector.l1sdid.capabilities

Capabilities.

MODELS All

Access: R

Name	Type
------	------

capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.connector.l1sdid.colorprimaries

Override the detected signal color primaries. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorprimaries	enum
Values	
"Auto"	
"Uncorrected"	
"REC709"	
"REC2020"	
"DCI-P3-D65"	
"DCI-P3-Theater"	

image.connector.l1sdid.colorprimariesavailable

true if Color Primaries is available

MODELS All

Access: R

Name	Type
colorprimariesavailable	bool

image.connector.l1sdid.colorspace

Override the detected signal color space. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
colorspace	enum
Values	
"Auto"	
"RGB"	
"REC709"	
"REC601"	
"REC2020"	

image.connector.l1sdid.detectedsignal

The signal information of the currently detected signal. If 'active' is false, there is no detected signal and the rest of the information should be disregarded. is_stereo indicates if stereo_mode is different from none.

MODELS

All

Access: R

Name	Type																																																																								
detectedsignal	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>active</td><td>bool</td></tr><tr><td>name</td><td>string</td></tr><tr><td>vertical_total</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>horizontal_total</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_resolution</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>horizontal_resolution</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_sync_width</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_front_porch</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>vertical_back_porch</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	active	bool	name	string	vertical_total	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	horizontal_total	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_resolution	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	horizontal_resolution	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_sync_width	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_front_porch	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	vertical_back_porch	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type																																																																								
active	bool																																																																								
name	string																																																																								
vertical_total	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																																		
Name	Type																																																																								
base	int																																																																								
hints	{ }																																																																								
horizontal_total	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																																		
Name	Type																																																																								
base	int																																																																								
hints	{ }																																																																								
vertical_resolution	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																																		
Name	Type																																																																								
base	int																																																																								
hints	{ }																																																																								
horizontal_resolution	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																																		
Name	Type																																																																								
base	int																																																																								
hints	{ }																																																																								
vertical_sync_width	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																																		
Name	Type																																																																								
base	int																																																																								
hints	{ }																																																																								
vertical_front_porch	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																																		
Name	Type																																																																								
base	int																																																																								
hints	{ }																																																																								
vertical_back_porch	{ }																																																																								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																																						
Name	Type																																																																								

horizontal_sync_width	{ }	base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
horizontal_front_porch	{ }	base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
horizontal_back_porch	{ }	base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
horizontal_frequency	{ }	base float	
		hints { }	
		Name Type	
		base float	
		hints { }	
		Name Type	
		base float	
		hints { }	
		Name Type	
		base float	
vertical_frequency	{ }	base float	
		hints { }	
		Name Type	
		base float	
		hints { }	
		Name Type	
		base float	
		hints { }	
		Name Type	
		base float	
pixel_rate	{ }	base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
scan	enum	base enum	
		hints { }	
		Name Type	
		base enum	
		hints { }	
		Name Type	
		base enum	
		hints { }	
		Name Type	
		base enum	
bits_per_component	{ }	base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	
		hints { }	
		Name Type	
		base int	

color_space	enum	Values "RGB" "REC709" "REC601" "REC2020"
signal_range	enum	Values "0-255" "16-235"
chroma_sampling	enum	Values "4:4:4" "4:2:2" "4:2:0"
gamma_type	enum	Values "POWER" "sRGB" "REC_BT1886" "SIM1" "SMPTE_ST2084" "DICOM_10" "DICOM_60" "DICOM_180" "DICOM_250" "DICOM_300" "DICOM_400"
color_primaries	enum	Values "REC709" "REC2020" "DCI-P3-D65" "DCI-P3-Theater"
mastering_luminance	{ }	Name Type base float hints { }
max_cll	{ }	Name Type base float

max_fall	{ }	hints	{ }				
		<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type		
Name	Type						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						
content_aspect_ratio	enum	<table><tr><th>Values</th></tr></table>		Values			
Values							
"5:4"							
"4:3"							
"16:10"							
"16:9"							
"1.85:1"							
"1.88:1"							
"1.9:1"							
"2.20:1"							
"2.00:1"							
"2.35:1"							
"2.37:1"							
"2.39:1"							
"Unknown"							
content_type	enum	<table><tr><th>Values</th></tr></table>		Values			
Values							
"Graphics"							
"Photo"							
"Cinema"							
"Game"							
"Testpattern"							
"Unknown"							
is_stereo	bool						
stereo_mode	enum	<table><tr><th>Values</th></tr></table>		Values			
Values							
"None"							
"Sequential"							
"FramePacked"							
"TopBottom"							
"SideBySide"							

image.connector.l1sdid.name

Name of the source

Access: R

Name	Type
name	string

image.connector.l1sdid.signalrange

Override the detected signal signal range. Set to Auto for automatic control.

MODELS All

Access: RW

Name	Type
signalrange	enum
Values	
"Auto"	
"0-255"	
"16-235"	

image.contrast

Image contrast/gain. The value is normalized, 1 is default.

MODELS All

Access: RW

Name	Type
contrast	float
Constraints	
Minimum	0
Maximum	2
Step size	1
Precision	0.01

image.convergence.blue

Horizontal and vertical convergence offsets for blue: -2..2

MODELS All

Access: RW

Name	Type
blue	{ }
NameType	
X	{ }
NameType	
base	int
hints	{ }

		Name	Type
Y		{ }	
		Name	Type
		base	int
		hints	{ }
		Name	Type

image.convergence.green

Horizontal and vertical convergence offsets for green: -2..2

MODELS All

Access: RW

Name	Type
green	{ }
Name	Type
X	{ }
Name	Type
base	int
hints	{ }
Name	Type
Y	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.convergence.red

Horizontal and vertical convergence offsets for red: -2..2

MODELS All

Access: RW

Name	Type		
red	{ }		
		Name	Type
		X	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
Y		{ }	
		Name	Type
		base	int

hints { }

Name Type

image.display.desireddisplaymode

The desired display mode.

MODELS All

Access: RW

Name	Type
desireddisplaymode	enum
Values	
"Mono"	
"AutoStereo"	
"ActiveStereo"	
"NightVision"	
"IGPixelShift"	
"IGPixelShiftNV"	
"IGPixelShiftFullNV"	

image.display.displaymode

The current display mode.

MODELS All

Access: R

Name	Type
displaymode	enum
Values	
"Mono"	
"AutoStereo"	
"ActiveStereo"	
"NightVision"	
"IGPixelShift"	
"IGPixelShiftNV"	
"IGPixelShiftFullNV"	

image.display.frequency

The display frequency.

MODELS All

Access: R

Name	Type
frequency	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.display.synchronouslock

The display synchronous lock state.

MODELS All

Access: R

Name	Type
synchronouslock	bool

image.dynamiccontrast.level

Amount of dynamic contrast that will be applied [0:4]

MODELS All

Access: RW

Name	Type
level	enum
Values	
"OFF"	
"LOW"	
"MEDIUM"	
"HIGH"	
"MAX"	

image.gamma

Image gamma. Default is 2.2

MODELS All

Access: RW

Name	Type
gamma	float
Constraints	
Minimum	1
Maximum	3
Step size	1
Precision	0.1

image.gammatype

Override the detected signal gamma type. Set to Auto for automatic control.

MODELS

 All

Access: RW

Name	Type
gammatype	string

image.intensity

Intensity

MODELS

 All

Access: RW

Name	Type
intensity	float
Constraints	
Minimum	0
Maximum	1
Step size	1
Precision	0.01

image.orientation

Description not provided

MODELS

 All

Access: RW

Name	Type
orientation	enum
Values	
"DESKTOP_FRONT"	
"DESKTOP_REAR"	
"CEILING_FRONT"	
"CEILING_REAR"	

image.orientationmode

The mode of orientation as auto detected or a manual fixed setting

MODELS

 All

Access: RW

Name	Type
orientationmode	enum
Values	
"MANUAL "	
"AUTO_FRONT "	

"AUTO_REAR"

image.processing.blacklevel.basicblacklevel.bottom

Bottom edge.

MODELS All

Access: RW

Name	Type						
bottom	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.processing.blacklevel.basicblacklevel.enable

Description not provided

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.blacklevel.basicblacklevel.left

Left edge.

MODELS All

Access: RW

Name	Type						
left	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.processing.blacklevel.basicblacklevel.level

Change the upper level of the black level adjustment

MODELS All

Access: RW

Name	Type
level	int
Constraints	

Minimum	0
Maximum	65535
Step size	1
Precision	1

image.processing.blacklevel.basicblacklevel.right

Right edge.

MODELS All

Access: RW

Name	Type						
right	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.processing.blacklevel.basicblacklevel.top

Top edge.

MODELS All

Access: RW

Name	Type						
top	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.processing.blacklevel.bluegain

The gain blue for black level

MODELS All

Access: RW

Name	Type										
bluegain	float										
<table><tr><th colspan="2">Constraints</th></tr><tr><td>Minimum</td><td>0</td></tr><tr><td>Maximum</td><td>1</td></tr><tr><td>Step size</td><td>1</td></tr><tr><td>Precision</td><td>0.001</td></tr></table>		Constraints		Minimum	0	Maximum	1	Step size	1	Precision	0.001
Constraints											
Minimum	0										
Maximum	1										
Step size	1										
Precision	0.001										

image.processing.blacklevel.file.enable

Enable/Disable black level correction

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.blacklevel.file.selected

Currently selected file

MODELS All

Access: RW

Name	Type
selected	string

image.processing.blacklevel.greengain

The gain green for black level

MODELS All

Access: RW

Name	Type
greengain	float
Constraints	
Minimum	0
Maximum	1
Step size	1
Precision	0.001

image.processing.blacklevel.linkrgb

When enabled and setting one color, the others will move in the same direction

MODELS All

Access: RW

Name	Type
linkrgb	bool

image.processing.blacklevel.redgain

The gain red for black level

MODELS All

Access: RW

Name	Type
redgain	float
Constraints	
Minimum	0
Maximum	1
Step size	1
Precision	0.001

image.processing.blend.basicblend.bottom

Bottom blend edge.

MODELS

All

Access: RW

Name	Type
bottom	{ }

Name	Type
Start	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

Name	Type
base	int
hints	{ }

Name	Type
------	------

Name	Type
Width	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.processing.blend.basicblend.enable

Description not provided

MODELS

All

Access: RW

Name	Type
enable	bool

image.processing.blend.basicblend.left

Left blend edge.

MODELS

All

Access: RW

Name	Type
left	{ }

Name	Type
Start	{ }
Name	Type
base	int
hints	{ }
Name	Type
Width	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.processing.blend.basicblend.right

Right blend edge.

MODELS All

Access: RW

Name	Type
right	{ }
Name	Type
Start	{ }
Name	Type
base	int
hints	{ }
Name	Type
Width	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.processing.blend.basicblend.top

Top blend edge.

MODELS All

Access: RW

Name	Type
top	{ }
Name	Type
Start	{ }
Name	Type
base	int

	hints	{ }		
			Name	Type
Width	{ }			
			Name	Type
			base	int
			hints	{ }
			Name	Type

image.processing.blend.file.enable

Enable/Disable file blend

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.blend.file.maxselected

Max number of selected files

MODELS All

Access: R

Name	Type						
maxselected	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

image.processing.blend.file.selected

Currently selected files

MODELS All

Access: RW

Name	Type
selected	[string]

image.processing.blend.scurve

S-Curve exponent strength.

MODELS All

Access: RW

Name	Type
------	------

scurve	float
Constraints	
Minimum	1
Maximum	4
Step size	1
Precision	0.1

image.processing.transportdelay.actual

Actual transport delay.

MODELS

 All

Access: R

Name	Type
actual	{ }
Name Type	
base	int
hints	{ }
Name Type	

image.processing.transportdelay.desired

Desired transport delay.

MODELS

 All

Access: RW

Name	Type
desired	{ }
Name Type	
base	int
hints	{ }
Name Type	

image.processing.transportdelay.minimum

Minimum transport delay.

MODELS

 All

Access: R

Name	Type
minimum	{ }
Name Type	
base	int
hints	{ }
Name Type	

image.processing.warp.bow.bottomleftu

U vector for bottom left corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type
bottomleftu	{ }
Name Type angle { }	
Name Type base float hints { }	
Name Type	
length	{ }
Name Type base float hints { }	
Name Type	

image.processing.warp.bow.bottomleftv

V vector for bottom left corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type
bottomleftv	{ }
Name Type angle { }	
Name Type base float hints { }	
Name Type	
length	{ }
Name Type base float hints { }	
Name Type	

image.processing.warp.bow.bottomrightu

U vector for bottom right corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type												
bottomrightu	{ }												
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>angle</td><td>{ }</td></tr></table>	Name	Type	angle	{ }								
Name	Type												
angle	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }						
Name	Type												
base	float												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
length	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }						
Name	Type												
base	float												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												

image.processing.warp.bow.bottomrightv

V vector for bottom right corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type												
bottomrightv	{ }												
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>angle</td><td>{ }</td></tr></table>	Name	Type	angle	{ }								
Name	Type												
angle	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }						
Name	Type												
base	float												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
length	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }						
Name	Type												
base	float												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												

image.processing.warp.bow.enable

Enable/Disable bow warp

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.warp.bow.symmetric

Enable/Disable symmetric mode.

MODELS All

Access: RW

Name	Type
symmetric	bool

image.processing.warp.bow.topleftu

U vector for top left corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type																							
topleftu	{ }																							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>angle</td><td>{ }</td></tr></table>		Name	Type	angle	{ }																			
Name	Type																							
angle	{ }																							
<table><tr><td>length</td><td>{ }</td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>		length	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
length	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																
Name	Type																							
base	float																							
hints	{ }																							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																		
Name	Type																							
base	float																							
hints	{ }																							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																		
Name	Type																							
base	float																							
hints	{ }																							

image.processing.warp.bow.topleftv

V vector for top left corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type									
topleftv	{ }									
<table><tr><th>Name</th><th>Type</th></tr><tr><td>angle</td><td>{ }</td></tr></table>		Name	Type	angle	{ }					
Name	Type									
angle	{ }									
<table><tr><td>length</td><td>{ }</td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>		length	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
length	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		
Name	Type									
base	float									
hints	{ }									

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.processing.warp.bow.toprightu

U vector for top right corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type
toprightu	{ }

Name	Type
angle	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

length { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.processing.warp.bow.toprightv

V vector for top right corner. Positive angle is outwards.

MODELS All

Access: RW

Name	Type
toprightv	{ }

Name	Type
angle	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

length { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.processing.warp.enable

Enable/Disable all warp functions

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.warp.file.enable

Enable/Disable file warp

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.warp.file.selected

Currently selected file

MODELS All

Access: RW

Name	Type
selected	string

image.processing.warp.fourcorners.bottomleft

Bottom left coordinate in output resolution. Negative values allowed to go outside displayed area.

MODELS All

Access: RW

Name	Type
bottomleft	{ }

Name	Type
x	{ }
y	{ }

Name	Type
base	int
hints	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.processing.warp.fourcorners.bottomright

Bottom right coordinate in output resolution. Negative values allowed to go outside displayed area.

MODELS All

Access: RW

Name	Type																						
bottomright	{ }																						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>y</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>	Name	Type	x	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	y	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type																						
x	{ }																						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																
Name	Type																						
base	int																						
hints	{ }																						
y	{ }																						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																
Name	Type																						
base	int																						
hints	{ }																						

image.processing.warp.fourcorners.enable

Enable/Disable FourCorners warp

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.warp.fourcorners.screenheight

Deprecated: The height of the screen we are projecting on. Only used as in the ratio ScreenWidht/ScreenHeight, hence the unit is arbitrary.

MODELS All

Access: RW

Name	Type						
screenheight	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						

image.processing.warp.fourcorners.screenwidth

Deprecated: The width of the screen we are projecting on. Only used as in the ratio ScreenWidht/ScreenHeight, hence the unit is arbitrary.

MODELS All

Access: RW

Name	Type
screenwidth	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.processing.warp.fourcorners.topleft

Top left coordinate in output resolution. Negative values allowed to go outside displayed area.

MODELS All

Access: RW

Name	Type
topleft	{ }

Name	Type
x	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

y	{ }
---	-----

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.processing.warp.fourcorners.topright

Top right coordinate in output resolution. Negative values allowed to go outside displayed area.

MODELS All

Access: RW

Name	Type
topright	{ }

Name	Type
x	{ }

		Name	Type
		base	int
		hints	{ }
		Name	Type
y	{ }		
		Name	Type
		base	int
		hints	{ }
		Name	Type

image.processing.warp.position.enable

Enable/Disable Position warp

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.warp.position.horizontal

The horizontal displacement factor relative to output resolution. From -HRes/2 to + HRes/2

MODELS All

Access: RW

Name	Type
horizontal	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.processing.warp.position.vertical

The vertical displacement factor relative to output resolution. From -VRes/2 to +VRes/2

MODELS All

Access: RW

Name	Type
vertical	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.processing.warp.warpstatus

The warp status.

MODELS All

Access: R

Name	Type
warpstatus	enum
Values	
"Active"	
"Inactive"	
"Warning"	
"Error"	

image.processing.warp.warpstatusdescription

Warp status description

MODELS All

Access: R

Name	Type
warpstatusdescription	string

image.processing.warp.zoom.enable

Enable/Disable Zoom warp

MODELS All

Access: RW

Name	Type
enable	bool

image.processing.warp.zoom.factor

The zoom factor

MODELS All

Access: RW

Name	Type
factor	float
Constraints	
Minimum	0.5
Maximum	2
Step size	1
Precision	0.1

image.processing.warp.zoom.size

Resulting zoomed resolution

MODELS All

Access: R

Name	Type
size	{ }

Name	Type
x	{ }

Name	Type
base	int
hints	{ }

Name	Type
y	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.resolution.alpha.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type
size	{ }

Name	Type
pixels	{ }

Name	Type
base	int
hints	{ }

Name	Type
lines	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.resolution.available resolutions

List all resolutions that can currently be selected.

MODELS All

Access: R

Name	Type
available resolutions	[string]

image.resolution.beta.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type																															
size	{ }																															
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>pixels</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lines</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>	Name	Type	pixels	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lines</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	lines	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type																															
pixels	{ }																															
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lines</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	lines	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																						
Name	Type																															
base	int																															
hints	{ }																															
Name	Type																															
lines	{ }																															
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																									
Name	Type																															
base	int																															
hints	{ }																															
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																													
Name	Type																															

image.resolution.display.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type																														
size	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>pixels</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lines</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table></td></tr></table>	Name	Type	pixels	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lines</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	lines	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type																														
pixels	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>lines</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	lines	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type																														
base	int																														
hints	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																												
Name	Type																														
lines	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																				
Name	Type																														
base	int																														
hints	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																												
Name	Type																														

image.resolution.osd.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type
size	{ }

Name	Type
pixels	{ }

Name	Type
base	int
hints	{ }

Name	Type
lines	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.resolution.output.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type
size	{ }

Name	Type
pixels	{ }

Name	Type
base	int
hints	{ }

Name	Type
lines	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.resolution.processing.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type												
size	{ }												
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>pixels</td><td>{ }</td></tr></table>	Name	Type	pixels	{ }								
Name	Type												
pixels	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
lines	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												

image.resolution.resolution

The current resolution description.

MODELS All

Access: RW

Name	Type
resolution	string

image.resolution.size

The current resolution size (pixels x lines).

MODELS All

Access: R

Name	Type												
size	{ }												
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>pixels</td><td>{ }</td></tr></table>	Name	Type	pixels	{ }								
Name	Type												
pixels	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
lines	{ }												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type										
Name	Type												

image.saturation

Image color saturation. The value is normalized, 1 is default.

MODELS

 All

Access: RW

Name	Type
saturation	float
Constraints	
Minimum	0
Maximum	2
Step size	1
Precision	0.01

image.sharpness

Image sharpness. The value is normalized.

MODELS

 All

Access: RW

Name	Type
sharpness	int
Constraints	
Minimum	-2
Maximum	8
Step size	1
Precision	1

image.source.enable

Enable or disable the image preview

MODELS

 All

Access: R

Name	Type
enable	bool

image.source.l1displayport.capabilities

Capabilities

MODELS

 All

Access: R

Name	Type
capabilities	[enum]

Capabilities

MODELS

 All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.source.l1hdbaset1.layout

Source layout

MODELS

 All

Access: R

Name	Type
layout	{ }
NameType	
rows	{ }
NameType	
base	int
hints	{ }
NameType	
columns	{ }
NameType	
base	int
hints	{ }
NameType	
views	{ }
NameType	
base	int
hints	{ }
NameType	
planes	{ }
NameType	
base	int
hints	{ }
NameType	

image.source.l1hdbaset1.name

Name of the source

MODELS

 All

Access: R

Name	Type
name	string

image.source.l1hdbaset2.capabilities

Capabilities

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.source.l1hdbaset2.layout

Source layout

MODELS All

Access: R

Name	Type
layout	{ }
NameType	
rows	{ }
NameType	
base	int
hints	{ }
NameType	
columns	{ }
NameType	
base	int
hints	{ }
NameType	
views	{ }
NameType	
base	int
hints	{ }
NameType	
planes	{ }
NameType	
base	int
hints	{ }

Name	Type
------	------

image.source.l1hdbaset2.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.source.l1hdmi.capabilities

Capabilities

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.source.l1hdmi.layout

Source layout

MODELS All

Access: R

Name	Type
layout	{ }
NameType	
rows	{ }
NameType	
base	int
hints	{ }
NameType	
columns	{ }
NameType	
base	int
hints	{ }
NameType	
views	{ }
NameType	
base	int

hints { }	
Name Type	
planes { }	
Name Type	
base	int
hints { }	
Name Type	

image.source.l1hdmi.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.source.l1quadsdi.capabilities

Capabilities

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.source.l1quadsdi.layout

Source layout

MODELS All

Access: R

Name	Type
layout	{ }
Name Type	
rows	{ }
Name Type	
base	int
hints { }	
Name Type	
columns	{ }
Name Type	

		<table><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	base	int	hints	{ }		
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
views	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
planes	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							

image.source.l1quad.sdi.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
columns	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
views	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
planes	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

image.source.l1sdia.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.source.l1sdib.capabilities

Capabilities

MODELS All

Access: R

Name	Type			
capabilities	[enum]			
	<table><tr><th>Values</th></tr><tr><td>"HDCP"</td></tr><tr><td>"HDCP2"</td></tr></table>	Values	"HDCP"	"HDCP2"
Values				
"HDCP"				
"HDCP2"				

image.source.l1sdib.layout

Source layout

MODELS All

Access: R

Name	Type																																										
layout	{ }																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>rows</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>columns</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>	Name	Type	rows	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	columns	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	views	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	planes	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type																																										
rows	{ }																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																				
Name	Type																																										
base	int																																										
hints	{ }																																										
columns	{ }																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																				
Name	Type																																										
base	int																																										
hints	{ }																																										
views	{ }																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																				
Name	Type																																										
base	int																																										
hints	{ }																																										
planes	{ }																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																				
Name	Type																																										
base	int																																										
hints	{ }																																										

image.source.l1sdib.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.source.l1sdic.capabilities

Capabilities

MODELS All

Access: R

Name	Type		
capabilities	[enum]		
	<table><tr><th>Values</th></tr><tr><td>"HDCP"</td></tr></table>	Values	"HDCP"
Values			
"HDCP"			

"HDCP2"

image.source.l1sdc.layout

Source layout

MODELS All

Access: R

Name	Type																																																										
layout	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>rows</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>columns</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table></td></tr></table>	Name	Type	rows	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>columns</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>	Name	Type	columns	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	views	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>	Name	Type	planes	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type																																																										
rows	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																				
Name	Type																																																										
base	int																																																										
hints	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>columns</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>	Name	Type	columns	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	views	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>	Name	Type	planes	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type														
Name	Type																																																										
columns	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																				
Name	Type																																																										
base	int																																																										
hints	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>views</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	views	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>	Name	Type	planes	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																												
Name	Type																																																										
views	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																				
Name	Type																																																										
base	int																																																										
hints	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>planes</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr></table>	Name	Type	planes	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																														
Name	Type																																																										
planes	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																				
Name	Type																																																										
base	int																																																										
hints	{ }																																																										
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																								
Name	Type																																																										

image.source.l1sdc.name

Name of the source

MODELS All

Access: R

Name	Type
name	string

image.source.l1sdd.capabilities

Capabilities

MODELS All

Access: R

Name	Type
capabilities	[enum]
Values	
"HDCP"	
"HDCP2"	

image.source.l1sdid.layout

Source layout

MODELS

 All

Access: R

Name	Type						
layout	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>rows</td><td>{ }</td></tr></table>	Name	Type	rows	{ }		
Name	Type						
rows	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						
columns	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						
views	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						
planes	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

image.source.l1sdid.name

Name of the source

MODELS

 All

Access: R

Name	Type
------	------

name

string

image.source.preview.enable

Enable or disable the image preview

MODELS

All

Access: R

Name	Type
enable	bool

image.stereo.darktime

Darktime in us.

MODELS

All

Access: RW

Name	Type
darktime	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.stereo.glassync.delay

Sync delay in us.

MODELS

All

Access: RW

Name	Type
delay	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.stereo.glassync.delaymaximum

Maximum sync delay in us.

MODELS

All

Access: R

Name	Type
delaymaximum	{ }
Name	Type

base	int
hints	{ }

Name	Type
------	------

image.stereo.glassync.delayminimum

Minimum sync delay in us.

MODELS All

Access: R

Name	Type
delayminimum	{ }

base	int
hints	{ }

Name	Type
------	------

image.stereo.glassync.invert

Sync invert.

MODELS All

Access: RW

Name	Type
invert	bool

image.stereo.swapframepair

swap which stereo frames belong to each other

MODELS All

Access: RW

Name	Type
swapframepair	bool

image.testpattern.available

When false, the service will deny outside property change requests

MODELS All

Access: R

Name	Type
available	bool

image.testpattern.iterator.calibration.list

The list of testpatterns to iterate in the viewing order.

MODELS All

Access: R

Name	Type
list	[string]

image.testpattern.iterator.calibration.size

Return the size of the list of test patterns to iterate upon.

MODELS All

Access: R

Name	Type						
size	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.testpattern.iterator.iris.list

The list of testpatterns to iterate in the viewing order.

MODELS All

Access: R

Name	Type
list	[string]

image.testpattern.iterator.iris.size

Return the size of the list of test patterns to iterate upon.

MODELS All

Access: R

Name	Type						
size	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

image.testpattern.iterator.optics.list

The list of testpatterns to iterate in the viewing order.

MODELS All

Access: R

Name	Type
list	[string]

image.testpattern.iterator.optics.size

Return the size of the list of test patterns to iterate upon.

MODELS All

Access: R

Name	Type						
size	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

image.testpattern.iterator.shift.list

The list of testpatterns to iterate in the viewing order.

MODELS All

Access: R

Name	Type
list	[string]

image.testpattern.iterator.shift.size

Return the size of the list of test patterns to iterate upon.

MODELS All

Access: R

Name	Type						
size	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

image.testpattern.iterator.stereo.list

The list of testpatterns to iterate in the viewing order.

MODELS All

Access: R

Name	Type
list	[string]

image.testpattern.iterator.stereo.size

Return the size of the list of test patterns to iterate upon.

MODELS All

Access: R

Name	Type
size	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.testpattern.selected

The unique id of the selected pattern

MODELS All

Access: RW

Name	Type
selected	string

image.testpattern.show

Enable/disable the display of the selected pattern

MODELS All

Access: RW

Name	Type
show	bool

image.window.main.maximumsourceresolution

Maximum source resolution

MODELS All

Access: R

Name	Type
maximumsourceresolution	{ }

Name	Type
pixels	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

lines	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.window.main.position

Window position

MODELS All

Access: R

Name	Type
position	{ }
Name	Type
x	{ }
Name	Type
base	int
hints	{ }
Name	Type
y	{ }
Name	Type
base	int
hints	{ }
Name	Type

image.window.main.scalingmode

The scaling mode to apply to the source

MODELS All

Access: RW

Name	Type
scalingmode	enum
Values	
"Fill"	
"OneToOne"	
"FillScreen"	
"Stretch"	

image.window.main.size

Window size

MODELS All

Access: R

Name	Type
size	{ }

Name	Type
width	{ }

Name	Type
base	int
hints	{ }

Name	Type
height	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.window.main.source

The source that is dispayed in this window

MODELS All

Access: RW

Name	Type
source	string

image.window.main.sourceresolutionlimit

Is the source resolution limited.

MODELS All

Access: R

Name	Type
sourceresolutionlimit	bool

image.window.main.sourcevalidity

The reason(s) the source is not valid

MODELS All

Access: R

Name	Type
sourcevalidity	[enum]

Values
"ResolutionOutsideFlexRange"

iot.certificate.applicable

True if the certificate applies to this projector

MODELS

 All

Access: R

Name	Type
applicable	bool

iot.certificate.available

True if a certificate was found

MODELS

 All

Access: R

Name	Type
available	bool

iot.certificate.valid

Result of the validity check of the certificate

MODELS

 All

Access: R

Name	Type
valid	bool

iot.cloudservices.connectionstate

The current state of the connection.

MODELS

 All

Access: R

Name	Type
connectionstate	enum
Values	
"CONNECTED"	
"DISCONNECTED"	
"CONNECTING"	
"DISCONNECTING"	
"FAILED"	

iot.cloudservices.enable

True if IOT cloud connection is enabled, false otherwise

MODELS

 All

Access: RW

Name	Type
enable	bool

iot.proxy.address

This is the address (hostname/ip) of the proxy.

MODELS

All

Access: RW

Name	Type
address	string

iot.proxy.enabled

When enabled, iot connection will go over proxy specified in othe proxy parameters

MODELS

All

Access: RW

Name	Type
enabled	bool

iot.proxy.password

Password used to log in to proxy.

MODELS

All

Access: RW

Name	Type
password	string

iot.proxy.port

The port to use for the proxy conenction.

MODELS

All

Access: RW

Name	Type						
port	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

iot.proxy.username

Username used to log in to proxy.

MODELS

All

Access: RW

Name	Type
username	string

iot.requiretermagreement

Indiates whether or not the user is required to accept the term agreement.

MODELS All

Access: R

Name	Type
requiretermagreement	bool

iot.termagreement

Represents the users acknowledge of the IOT terms of agreement.

MODELS All

Access: RW

Name	Type
termagreement	bool

keydispatcher.repeatdelay

Initial key repeat delay in milliseconds

MODELS All

Access: R

Name	Type						
repeatdelay	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

keydispatcher.repeatinterval

Key repeat interval in milliseconds

MODELS All

Access: R

Name	Type						
repeatinterval	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
------	------

network.device.hdbaset.carrier

Whether the device has carrier or not

MODELS All

Access: R

Name	Type
carrier	bool

network.device.hdbaset.configuration

The configuration method of the device: auto or manual

MODELS All

Access: RW

Name	Type
configuration	enum
Values	
"AUTO"	
"MANUAL"	

network.device.hdbaset.devicetype

The general type of the network device

MODELS All

Access: R

Name	Type
devicetype	enum
Values	
"WIRED"	
"WIRELESS"	
"MOBILE"	

network.device.hdbaset.hwaddress

The active hardware (MAC) address

MODELS All

Access: R

Name	Type
hwaddress	string

network.device.hdbaset.ip4config

The current configuration for IP version 4

MODELS

 All

Access: R

Name	Type										
ip4config	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Mask</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Mask	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Mask	string										
Gateway	string										
NameServers	string										

network.device.hdbaset.ip4configmanual

Get/set the manual configuration for IP version 4

MODELS

 All

Access: RW

Name	Type										
ip4configmanual	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Mask</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Mask	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Mask	string										
Gateway	string										
NameServers	string										

network.device.hdbaset.ip6config

The current configuration for IP version 6

MODELS

 All

Access: R

Name	Type										
ip6config	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Prefix</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Prefix	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Prefix	string										
Gateway	string										
NameServers	string										

network.device.hdbaset.ip6configmanual

Get/set the manual configuration for IP version 4

MODELS

 All

Access: RW

Name	Type
ip6configmanual	{ }
Name	Type
Address	string
Prefix	string
Gateway	string
NameServers	string

network.device.hdbaset.speed

The speed of the device in Mbit/s

MODELS All

Access: R

Name	Type
speed	{ }
Name	Type
base	int
hints	{ }
Name	Type

network.device.hdbaset.state

The current state of the device

MODELS All

Access: R

Name	Type
state	enum
Values	
"CONNECTED"	
"DISCONNECTED"	
"CONNECTING"	
"NEEDAUTH"	
"IPCONFIG"	
"IPCHECK"	
"DISCONNECTING"	
"FAILED"	

network.device.hdbaset.stateinfo

Additional information about the device state. Can be empty

MODELS All

Access: R

Name	Type
stateinfo	string

network.device.lan.carrier

Whether the device has carrier or not

MODELS All

Access: R

Name	Type
carrier	bool

network.device.lan.configuration

The configuration method of the device: auto or manual

MODELS All

Access: RW

Name	Type
configuration	enum
	Values
	"AUTO"
	"MANUAL"

network.device.lan.devicetype

The general type of the network device

MODELS All

Access: R

Name	Type
devicetype	enum
	Values
	"WIRED"
	"WIRELESS"
	"MOBILE"

network.device.lan.hwaddress

The active hardware (MAC) address

MODELS All

Access: R

Name	Type
hwaddress	string

network.device.lan.ip4config

The current configuration for IP version 4

MODELS All

Access: R

Name	Type
ip4config	{ }
Name	Type
Address	string
Mask	string
Gateway	string
NameServers	string

network.device.lan.ip4configmanual

Get/set the manual configuration for IP version 4

MODELS All

Access: RW

Name	Type
ip4configmanual	{ }
Name	Type
Address	string
Mask	string
Gateway	string
NameServers	string

network.device.lan.ip6config

The current configuration for IP version 6

MODELS All

Access: R

Name	Type
ip6config	{ }
Name	Type
Address	string
Prefix	string
Gateway	string
NameServers	string

network.device.lan.ip6configmanual

Get/set the manual configuration for IP version 4

MODELS

 All

Access: RW

Name	Type										
ip6configmanual	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Prefix</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Prefix	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Prefix	string										
Gateway	string										
NameServers	string										

network.device.lan.speed

The speed of the device in Mbit/s

MODELS

 All

Access: R

Name	Type						
speed	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

network.device.lan.state

The current state of the device

MODELS

 All

Access: R

Name	Type									
state	enum									
	<table><tr><th>Values</th></tr><tr><td>"CONNECTED"</td></tr><tr><td>"DISCONNECTED"</td></tr><tr><td>"CONNECTING"</td></tr><tr><td>"NEEDAUTH"</td></tr><tr><td>"IPCONFIG"</td></tr><tr><td>"IPCHECK"</td></tr><tr><td>"DISCONNECTING"</td></tr><tr><td>"FAILED"</td></tr></table>	Values	"CONNECTED"	"DISCONNECTED"	"CONNECTING"	"NEEDAUTH"	"IPCONFIG"	"IPCHECK"	"DISCONNECTING"	"FAILED"
Values										
"CONNECTED"										
"DISCONNECTED"										
"CONNECTING"										
"NEEDAUTH"										
"IPCONFIG"										
"IPCHECK"										
"DISCONNECTING"										
"FAILED"										

network.device.lan.stateinfo

Additional information about the device state. Can be empty

MODELS

 All

Access: R

Name	Type
stateinfo	string

network.device.mobile.apn

the Access Point Name (APN) to connect to.

MODELS

 All

Access: RW

Name	Type
apn	string

network.device.mobile.configuration

The configuration method of the device: auto or manual

MODELS

 All

Access: RW

Name	Type
configuration	enum
Values	
"AUTO"	
"MANUAL"	

network.device.mobile.devicetype

The general type of the network device

MODELS

 All

Access: R

Name	Type
devicetype	enum
Values	
"WIRED"	
"WIRELESS"	
"MOBILE"	

network.device.mobile.hwaddress

The active hardware (MAC) address

MODELS

 All

Access: R

Name	Type
hwaddress	string

network.device.mobile.ip4config

The current configuration for IP version 4

MODELS All

Access: R

Name	Type										
ip4config	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Mask</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Mask	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Mask	string										
Gateway	string										
NameServers	string										

network.device.mobile.ip4configmanual

Get/set the manual configuration for IP version 4

MODELS All

Access: RW

Name	Type										
ip4configmanual	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Mask</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Mask	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Mask	string										
Gateway	string										
NameServers	string										

network.device.mobile.ip6config

The current configuration for IP version 6

MODELS All

Access: R

Name	Type										
ip6config	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Prefix</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Prefix	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Prefix	string										
Gateway	string										
NameServers	string										

network.device.mobile.ip6configmanual

Get/set the manual configuration for IP version 4

MODELS

 All

Access: RW

Name	Type
ip6configmanual	{ }

Name	Type
Address	string
Prefix	string
Gateway	string
NameServers	string

network.device.mobile.state

The current state of the device

MODELS

 All

Access: R

Name	Type
state	enum

Values
"CONNECTED"
"DISCONNECTED"
"CONNECTING"
"NEEDAUTH"
"IPCONFIG"
"IPCHECK"
"DISCONNECTING"
"FAILED"

network.device.mobile.stateinfo

Additional information about the device state. Can be empty

MODELS

 All

Access: R

Name	Type
stateinfo	string

network.device.wifi.accesspoints

A list of detected access points.

MODELS

 All

Access: R

Name	Type						
accesspoints	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>SSID</td><td>string</td></tr><tr><td>Strength</td><td>{ }</td></tr></table>	Name	Type	SSID	string	Strength	{ }
Name	Type						
SSID	string						
Strength	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>security</td><td>enum</td></tr></table>	Name	Type	security	enum		
Name	Type						
security	enum						
	<table><tr><th>Values</th></tr><tr><td>"NONE "</td></tr><tr><td>"WEP "</td></tr><tr><td>"WPA "</td></tr><tr><td>"WPA2 "</td></tr></table>	Values	"NONE "	"WEP "	"WPA "	"WPA2 "	
Values							
"NONE "							
"WEP "							
"WPA "							
"WPA2 "							
known	bool						

network.device.wifi.bitrate

The bit rate of the device in Kbit/s

MODELS All

Access: R

Name	Type						
bitrate	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

network.device.wifi.configuration

The configuration method of the device: auto or manual

MODELS All

Access: RW

Name	Type			
configuration	enum			
	<table><tr><th>Values</th></tr><tr><td>"AUTO"</td></tr><tr><td>"MANUAL"</td></tr></table>	Values	"AUTO"	"MANUAL"
Values				
"AUTO"				
"MANUAL"				

network.device.wifi.devicetype

The general type of the network device

MODELS All

Access: R

Name	Type
devicetype	enum
Values	
"WIRED"	
"WIRELESS"	
"MOBILE"	

network.device.wifi.enable

Enable or disable the wireless device.

MODELS All

Access: RW

Name	Type
enable	bool

network.device.wifi.hwaddress

The active hardware (MAC) address

MODELS All

Access: R

Name	Type
hwaddress	string

network.device.wifi.ip4config

The current configuration for IP version 4

MODELS All

Access: R

Name	Type										
ip4config	{ }										
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Mask</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>		Name	Type	Address	string	Mask	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Mask	string										
Gateway	string										
NameServers	string										

network.device.wifi.ip4configmanual

Get/set the manual configuration for IP version 4

MODELS

 All

Access: RW

Name	Type										
ip4configmanual	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Mask</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Mask	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Mask	string										
Gateway	string										
NameServers	string										

network.device.wifi.ip6config

The current configuration for IP version 6

MODELS

 All

Access: R

Name	Type										
ip6config	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Prefix</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Prefix	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Prefix	string										
Gateway	string										
NameServers	string										

network.device.wifi.ip6configmanual

Get/set the manual configuration for IP version 4

MODELS

 All

Access: RW

Name	Type										
ip6configmanual	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Address</td><td>string</td></tr><tr><td>Prefix</td><td>string</td></tr><tr><td>Gateway</td><td>string</td></tr><tr><td>NameServers</td><td>string</td></tr></table>	Name	Type	Address	string	Prefix	string	Gateway	string	NameServers	string
Name	Type										
Address	string										
Prefix	string										
Gateway	string										
NameServers	string										

network.device.wifi.maximumbitrate

The maximum bit rate of the device in Kbit/s

MODELS

 All

Access: R

Name	Type						
maximumbitrate	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

network.device.wifi.ssid

The SSID of the access point.

MODELS All

Access: R

Name	Type
ssid	string

network.device.wifi.state

The current state of the device

MODELS All

Access: R

Name	Type									
state	enum									
	<table><tr><th>Values</th></tr><tr><td>"CONNECTED"</td></tr><tr><td>"DISCONNECTED"</td></tr><tr><td>"CONNECTING"</td></tr><tr><td>"NEEDAUTH"</td></tr><tr><td>"IPCONFIG"</td></tr><tr><td>"IPCHECK"</td></tr><tr><td>"DISCONNECTING"</td></tr><tr><td>"FAILED"</td></tr></table>	Values	"CONNECTED"	"DISCONNECTED"	"CONNECTING"	"NEEDAUTH"	"IPCONFIG"	"IPCHECK"	"DISCONNECTING"	"FAILED"
Values										
"CONNECTED"										
"DISCONNECTED"										
"CONNECTING"										
"NEEDAUTH"										
"IPCONFIG"										
"IPCHECK"										
"DISCONNECTING"										
"FAILED"										

network.device.wifi.stateinfo

Additional information about the device state. Can be empty

MODELS All

Access: R

Name	Type
stateinfo	string

network.device.wifi.strength

The current signal strength of the access point.

MODELS

All

Access: R

Name	Type
strength	{ }
Name	Type
base	int
hints	{ }
Name	Type

network.domain

The domain name

MODELS

All

Access: RW

Name	Type
domain	string

network.hostname

The host name

MODELS

All

Access: RW

Name	Type
hostname	string

network.version

The Networking Service version

MODELS

All

Access: R

Name	Type
version	string

notification.count

The number of notifications received and dismissed

MODELS

All

Access: R

Name	Type
count	{ }
Name	Type

```
base    int
hints   { }
```

Name	Type
------	------

optics.autofocus.calibration.calibrateddark

Calibration has been performed with dark content.

MODELS All

Access: R

Name	Type
calibrateddark	bool

optics.autofocus.calibration.calibratedlight

Calibration has been performed with light content.

MODELS All

Access: R

Name	Type
calibratedlight	bool

optics.autofocus.enabled

Focus drift compensation is enabled.

MODELS All

Access: RW

Name	Type
enabled	bool

optics.autofocus.integrationinterval

The time the lens needs to cool down after warming up completely.

MODELS All

Access: RW

Name	Type						
integrationinterval	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

optics.autofocus.maxfocus

The maximum focus position of the lens.

MODELS

All

Access: RW

Name	Type						
maxfocus	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

optics.autofocus.maxzoom

The maximum zoom position of the lens.

MODELS

All

Access: RW

Name	Type						
maxzoom	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

optics.autofocus.minfocus

The minimum focus position of the lens.

MODELS

All

Access: RW

Name	Type						
minfocus	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

optics.autofocus.minzoom

The minimum zoom position of the lens.

MODELS

All

Access: RW

Name	Type		
minzoom	{ }		
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type		

```
base    float
hints   { }
```

Name	Type
------	------

optics.focus.calibrationstate

Current calibration state

MODELS All

Access: R

Name	Type
calibrationstate	enum
Values	
"Unknown"	
"Ok"	
"Busy"	
"Error"	
"NotImplemented"	

optics.focus.enabled

Enabled state

MODELS All

Access: RW

Name	Type
enabled	bool

optics.focus.limits.forward

Forward limit reached

MODELS All

Access: R

Name	Type
forward	bool

optics.focus.limits.reverse

Reverse limit reached

MODELS All

Access: R

Name	Type
reverse	bool

optics.focus.maxposition

Maximum available position

MODELS

 All

Access: R

Name	Type
maxposition	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.focus.minposition

Minimum available position

MODELS

 All

Access: R

Name	Type
minposition	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.focus.position

Current position

MODELS

 All

Access: R

Name	Type
position	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.focus.safetocalibrate

Safe to calibrate

MODELS

 All

Access: R

Name	Type
safetocalibrate	bool

optics.focus.safetooperate

Safe to operate state

MODELS All

Access: R

Name	Type
safetooperate	bool

optics.focus.state

Current state

MODELS All

Access: R

Name	Type
state	enum
Values	
"Stopped"	
"Running"	
"Calibrating"	
"Homing"	

optics.focus.target

Desired target

MODELS All

Access: RW

Name	Type
target	{ }
NameType	
base	int
hints	{ }
NameType	

optics.irisoperation

Property for selecting whether iris operation should be linked or individual

MODELS All

Access: RW

Name	Type
irisoperation	enum
Values	

"Individual"

"Linked"

optics.irisoperationavailable

Description not provided

MODELS

 All

Access: R

Name	Type
irisoperationavailable	bool

optics.lens.lens

Currently selected lens

MODELS

 All

Access: R

Name	Type
lens	{ }

Name	Type
Id	string
Description	string

optics.lens.metadata

Current lens metadata

MODELS

 All

Access: R

Name	Type
metadata	[{ }]

{ }

Name	Type
key	string
value	string

optics.lenspresent

Lens present

MODELS

 All

Access: R

Name	Type
lenspresent	bool

optics.lensshift.horizontal.calibrationstate

Current calibration state

MODELS

 All

Access: R

Name	Type
calibrationstate	enum
Values	
"Unknown"	
"Ok"	
"Busy"	
"Error"	
"NotImplemented"	

optics.lensshift.horizontal.enabled

Enabled state

MODELS

 All

Access: RW

Name	Type
enabled	bool

optics.lensshift.horizontal.limits.forward

Forward limit reached

MODELS

 All

Access: R

Name	Type
forward	bool

optics.lensshift.horizontal.limits.reverse

Reverse limit reached

MODELS

 All

Access: R

Name	Type
reverse	bool

optics.lensshift.horizontal.maxposition

Maximum available position

MODELS

 All

Access: R

Name	Type
maxposition	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.lensshift.horizontal.minposition

Minimum available position

MODELS All

Access: R

Name	Type
minposition	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.lensshift.horizontal.position

Current position

MODELS All

Access: R

Name	Type
position	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.lensshift.horizontal.safetocalibrate

Safe to calibrate

MODELS All

Access: R

Name	Type
safetocalibrate	bool

optics.lensshift.horizontal.safetooperate

Safe to operate state

MODELS All

Access: R

Name	Type
safetooperate	bool

optics.lensshift.horizontal.state

Current state

MODELS All

Access: R

Name	Type
state	enum
Values	
"Stopped"	
"Running"	
"Calibrating"	
"Homing"	

optics.lensshift.horizontal.target

Desired target

MODELS All

Access: RW

Name	Type
target	{ }
Name Type	
base	int
hints	{ }
Name Type	

optics.lensshift.vertical.calibrationstate

Current calibration state

MODELS All

Access: R

Name	Type
calibrationstate	enum
Values	
"Unknown"	
"Ok"	
"Busy"	
"Error"	
"NotImplemented"	

optics.lensshift.vertical.enabled

Enabled state

MODELS All

Access: RW

Name	Type
enabled	bool

optics.lensshift.vertical.limits.forward

Forward limit reached

MODELS All

Access: R

Name	Type
forward	bool

optics.lensshift.vertical.limits.reverse

Reverse limit reached

MODELS All

Access: R

Name	Type
reverse	bool

optics.lensshift.vertical.maxposition

Maximum available position

MODELS All

Access: R

Name	Type						
maxposition	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

optics.lensshift.vertical.minposition

Minimum available position

MODELS All

Access: R

Name	Type
minposition	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.lensshift.vertical.position

Current position

MODELS All

Access: R

Name	Type
position	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

optics.lensshift.vertical.safetocalibrate

Safe to calibrate

MODELS All

Access: R

Name	Type
safetocalibrate	bool

optics.lensshift.vertical.safetooperate

Safe to operate state

MODELS All

Access: R

Name	Type
safetooperate	bool

optics.lensshift.vertical.state

Current state

MODELS All

Access: R

Name	Type
state	enum

Values
"Stopped"
"Running"
"Calibrating"
"Homing"

optics.lensshift.vertical.target

Desired target

MODELS All

Access: RW

Name	Type
target	{ }
Name	Type
base	int
hints	{ }
Name	Type

optics.shutter.position

Position of shutter

MODELS All

Access: R

Name	Type
position	enum
Values	
"Open"	
"Closed"	

optics.shutter.target

Target of shutter

MODELS All

Access: RW

Name	Type
target	enum
Values	
"Open"	
"Closed"	

optics.zoom.calibrationstate

Current calibration state

MODELS

 All

Access: R

Name	Type
calibrationstate	enum
	Values
	"Unknown"
	"Ok"
	"Busy"
	"Error"
	"NotImplemented"

optics.zoom.enabled

Enabled state

MODELS

 All

Access: RW

Name	Type
enabled	bool

optics.zoom.limits.forward

Forward limit reached

MODELS

 All

Access: R

Name	Type
forward	bool

optics.zoom.limits.reverse

Reverse limit reached

MODELS

 All

Access: R

Name	Type
reverse	bool

optics.zoom.maxposition

Maximum available position

MODELS

 All

Access: R

Name	Type
------	------

maxposition	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

optics.zoom.minposition

Minimum available position

MODELS All

Access: R

minposition	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

optics.zoom.position

Current position

MODELS All

Access: R

position	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

optics.zoom.safetocalibrate

Safe to calibrate

MODELS All

Access: R

safetocalibrate	bool
-----------------	------

optics.zoom.safetooperate

Safe to operate state

MODELS All

Access: R

Name	Type
safetooperate	bool

optics.zoom.state

Current state

MODELS All

Access: R

Name	Type
state	enum
Values	
"Stopped"	
"Running"	
"Calibrating"	
"Homing"	

optics.zoom.target

Desired target

MODELS All

Access: RW

Name	Type
target	{ }
Name Type	
base	int
hints	{ }
Name Type	

peripheral.camera.enable

Enable/Disable camera

MODELS All

Access: R

Name	Type
enable	bool

peripheral.camera.exposuremode

Set the exposure mode of the camera if available

MODELS All

Access: RW

Name	Type
exposuremode	enum
Values	
"Manual"	
"Auto"	

peripheral.camera.exposuretime

Exposure time

MODELS All

Access: RW

Name	Type						
exposuretime	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

peripheral.camera.exposuretimemaximum

Maximum exposure time

MODELS All

Access: R

Name	Type						
exposuretimemaximum	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

peripheral.camera.exposuretimeminimum

Minimum exposure time

MODELS All

Access: R

Name	Type						
exposuretimeminimum	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

peripheral.camera.exposuretimestep

Inc/dec step exposure time

MODELS All

Access: R

Name	Type
exposuretimestep	{ }
Name	Type
base	float
hints	{ }
Name	Type

peripheral.camera.initialised

Indicates if the camera has been initialised and ready for use

MODELS All

Access: R

Name	Type
initialised	bool

peripheral.camera.resolution

The resolution that is currently used for taking pictures.

MODELS All

Access: RW

Name	Type
resolution	enum
Values	
"Low"	
"Medium"	
"High"	

peripheral.ranging.lastdistance

The last distance measured

MODELS All

Access: R

Name	Type
lastdistance	{ }
Name	Type
base	float

hints { }

Name Type

peripheral.ranging.sampletype

Set the current sample type and start measurement

MODELS All

Access: RW

Name	Type
sampletype	enum
Values	
"Single"	
"Continuous"	

peripheral.xlrfront.enable

Enable/Disable front XLR connector

MODELS All

Access: RW

Name	Type
enable	bool

peripheral.xlrfront.outputvoltage

Output Voltage

MODELS All

Access: RW

Name	Type
outputvoltage	enum
Values	
"V0"	
"V12"	
"V24"	

profile.domains

The list of available domains for profiles.

MODELS All

Access: R

Name	Type
domains	[string]

profile.presetassignments

All preset assignments.

MODELS

 All

Access: R

Name	Type
presetassignments	[{ }]

{ }

Name	Type
key	{ }

Name	Type
base	int
hints	{ }

Name	Type
value	string

profile.profiles

The list of created profiles.

MODELS

 All

Access: R

Name	Type
profiles	[string]

remotecontrol.address

The address of the remote control that the projector will respond to

MODELS

 All

Access: RW

Name	Type
address	int

Constraints

Minimum	1
Maximum	31
Step size	1
Precision	1

remotecontrol.broadcastaddress

The broadcast address

MODELS

 All

Access: RW

Name	Type
------	------

broadcastaddress	int
Constraints	
Minimum	0
Maximum	1
Step size	1
Precision	1

remotecontrol.sensors.front.enable

Enable or disable the IR sensor

MODELS All

Access: RW

Name	Type
enable	bool

remotecontrol.sensors.front.name

The display name of the IR sensor

MODELS All

Access: R

Name	Type
name	string

remotecontrol.sensors.rear.enable

Enable or disable the IR sensor

MODELS All

Access: RW

Name	Type
enable	bool

remotecontrol.sensors.rear.name

The display name of the IR sensor

MODELS All

Access: R

Name	Type
name	string

remotecontrol.sensors.side.enable

Enable or disable the IR sensor

MODELS

 All

Access: RW

Name	Type
enable	bool

remotecontrol.sensors.side.name

The display name of the IR sensor

MODELS

 All

Access: R

Name	Type
name	string

scheduler.actions

List of all scheduled actions' name.

MODELS

 All

Access: R

Name	Type
actions	[string]

scheduler.actiontypes

List of all available action types.

MODELS

 All

Access: R

Name	Type
actiontypes	[string]

scheduler.pause

Return True if action is paused, otherwise return False.

MODELS

 All

Access: RW

Name	Type
pause	bool

scheduler.scheduledactions

List of all scheduled actions data.

MODELS

 All

Access: RW

Name	Type
scheduledactions	string

scheduler.transitions

List of all available transitions for system action.

MODELS

 All

Access: R

Name	Type
transitions	[string]

screen.hdrboost

The HDR intensity

MODELS

 All

Access: RW

Name	Type
hdrboost	float
Constraints	
Minimum	0.5
Maximum	2
Step size	0.01
Precision	0.1

screen.luminance

The maximum luminance measured on the screen in cd/m2

MODELS

 All

Access: RW

Name	Type
luminance	float
Constraints	
Minimum	50
Maximum	10000
Step size	10
Precision	1

screen.screenaspect

The calculated aspect of the ScreenWidth/ScreenHeight

MODELS

 All

Access: R

Name	Type						
screenaspect	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

screen.screensize

The width and height of the screen we are projecting on.

MODELS

All

Access: RW

Name	Type																														
screensize	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>width</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>height</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>	Name	Type	width	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	height	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type																														
width	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																								
Name	Type																														
base	float																														
hints	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																												
Name	Type																														
height	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																								
Name	Type																														
base	float																														
hints	{ }																														
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																												
Name	Type																														

statistics.laser plate 1 bank 1 runtime.value

Counter value

MODELS

All

Access: R

Name	Type						
value	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

statistics.laser plate 1 bank 2 runtime.value

Counter value

MODELS

All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.laser plate 1 bank 3 runtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.laser plate 2 bank 1 runtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.laser plate 2 bank 2 runtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }

Name	Type
statistics.laser plate 2 bank 3 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }

Name	Type
statistics.laser plate 3 bank 1 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }

Name	Type
statistics.laser plate 3 bank 2 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }

Name	Type
statistics.laser plate 3 bank 3 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.laser plate 4 bank 1 runtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.laser plate 4 bank 2 runtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.laser plate 4 bank 3 runtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }

Name	Type
statistics.laser plate 5 bank 1 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type						
value	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
statistics.laser plate 5 bank 2 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type						
value	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
statistics.laser plate 5 bank 3 runtime.value	

Counter value

MODELS

All

Access: R

Name	Type						
value	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						

Name	Type
statistics.laserruntime.value	

Counter value

MODELS

All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.projectorruntime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.systemtime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }
Name	Type

statistics.uptime.value

Counter value

MODELS All

Access: R

Name	Type
value	{ }
Name	Type
base	int
hints	{ }

Name	Type
------	------

system.articlenumber

Article number.

MODELS All

Access: R

Name	Type
articlenumber	string

system.colorwheel

Article number of installed color wheel

MODELS All

Access: R

Name	Type
colorwheel	string

system.colorwheelname

Name of installed color wheel

MODELS All

Access: R

Name	Type
colorwheelname	string

system.date.availabletimezones

An array of all available time zones and their information.

MODELS All

Access: R

Name	Type
availabletimezones	[{ }]

{ }

Name	Type
name	string

system.date.ntp.connected

True if there is a connection to the NTP-server.

MODELS All

Access: R

Name	Type
connected	bool

system.date.ntp.enabled

True if NTP time synchronization should be used.

MODELS All

Access: R

Name	Type
enabled	bool

system.date.ntp.server

The NTP server hostname or address.

MODELS All

Access: R

Name	Type
server	string

system.date.timezone

The configured time zone of the system.

MODELS All

Access: R

Name	Type
timezone	string

system.eco.available

Returns true if state is available for this projector

MODELS All

Access: R

Name	Type
available	bool

system.eco.enable

Enable/disable the use of this state. Check if available first.

MODELS All

Access: RW

Name	Type
enable	bool

system.familyname

Family name.

MODELS All

Access: R

Name	Type
familyname	string

system.firmwareversion

Firmware version.

MODELS All

Access: R

Name	Type
firmwareversion	string

system.initialstate

State to transition to when the unit is started

MODELS All

Access: RW

Name	Type
initialstate	enum
Values	
"boot"	
"eco"	
"standby"	
"ready"	
"conditioning"	
"on"	
"service"	
"deconditioning"	
"error"	

system.license.applicable

Applicability of the license file.

MODELS All

Access: R

Name	Type
applicable	bool

system.license.available

Availability of a license file.

MODELS All

Access: R

Name	Type
available	bool

system.license.options

A dictionary of options and their values.

MODELS All

Access: R

Name	Type						
options	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>	Name	Type	key	string	value	string
Name	Type						
key	string						
value	string						

system.license.register.mandatory

Shows if registering the product is mandatory or optional

MODELS All

Access: R

Name	Type
mandatory	bool

system.license.register.remaining

Time left in minutes until registration must be performed.

MODELS All

Access: R

Name	Type						
remaining	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

system.license.register.total

Total grace period in minutes until registration must be performed.

MODELS

 All

Access: R

Name	Type
total	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

system.license.register.valid

True when the product has been successfully registered.

MODELS

 All

Access: R

Name	Type
valid	bool

system.license.valid

Validity of the license file.

MODELS

 All

Access: R

Name	Type
valid	bool

system.modelname

Model name.

MODELS

 All

Access: R

Name	Type
modelName	string

system.name

Custom name for this device.

MODELS

 All

Access: R

Name	Type
name	string

system.on.timeout.duration

Time (in seconds) to wait in this state before entering lower state.

MODELS All

Access: RW

Name	Type
duration	int
Constraints	
Minimum	60
Maximum	7200
Step size	30
Precision	1

system.on.timeout.enable

Enable/disable the timeout.

MODELS All

Access: RW

Name	Type
enable	bool

system.operationalmodes.availablemodes

List of available operational modes

MODELS All

Access: R

Name	Type
availablemodes	[enum]
Values	
"normal"	
"silent"	
"longlife"	
"highbrightness"	

system.operationalmodes.mode

The currently active operational mode

MODELS All

Access: RW

Name	Type
mode	enum

Values
"normal"
"silent"
"longlife"
"highbrightness"

system.ready.timeout.duration

Time (in seconds) to wait in this state before entering lower state.

MODELS All

Access: RW

Name	Type
duration	int
Constraints	
Minimum	60
Maximum	7200
Step size	30
Precision	1

system.ready.timeout.enable

Enable/disable the timeout.

MODELS All

Access: RW

Name	Type
enable	bool

system.resetprogress

Reset progress: [0..100]

MODELS All

Access: R

Name	Type						
resetprogress	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

system.resetresult

Description not provided

MODELS All

Access: R

Name	Type						
resetresult	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>time</td><td>string</td></tr><tr><td>result</td><td>enum</td></tr></table>	Name	Type	time	string	result	enum
Name	Type						
time	string						
result	enum						
	<table><tr><th>Values</th></tr><tr><td>"UNKNOWN"</td></tr><tr><td>"SUCCESS"</td></tr><tr><td>"FAILED"</td></tr></table>	Values	"UNKNOWN"	"SUCCESS"	"FAILED"		
Values							
"UNKNOWN"							
"SUCCESS"							
"FAILED"							

system.resetstatus

Reset status

MODELS All

Access: R

Name	Type
resetstatus	enum
	Values
	"Idle"
	"Resetting"
	"Failed"
	"Finished"

system.serialnumber

Serial number.

MODELS All

Access: R

Name	Type
serialnumber	string

system.standby.available

Returns true if state is available for this projector

MODELS All

Access: R

Name	Type
available	bool

system.standby.enable

Enable/disable the use of this state. Check if available first.

MODELS

 All

Access: RW

Name	Type
enable	bool

system.standby.timeout.duration

Time (in seconds) to wait in this state before entering lower state.

MODELS

 All

Access: RW

Name	Type
duration	int
Constraints	
Minimum	60
Maximum	7200
Step size	30
Precision	1

system.standby.timeout.enable

Enable/disable the timeout.

MODELS

 All

Access: RW

Name	Type
enable	bool

system.state

The current state of the unit

MODELS

 All

Access: R

Name	Type
state	enum
Values	
"boot"	
"eco"	
"standby"	
"ready"	
"conditioning"	
"on"	
"service"	

"deconditioning"
"error"

system.targetstate

The state the unit is transitioning into

MODELS All

Access: R

Name	Type
targetstate	enum
Values	
"boot"	
"eco"	
"standby"	
"ready"	
"conditioning"	
"on"	
"service"	
"deconditioning"	
"error"	

ui.access.enduser

True and available when the user has end user access privileges.

MODELS All

Access: R

Name	Type
enduser	bool

ui.backlight.state

Description not provided

MODELS All

Access: RW

Name	Type
state	enum
Values	
"off"	
"on"	
"Auto"	

ui.backlight.timeout

The amount of seconds after which the lcd backlight will be switched off when the menu and stealth mode are not active and there is no activity.

MODELS

All

Access: RW

Name	Type
timeout	int
Constraints	
Minimum	5
Maximum	3600
Step size	1
Precision	1

ui.hasstealthmode

Description not provided

MODELS

All

Access: R

Name	Type
hasstealthmode	bool

ui.language

The user interface language

MODELS

All

Access: RW

Name	Type
language	string

ui.layer.advancedblend.drawing

Drawing commands in the form of a JSON object

MODELS

All

Access: RW

Name	Type
drawing	string

ui.layer.advancedblend.enable

Enable or disable the layer

MODELS

All

Access: RW

Name	Type
enable	bool

ui.layer.advancedblend.palette

Color palette that can be used when drawing the blend layer

MODELS All

Access: RW

Name	Type
palette	[string]

ui.layer.basicblacklevel.color

The edge color, e.g '#ffff00' or 'rgba(255,255,0,0.5)

MODELS All

Access: RW

Name	Type
color	string

ui.layer.basicblacklevel.enable

Enable or disable the layer

MODELS All

Access: RW

Name	Type
enable	bool

ui.layer.basicblacklevel.selection

Toggle edge selection

MODELS All

Access: RW

Name	Type										
selection	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Bottom</td><td>bool</td></tr><tr><td>Left</td><td>bool</td></tr><tr><td>Right</td><td>bool</td></tr><tr><td>Top</td><td>bool</td></tr></table>	Name	Type	Bottom	bool	Left	bool	Right	bool	Top	bool
Name	Type										
Bottom	bool										
Left	bool										
Right	bool										
Top	bool										

ui.layer.basicblacklevel.selectioncolor

The color to use for selected edges, e.g '#ff0000' or 'rgba(255,0,0,0.5)'

MODELS

 All

Access: RW

Name	Type
selectioncolor	string

ui.layer.basicblend.color

The edge color, e.g '#ffff00' or 'rgba(255,255,0,0.5)

MODELS

 All

Access: RW

Name	Type
color	string

ui.layer.basicblend.enable

Enable or disable the layer

MODELS

 All

Access: RW

Name	Type
enable	bool

ui.layer.basicblend.selection

Toggle edge selection

MODELS

 All

Access: RW

Name	Type
selection	{ }
Name	Type
Bottom	bool
Left	bool
Right	bool
Top	bool

ui.layer.basicblend.selectioncolor

The color to use for selected edges, e.g '#ff0000' or 'rgba(255,0,0,0.5)'

MODELS

 All

Access: RW

Name	Type
selectioncolor	string

ui.layer.blank.border

(Optional) Show/hide a border/outline of the screen

MODELS All

Access: RW

Name	Type
border	bool

ui.layer.blank.bordercolor

(Optional) Specify the border color, e.g '#ff0000' or 'rgba(255,0,0,0.75)'

MODELS All

Access: RW

Name	Type
bordercolor	string

ui.layer.blank.enable

Enable or disable the layer. When enabled, the screen will be covered in black.

MODELS All

Access: RW

Name	Type
enable	bool

ui.layer.blank.icon

A single UTF character from the Barco-Icons font set.

MODELS All

Access: RW

Name	Type
icon	string

ui.layer.blank.show

Used only for persisting whether or not to show the layer

MODELS All

Access: RW

Name	Type
show	bool

ui.layer.blank.showtext

Show or hide the text

MODELS All

Access: RW

Name	Type
showtext	bool

ui.layer.blank.text

(Optional) Specify a text to show at the center of the screen

MODELS All

Access: RW

Name	Type
text	string

ui.layer.fourcorner.cornercolor

The color to use for unselected corners, e.g '#ff0000' or 'rgba(255,0,0,0.75)'

MODELS All

Access: RW

Name	Type
cornercolor	string

ui.layer.fourcorner.enable

Enable or disable the layer

MODELS All

Access: RW

Name	Type
enable	bool

ui.layer.fourcorner.linecolor

The line color, e.g '#ffff00' or 'rgba(255,255,0,0.5)'

MODELS All

Access: RW

Name	Type
linecolor	string

ui.layer.fourcorner.lines

Show or hide lines between the corners

MODELS All

Access: RW

Name	Type
lines	bool

ui.layer.fourcorner.selection

Toggle corner selection

MODELS All

Access: RW

Name	Type
selection	{ }

Name	Type
TopLeft	bool
TopRight	bool
BottomRight	bool
BottomLeft	bool

ui.layer.fourcorner.selectioncolor

The color to use for selected corners, e.g '#ff0000' or 'rgba(255,0,0,0.75)'

MODELS All

Access: RW

Name	Type
selectioncolor	string

ui.layer.grid.color

Default color for grid points, e.g '#ff0000' or 'rgba(0,0,255,0.5)'

MODELS All

Access: RW

Name	Type
color	string

ui.layer.grid.enable

Enable or disable the layer

MODELS All

Access: RW

Name	Type
enable	bool

ui.layer.grid.lines

DEPRECATED. Use ShowLines instead. This is for backwards compability

MODELS

 All

Access: RW

Name	Type
lines	bool

ui.layer.grid.mark

List of row,column and color triplets for marking points in the grid. The color is specified as '#ff00ff' or 'rgba(0,255,255,0.75)'

MODELS

 All

Access: RW

Name	Type																																
mark	[{ }]																																
	{ }																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>Color</td><td>string</td></tr></table></td></tr></table></td></tr></table>	Name	Type	X	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>Color</td><td>string</td></tr></table></td></tr></table>	Name	Type	Y	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>Color</td><td>string</td></tr></table>	Name	Type	Color	string
Name	Type																																
X	{ }																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																										
Name	Type																																
base	int																																
hints	{ }																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>Color</td><td>string</td></tr></table></td></tr></table>	Name	Type	Y	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>Color</td><td>string</td></tr></table>	Name	Type	Color	string														
Name	Type																																
Y	{ }																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																										
Name	Type																																
base	int																																
hints	{ }																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Color</td><td>string</td></tr></table>	Name	Type	Color	string																												
Name	Type																																
Color	string																																

ui.layer.grid.points

Number of grid points

MODELS

 All

Access: RW

Name	Type						
points	enum						
	<table><tr><th>Values</th></tr><tr><td>" 2x2 "</td></tr><tr><td>" 3x3 "</td></tr><tr><td>" 5x5 "</td></tr><tr><td>" 9x9 "</td></tr><tr><td>" 17x17 "</td></tr></table>	Values	" 2x2 "	" 3x3 "	" 5x5 "	" 9x9 "	" 17x17 "
Values							
" 2x2 "							
" 3x3 "							
" 5x5 "							
" 9x9 "							
" 17x17 "							

"33x33 "

ui.layer.grid.showlines

Toggle drawing lines between grid points

MODELS All

Access: RW

Name	Type
showlines	bool

ui.layer.grid.showpoints

Toggle drawing grid points

MODELS All

Access: RW

Name	Type
showpoints	bool

ui.length

Preferred unit for display of lengths and distances

MODELS All

Access: RW

Name	Type
length	enum
Values	
"M"	
"CM"	
"FT"	
"IN"	

ui.lensmenu

Show or hide the lens menu

MODELS All

Access: RW

Name	Type
lensmenu	bool

ui.luminance

Preferred unit for luminance

MODELS All

Access: RW

Name	Type
luminance	enum
Values	
"NITS"	
"FL "	

ui.menu

Show or hide the menu

MODELS

 All

Access: RW

Name	Type
menu	bool

ui.menuposition

Placement of menu related to full screen.

MODELS

 All

Access: RW

Name	Type
menuposition	enum
Values	
"TopLeft"	
"Top"	
"TopRight"	
"Right"	
"BottomRight"	
"Bottom"	
"BottomLeft"	
"Left"	
"Center"	

ui.message

The presistent message to show on the screen. Rich text allowed

MODELS

 All

Access: R

Name	Type
message	{ }
Name	Type
title	string

iconstring

messagestring

ui.messageposition

Placement of the message relative to full screen

MODELS

 All

Access: RW

Name	Type
messageposition	enum
Values	
"TopLeft"	
"Top"	
"TopRight"	
"Right"	
"BottomRight"	
"Bottom"	
"BottomLeft"	
"Left"	
"Center"	

ui.minimize

Minimize the menu when it is enabled

MODELS

 All

Access: RW

Name	Type
minimize	bool

ui.osd

Enable or disable on screen display

MODELS

 All

Access: RW

Name	Type
osd	bool

ui.patternmenu

Show or hide the pattern menu shortcut

MODELS

 All

Access: RW

Name	Type
patternmenu	bool

ui.poweroffhint

When true, a dialog shows info about powering down

MODELS All

Access: RW

Name	Type
poweroffhint	bool

ui.showmessage

Show a persistent message on the screen

MODELS All

Access: R

Name	Type
showmessage	bool

ui.sourcemenu

Show or hide the input source shortcut menu

MODELS All

Access: RW

Name	Type
sourcemenu	bool

ui.sourcesignal

Show/hide the source signal information popup

MODELS All

Access: RW

Name	Type
sourcesignal	bool

ui.sourcesignalposition

Placement of the source signal information

MODELS All

Access: RW

Name	Type
sourcesignalposition	enum

Values
"TopLeft"
"Top"
"TopRight"
"Right"
"BottomRight"
"Bottom"
"BottomLeft"
"Left"
"Center"

ui.splashscreen.show

Value to indicate if the splash screen should be shown

MODELS All

Access: R

Name	Type
show	bool

ui.splashscreen.timeoutseconds

Time (in seconds) the splash screen is shown on startup

MODELS All

Access: R

Name	Type
timeoutseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

ui.stealthmode

When the projector is in stealth mode, all controllable LEDs are switched off.

MODELS All

Access: RW

Name	Type
stealthmode	enum
Values	
"Off"	
"On"	
"OnUntilReboot"	

ui.temperature

Preferred unit for display of temperature values

MODELS All

Access: RW

Name	Type
temperature	enum
Values	
"C"	
"F"	

ui.theme

The theme setting of the user interface.

MODELS All

Access: RW

Name	Type
theme	enum
Values	
"Light"	
"Dark"	

ui.touchscreen.mode

Enables or disables the touchscreen test.

MODELS All

Access: R

Name	Type
mode	enum
Values	
"Normal"	
"Test"	

ui.touchscreen.status

Current status of the touchscreen text

MODELS All

Access: RW

Name	Type
status	enum
Values	

"Inactive"

"Ongoing"

"Complete"

user.admin.activesessioncount

Description not provided

MODELS

All

Access: R

Name	Type
activesessioncount	{ }
Name	Type
base	int
hints	{ }
Name	Type

user.admin.enabled

Description not provided

MODELS

All

Access: R

Name	Type
enabled	bool

user.admin.group

Description not provided

MODELS

All

Access: R

Name	Type
group	string

user.admin.publickey

Description not provided

MODELS

All

Access: R

Name	Type
publickey	string

user.admin.username

Description not provided

MODELS

All

Access: R

Name	Type
username	string

user.admin.userslug

Description not provided

MODELS

All

Access: R

Name	Type
userslug	string

user.authenticationrequired

Require authentication of all users

MODELS

All

Access: R

Name	Type
authenticationrequired	bool

user.availablegroups

The available user groups

MODELS

All

Access: R

Name	Type
availablegroups	[string]

user.currentuser.activesessioncount

Description not provided

MODELS

All

Access: R

Name	Type
activesessioncount	{ }
Name	Type
base	int
hints	{ }
Name	Type

user.currentuser.enabled

Description not provided

MODELS

 All

Access: R

Name	Type
enabled	bool

user.currentuser.group

Description not provided

MODELS

 All

Access: R

Name	Type
group	string

user.currentuser.publickey

Description not provided

MODELS

 All

Access: R

Name	Type
publickey	string

user.currentuser.username

Description not provided

MODELS

 All

Access: R

Name	Type
username	string

user.currentuser.userslug

Description not provided

MODELS

 All

Access: R

Name	Type
userslug	string

user.list

The list of all users

MODELS

 All

Access: R

Name	Type										
list	[{ }]										
	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Username</td><td>string</td></tr><tr><td>UserSlug</td><td>string</td></tr><tr><td>Group</td><td>string</td></tr><tr><td>Sessions</td><td>{ }</td></tr></table>	Name	Type	Username	string	UserSlug	string	Group	string	Sessions	{ }
Name	Type										
Username	string										
UserSlug	string										
Group	string										
Sessions	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }				
Name	Type										
base	int										
hints	{ }										
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type								
Name	Type										

user.pincodeauthenticationenabled

Allow authenticating users with pin code

MODELS All

Access: R

Name	Type
pincodeauthenticationenabled	bool

user.poweruser.activesessioncount

Description not provided

MODELS All

Access: R

Name	Type						
activesessioncount	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

user.poweruser.enabled

Description not provided

MODELS All

Access: R

Name	Type
enabled	bool

user.poweruser.group

Description not provided

MODELS All

Access: R

Name	Type
group	string

user.poweruser.publickey

Description not provided

MODELS All

Access: R

Name	Type
publickey	string

user.poweruser.username

Description not provided

MODELS All

Access: R

Name	Type
username	string

user.poweruser.userslug

Description not provided

MODELS All

Access: R

Name	Type
userslug	string

user.resetadministratorchallenge

Challenge for resetting the administrator user

MODELS All

Access: R

Name	Type
resetadministratorchallenge	{ }
Name	Type
base	int

hints { }

Name Type

user.user.activesessioncount

Description not provided

MODELS All

Access: R

Name	Type
activesessioncount	{ }

Name	Type
base	int
hints	{ }

Name Type

user.user.enabled

Description not provided

MODELS All

Access: R

Name	Type
enabled	bool

user.user.group

Description not provided

MODELS All

Access: R

Name	Type
group	string

user.user.publickey

Description not provided

MODELS All

Access: R

Name	Type
publickey	string

user.user.username

Description not provided

MODELS All

Access: R

Name	Type
username	string

user.user.userslug

Description not provided

MODELS

All

Access: R

Name	Type
userslug	string

Methods

Alphabetical list of all methods

.authenticate

[DEPRECATED] Authenticate using pin code

MODELS

 All

Parameters

Name	Type
code	int

Return value

Name	Type
userslug	string

.generatenewtoken

Request a new token for the currently authenticated user

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
token	string

.getchallenge

Get the challenge for key-based authentication

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
challenge	string

.introspect

Introspect the API for a specified object path

MODELS

 All

Parameters

Name	Type
object	string
recursive	bool
descriptions	bool

Return value

Name	Type
------	------

.login

Login using username and signed challenge (set token true to receive a token)

MODELS All

Parameters

Name	Type
username	string
signature	string
token	bool

Return value

Name	Type
token	string

.logout

De-authenticate

MODELS All

This method does not require any parameters.

This method has no return value.

dmx.listchannels

Return a list of available channel names

MODELS All

This method does not require any parameters.

Return value

Name	Type
modes	[string]

dmx.listmodes

Return a list of all modes

MODELS All

This method does not require any parameters.

Return value

Name	Type
modes	[string]

environment.getalarminfo

Description not provided

MODELS All

This method does not require any parameters.

Return value

Name	Type										
alarminfo	[{ }]										
	{ }										
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>severity</td><td>string</td></tr><tr><td>timestamp</td><td>string</td></tr><tr><td>source</td><td>string</td></tr><tr><td>description</td><td>string</td></tr></table>	Name	Type	severity	string	timestamp	string	source	string	description	string
Name	Type										
severity	string										
timestamp	string										
source	string										
description	string										

environment.getcontrolblocks

Description not provided

MODELS All

Parameters

Name	Type							
type	enum							
	<table><tr><th>Values</th></tr><tr><td>"Actuator"</td></tr><tr><td>"Alarm"</td></tr><tr><td>"Controller"</td></tr><tr><td>"Filter"</td></tr><tr><td>"GenericBlock"</td></tr><tr><td>"Sensor"</td></tr></table>	Values	"Actuator"	"Alarm"	"Controller"	"Filter"	"GenericBlock"	"Sensor"
Values								
"Actuator"								
"Alarm"								
"Controller"								
"Filter"								
"GenericBlock"								
"Sensor"								
valuetype	enum							
	<table><tr><th>Values</th></tr><tr><td>"ADC"</td></tr></table>	Values	"ADC"					
Values								
"ADC"								

"Altitude"
"Angle"
"Any"
"Average"
"Comparison"
"Constant"
"Contamination"
"Coordinate"
"Current"
"Delay"
"Difference"
"Disabled"
"Driver"
"Formula"
"Humidity"
"Interpolation"
"Light"
"Limit"
"LiquidPressure"
"Manual"
"Median"
"Mode"
"Noise"
"PID"
"PWM"
"Peltier"
"Power"
"Pressure"
"Pump"
"Range"
"Ratio"
"Resistance"
"Simulation"
"Speed"
"State"
"Temperature"
"Threshold"
"Upgrader"
"Voltage"
"Volume"
"Waveform"
"Weighting"

Return value

Name	Type						
blocks	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>{ }</td></tr></table>	Name	Type	key	string	value	{ }
Name	Type						
key	string						
value	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

illumination.clo.engage

Engage CLO at the current light level

MODELS All

This method does not require any parameters.

This method has no return value.

image.color.p7.custom.copypresettocustom

Description not provided

MODELS All

Parameters

Name	Type
presetname	string

This method has no return value.

image.color.p7.custom.resetpreset

Reset preset back to its default values

MODELS All

Parameters

Name	Type
presetname	string

This method has no return value.

image.color.p7.custom.resettonative

Description not provided

MODELS All

This method does not require any parameters.

This method has no return value.

image.color.rgbmode.nextrgbmode

Change to the next RGB mode. Lets you cycle through all possible modes.

MODELS All

This method does not require any parameters.

This method has no return value.

image.connector.l1displayport.edid.list

List system EDIDs available for this connector

MODELS All

This method does not require any parameters.

Return value

Name	Type						
selections	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>group</td><td>string</td></tr><tr><td>edids</td><td>[string]</td></tr></table>	Name	Type	group	string	edids	[string]
Name	Type						
group	string						
edids	[string]						

image.connector.l1hdbaset1.edid.list

List system EDIDs available for this connector

MODELS All

This method does not require any parameters.

Return value

Name	Type						
selections	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>group</td><td>string</td></tr><tr><td>edids</td><td>[string]</td></tr></table>	Name	Type	group	string	edids	[string]
Name	Type						
group	string						
edids	[string]						

image.connector.l1hdbaset2.edid.list

List system EDIDs available for this connector

MODELS

 All

This method does not require any parameters.

Return value

Name	Type						
selections	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>group</td><td>string</td></tr><tr><td>edids</td><td>[string]</td></tr></table>	Name	Type	group	string	edids	[string]
Name	Type						
group	string						
edids	[string]						

image.connector.l1hdmi.edid.list

List system EDIDs available for this connector

MODELS

 All

This method does not require any parameters.

Return value

Name	Type						
selections	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>group</td><td>string</td></tr><tr><td>edids</td><td>[string]</td></tr></table>	Name	Type	group	string	edids	[string]
Name	Type						
group	string						
edids	[string]						

image.connector.list

Description not provided

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
connectors	[string]

image.display.listdisplaymodes

List possible display modes.

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
displaymodes	[enum]
	Values
	"Mono"
	"AutoStereo"
	"ActiveStereo"
	"NightVision"
	"IGPixelShift"
	"IGPixelShiftNV"
	"IGPixelShiftFullNV"

image.list

List all available orientation modes

MODELS All

This method does not require any parameters.

Return value

Name	Type
orientationmode	[string]

image.listgammatypes

List all available gamma types.

MODELS All

This method does not require any parameters.

Return value

Name	Type
gammatypes	[string]

image.processing.blacklevel.basicblacklevel.getblacklevelarea

Returns the four boxes describing the black level edges.

MODELS All

Parameters

Name	Type
resolution_width	{ }
	Name Type
	base float

hints { }

Name	Type
------	------

resolution_height { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Return value

Name	Type
output	{ }

Name	Type
Top	{ }

Name	Type
Start1	{ }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Start2 { }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Bottom { }

Name	Type
Start1	{ }

Name	Type
------	------

X	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Y	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Start2 { }									
X	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td></td><td></td></tr></table>		Name	Type				
		Name	Type						
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Y	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Left { }									
Start1	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td></td><td></td></tr></table>		Name	Type				
		Name	Type						
X	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Y	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Start2 { }									
X	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td></td><td></td></tr></table>		Name	Type				
		Name	Type						
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						

[illegible]

image.processing.blacklevel.basicblacklevel.getwarpedblacklevelarea

Returns the four boxes describing the black level edges, after warp.

MODELS

All

Parameters

Name	Type
resolution_width	{ }

Name	Type
base	float

hints { }

Name	Type
------	------

resolution_height { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Return value

Name	Type
output	{ }

Name	Type
Top	{ }

Name	Type
Start1	{ }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Start2 { }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Bottom { }

Name	Type
Start1	{ }

Name	Type
------	------

X	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Y	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Start2 { }									
X	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Y	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Left { }									
Start1	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
X	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Y	{ }								
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						
Start2 { }									
X	{ }	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
		Name	Type						
		base	float						
		hints	{ }						

[illegible]

image.processing.blacklevel.file.delete

Deletes a file with the given name.

MODELS All

Parameters

Name	Type
filename	string

This method has no return value.

image.processing.blacklevel.file.list

Returns a list of available black level correction files

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
filenames	[string]

image.processing.blend.basicblend.getblendarea

Returns the four boxes describing the blend edges.

MODELS

 All

Parameters

Name	Type						
resolution_width	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
resolution_height	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

Return value

Name	Type						
output	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Top</td><td>{ }</td></tr></table>	Name	Type	Top	{ }		
Name	Type						
Top	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Start1</td><td>{ }</td></tr></table>	Name	Type	Start1	{ }		
Name	Type						
Start1	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>	Name	Type	X	{ }		
Name	Type						
X	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr></table>	Name	Type	Y	{ }		
Name	Type						
Y	{ }						

		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
Start2	{ }							
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>	Name	Type	X	{ }			
Name	Type							
X	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
	Y	{ }						
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
Width1	{ }							
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>	Name	Type	X	{ }			
Name	Type							
X	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
	Y	{ }						
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
Width2	{ }							
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>	Name	Type	X	{ }			
Name	Type							
X	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
	Y	{ }						
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type							
base	float							
hints	{ }							
		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type							
Bottom	{ }							
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type					
Name	Type							

Start1 { }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Start2 { }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Width1 { }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Y { }

Name	Type
base	float
hints	{ }

Name	Type
------	------

Width2 { }

Name	Type
X	{ }

Name	Type
base	float
hints	{ }

																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					</
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

Width2 { }	hints { }	
	NameType	
	NameType	
	X { }	
	NameType	
	base float	
	hints { }	
	NameType	
	Y { }	
	NameType	
Right { }	base float	
	hints { }	
	NameType	
	Start1 { }	
	NameType	
	X { }	
	NameType	
	base float	
	hints { }	
	NameType	
Start2 { }	Y { }	
	NameType	
	base float	
	hints { }	
	NameType	
	Start2 { }	
	NameType	
	X { }	
	NameType	
	base float	
Width1 { }	hints { }	
	NameType	
	base float	
	hints { }	
	NameType	
	Y { }	
	NameType	
	base float	
	hints { }	
	NameType	

X	{ }	NameType	
		base	float
		hints	{ }
		NameType	
Y	{ }	NameType	
		base	float
		hints	{ }
		NameType	
Width2	{ }	NameType	
		X	{ }
		NameType	
		base	float
		hints	{ }
		NameType	
		Y	{ }
		NameType	
		base	float
		hints	{ }
		NameType	

image.processing.blend.basicblend.getwarpedblendarea

Returns the four boxes describing the blend edges, after warp.

MODELS All

Parameters

resolution_width	{ }	NameType	
		base	float
		hints	{ }
		NameType	
resolution_height	{ }	NameType	
		base	float
		hints	{ }
		NameType	

Return value

Name	Type
------	------

output { }							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Top</td><td>{ }</td></tr></table>		Name	Type	Top	{ }		
Name	Type						
Top	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Start1</td><td>{ }</td></tr></table>		Name	Type	Start1	{ }		
Name	Type						
Start1	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>		Name	Type	X	{ }		
Name	Type						
X	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr></table>		Name	Type	Y	{ }		
Name	Type						
Y	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Start2</td><td>{ }</td></tr></table>		Name	Type	Start2	{ }		
Name	Type						
Start2	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>		Name	Type	X	{ }		
Name	Type						
X	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr></table>		Name	Type	Y	{ }		
Name	Type						
Y	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Width1</td><td>{ }</td></tr></table>		Name	Type	Width1	{ }		
Name	Type						
Width1	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>		Name	Type	X	{ }		
Name	Type						
X	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr></table>		Name	Type	Y	{ }		
Name	Type						
Y	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>Width2</td><td>{ }</td></tr></table>		Name	Type	Width2	{ }		
Name	Type						
Width2	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

Bottom { }	X { }		
		Name	Type
		base	float
		hints	{ }
	Y { }		
		Name	Type
		base	float
		hints	{ }
	Start1 { }		
		Name	Type
	X { }		
		Name	Type
		base	float
		hints	{ }
	Y { }		
		Name	Type
		base	float
		hints	{ }
	Start2 { }		
		Name	Type
	X { }		
		Name	Type
		base	float
		hints	{ }
	Y { }		
		Name	Type
		base	float
		hints	{ }
	Width1 { }		
		Name	Type
	X { }		
		Name	Type
		base	float
		hints	{ }

```

graph TD
    Program["Program { }"]
    Left["Left { }"]
    Width2["Width2 { }"]
    Y1["Y { }"]
    Start1["Start1 { }"]
    X["X { }"]
    Y2["Y { }"]
    base1["base float"]
    base2["base float"]
    base3["base float"]
    hints1["hints { }"]
    hints2["hints { }"]
    hints3["hints { }"]

    Program --- Left
    Program --- Width2
    Program --- Y1
    Left --- Start1
    Width2 --- X
    Y1 --- Y2
    Start1 --- base1
    Start1 --- hints1
    X --- base2
    X --- hints2
    Y2 --- base3
    Y2 --- hints3
  
```

Width1 { }	hints { }	
	NameType	
	NameType	
	X { }	
	NameType	
	base float	
	hints { }	
	NameType	
	Y { }	
	NameType	
Width2 { }	NameType	
	X { }	
	NameType	
	base float	
	hints { }	
	NameType	
	Y { }	
	NameType	
	base float	
	hints { }	
Right { }	NameType	
	NameType	
	Start1 { }	
	NameType	
	X { }	
	NameType	
	base float	
	hints { }	
	NameType	
	Y { }	
Start2 { }	NameType	
	NameType	
	base float	
	hints { }	
	NameType	
	NameType	
	NameType	
	NameType	
	NameType	
	NameType	

X	{ }			
		Name	Type	
		base	float	
		hints	{ }	
		Name Type		
Y	{ }			
		Name	Type	
		base	float	
		hints	{ }	
		Name Type		
Width1 { }				
X	{ }			
		Name	Type	
		base	float	
		hints	{ }	
		Name Type		
Y	{ }			
		Name	Type	
		base	float	
		hints	{ }	
		Name Type		
Width2 { }				
X	{ }			
		Name	Type	
		base	float	
		hints	{ }	
		Name Type		
Y	{ }			
		Name	Type	
		base	float	
		hints	{ }	
		Name Type		

image.processing.blend.file.delete

Deletes a file with the given name.

MODELS All

Parameters

Name	Type
filename	string

This method has no return value.

image.processing.blend.file.list

Returns a list of available blend files

MODELS All

This method does not require any parameters.

Return value

Name	Type
filenames	[string]

image.processing.warp.file.delete

Deletes a file with the given name.

MODELS All

Parameters

Name	Type
filename	string

This method has no return value.

image.processing.warp.file.list

Returns a list of available warp files

MODELS All

This method does not require any parameters.

Return value

Name	Type
filenames	[string]

image.processing.warp.fourcorners.getscaledcorners

Get the corners scaled to the given resolution

MODELS All

Parameters

Name	Type				
resolution	{ }				
<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>		Name	Type	x	{ }
Name	Type				
x	{ }				

					<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type										
base	int										
hints	{ }										
				<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type					
Name	Type										
y	{ }										
				<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	
Name	Type										
base	int										
hints	{ }										
				<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type					
Name	Type										
Return value											

		<table><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	base	int	hints	{ }		
base	int							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
y	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
BottomRight	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>x</td><td>{ }</td></tr></table>	Name	Type	x	{ }		
Name	Type							
x	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							
y	{ }							
		<table><tr><td>Name</td><td>Type</td></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type							
base	int							
hints	{ }							
		<table><tr><td>Name</td><td>Type</td></tr></table>	Name	Type				
Name	Type							

image.processing.warp.warpscaledpoints

Takes an array of points and returns their warped equivalents.

MODELS All

Parameters

Name	Type						
points	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>X</td><td>{ }</td></tr></table>	Name	Type	X	{ }		
Name	Type						
X	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>Y</td><td>{ }</td></tr></table>	Name	Type	Y	{ }		
Name	Type						
Y	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						
resolution	{ }						
Name	Type						

X	{ }		
		Name	Type
		base	float
		hints	{ }
Y	{ }		
		Name	Type
		base	float
		hints	{ }
		Name	Type

Return value

		Name	Type
		points	[{ }]
		{ }	
		Name	Type
		X	{ }
		Name	Type
		base	float
		hints	{ }
		Name	Type
		Y	{ }
		Name	Type
		base	float
		hints	{ }
		Name	Type

image.processing.warpgrid.getgrid

Get the current grid points as normalized and relative

MODELS

All

This method does not require any parameters.

Return value

		Name	Type
		grid	[{ }]
		{ }	
		Name	Type
		x	{ }
		Name	Type
		base	float
		hints	{ }

		Name	Type
y	{ }		
		Name	Type
		base	float
		hints	{ }
		Name	Type

image.processing.warpgrid.getgridsize

Description not provided

MODELS All

This method does not require any parameters.

Return value

Name	Type
size	{ }
Name	Type
x	{ }
Name	Type
base	int
hints	{ }

image.processing.warpgrid.getscaledgrid

Get the current grid scaled to the given resolution

MODELS All

Parameters

Name	Type
resolution	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

Return value

Name	Type
grid	[{ }]

{ }

Name	Type
x	{ }

Name	Type
base	float
hints	{ }

Name	Type
------	------

y	{ }
---	-----

Name	Type
base	float
hints	{ }

Name	Type
------	------

image.resolution.list

List all possible resolutions.

MODELS All

This method does not require any parameters.

Return value

Name	Type
resolutions	[string]

image.source.l1displayport.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

Name	Type
connectors	[{ }]

{ }

Name	Type
------	------

gridposition { }													
<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>		Name	Type	row	{ }								
Name	Type												
row	{ }												
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
column { }													
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
view { }													
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
plane { }													
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
name	string												

image.source.l1hdbaset1.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

Name	Type								
connectors	[{ }]								
	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>gridposition</td><td>{ }</td></tr></table>	Name	Type	gridposition	{ }				
Name	Type								
gridposition	{ }								
	<table><tr><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table></td><td></td></tr></table>	<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>	Name	Type	row	{ }			
<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>	Name	Type	row	{ }					
Name	Type								
row	{ }								
	<table><tr><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td><td></td></tr></table>	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }			
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								

column	{ }		
		<u>Name</u>	<u>Type</u>
		base	int
		hints	{ }
view	{ }		
		<u>Name</u>	<u>Type</u>
		base	int
		hints	{ }
plane	{ }		
		<u>Name</u>	<u>Type</u>
		base	int
		hints	{ }
		<u>Name</u>	<u>Type</u>

image.source.l1hdbaset2.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

		Name	Type
		connectors	[{ }]
		{ }	
		Name	Type
		gridposition	{ }
		Name	Type
		row	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
		column	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
		view	{ }
		Name	Type

```

    base    int
    hints   { }

plane      { }

```

Name	Type
base	int
hints	{ }

```

    base    int
    hints   { }

```

Name	Type
base	int
hints	{ }

```
image.source.l1hdmi.listconnectors
```

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

Name	Type
connectors	[{ }]
	{ }
Name	Type
gridposition	{ }
Name	Type
row	{ }
Name	Type
base	int
hints	{ }
Name	Type
column	{ }
Name	Type
base	int
hints	{ }
Name	Type
view	{ }
Name	Type
base	int
hints	{ }
Name	Type
plane	{ }
Name	Type
base	int
hints	{ }

Name	Type
------	------

name	string
------	--------

image.source.l1quadsdi.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
connectors	[{ }]

{ }	
gridposition	{ }

row	{ }
-----	-----

base	int
hints	{ }

column	{ }
--------	-----

base	int
hints	{ }

view	{ }
------	-----

base	int
hints	{ }

plane	{ }
-------	-----

base	int
hints	{ }

Name	Type
------	------

name	string
------	--------

image.source.l1sdia.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS

 All

This method does not require any parameters.

Return value

Name	Type						
connectors	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>gridposition</td><td>{ }</td></tr></table>	Name	Type	gridposition	{ }		
Name	Type						
gridposition	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>	Name	Type	row	{ }		
Name	Type						
row	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>column</td><td>{ }</td></tr></table>	Name	Type	column	{ }		
Name	Type						
column	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>view</td><td>{ }</td></tr></table>	Name	Type	view	{ }		
Name	Type						
view	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>plane</td><td>{ }</td></tr></table>	Name	Type	plane	{ }		
Name	Type						
plane	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>name</td><td>string</td></tr></table>	Name	Type	name	string		
Name	Type						
name	string						

image.source.l1sdib.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

Name	Type		
connectors	[{ }]		
	{ }		
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
Name	Type		

gridposition { }													
<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>		Name	Type	row	{ }								
Name	Type												
row	{ }												
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
column { }													
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
view { }													
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
plane { }													
<table><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td colspan="2"><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }	<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }						
Name	Type												
base	int												
hints	{ }												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type										
Name	Type												
name	string												

image.source.l1sdic.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

Name	Type								
connectors	[{ }]								
	{ }								
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>gridposition</td><td>{ }</td></tr></table>	Name	Type	gridposition	{ }				
Name	Type								
gridposition	{ }								
	<table><tr><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table></td><td></td></tr></table>	<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>	Name	Type	row	{ }			
<table><tr><th>Name</th><th>Type</th></tr><tr><td>row</td><td>{ }</td></tr></table>	Name	Type	row	{ }					
Name	Type								
row	{ }								
	<table><tr><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td><td></td></tr></table>	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }			
Name	Type								
base	int								
hints	{ }								
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type						
Name	Type								

column	{ }		
		<u>Name</u>	<u>Type</u>
		base	int
		hints	{ }
view	{ }		
		<u>Name</u>	<u>Type</u>
		base	int
		hints	{ }
plane	{ }		
		<u>Name</u>	<u>Type</u>
		base	int
		hints	{ }
		<u>Name</u>	<u>Type</u>

image.source.l1sdid.listconnectors

Get all connectors that are assigned to this source with their layout position

MODELS All

This method does not require any parameters.

Return value

		Name	Type
		connectors	[{ }]
		{ }	
		Name	Type
		gridposition	{ }
		Name	Type
		row	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
		column	{ }
		Name	Type
		base	int
		hints	{ }
		Name	Type
		view	{ }
		Name	Type

```

base    int
hints   { }

Name    Type
-----
plane   { }

Name    Type
-----
base    int
hints   { }

Name    Type
-----
```

image.source.list

List all available sources

MODELS All

This method does not require any parameters.

Return value

Name	Type
sources	[string]

image.stereo.listdarktime

List all possible darktime values (in us).

MODELS All

This method does not require any parameters.

Return value

Name	Type
darktime	[{ }]
	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

image.testpattern.file.delete

Deletes a file with the given name.

MODELS All

Parameters

Name	Type
filename	string

This method has no return value.

image.testpattern.file.list

Get a list of available custom uploaded patterns

MODELS All

This method does not require any parameters.

Return value

Name	Type
patterns	[string]

image.testpattern.getProperties

Get the properties of a pattern

MODELS All

Parameters

Name	Type
id	string

Return value

Name	Type						
properties	[{ }]						
	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>	Name	Type	key	string	value	string
Name	Type						
key	string						
value	string						

image.testpattern.iterator.calibration.listpossiblepatterns

Return the list of possible testpatterns to iterate.

MODELS All

This method does not require any parameters.

Return value

Name	Type
list	[string]

image.testpattern.iterator.calibration.next

Show the next testpattern in the list.

MODELS All

This method does not require any parameters.

This method has no return value.

image.testpattern.iterator.iris.listpossiblepatterns

Return the list of possible testpatterns to iterate.

MODELS All

This method does not require any parameters.

Return value

Name	Type
list	[string]

image.testpattern.iterator.iris.next

Show the next testpattern in the list.

MODELS All

This method does not require any parameters.

This method has no return value.

image.testpattern.iterator.optics.listpossiblepatterns

Return the list of possible testpatterns to iterate.

MODELS All

This method does not require any parameters.

Return value

Name	Type
list	[string]

image.testpattern.iterator.optics.next

Show the next testpattern in the list.

MODELS All

This method does not require any parameters.

This method has no return value.

image.testpattern.iterator.shift.listpossiblepatterns

Return the list of possible testpatterns to iterate.

MODELS All

This method does not require any parameters.

Return value

Name	Type
list	[string]

image.testpattern.iterator.shift.next

Show the next testpattern in the list.

MODELS All

This method does not require any parameters.

This method has no return value.

image.testpattern.iterator.stereo.listpossiblepatterns

Return the list of possible testpatterns to iterate.

MODELS All

This method does not require any parameters.

Return value

Name	Type
list	[string]

image.testpattern.iterator.stereo.next

Show the next testpattern in the list.

MODELS All

This method does not require any parameters.

This method has no return value.

image.testpattern.list

Get a list of available patterns

MODELS All

This method does not require any parameters.

Return value

Name	Type
patterns	[{ }]

{ }

Name	Type
------	------

name	string
location	string
id	string
above	bool
internal	bool
properties	[{ }]

{ }

Name	Type
key	string
value	string

image.testpattern.listinternal

Get a list of available internal patterns

MODELS All

This method does not require any parameters.

Return value

Name	Type
patterns	[{ }]

{ }

Name	Type
name	string
location	string
id	string
above	bool
internal	bool
properties	[{ }]

{ }

Name	Type
key	string
value	string

image.testpattern.setproperties

Set the properties of a pattern

MODELS All

Parameters

Name	Type
id	string
properties	[{ }]

{ }

Name	Type
key	string
value	string

This method has no return value.

image.testpattern.toggle

Toggle display of selected test pattern on/off

MODELS All

This method does not require any parameters.

This method has no return value.

image.window.list

List all available windows

MODELS All

This method does not require any parameters.

Return value

Name	Type
windows	[string]

keydispatcher.postevent

Create and send a synthetic key event

MODELS All

Parameters

Name	Type
event type	enum
	Values
	"PRESS"
	"RELEASE"
	"REPEAT"
	"CLICKED"
key	enum
	Values
	"RC_SHUTTER_OPEN"
	"RC_SHUTTER_CLOSE"
	"RC_POWER_ON"
	"RC_POWER_OFF"
	"RC OSD"

"RC_LCD"
"RC_PATTERN"
"RC_RGB"
"RC_ZOOM_PLUS"
"RC_ZOOM_MINUS"
"RC_SHIFT_LEFT"
"RC_SHIFT_UP"
"RC_SHIFT_RIGHT"
"RC_SHIFT_DOWN"
"RC_FOCUS_PLUS"
"RC_FOCUS_MINUS"
"RC_MENU"
"RC_DEFAULT"
"RC_BACK"
"RC_UP"
"RC_LEFT"
"RC_OK"
"RC_RIGHT"
"RC_DOWN"
"RC_ADDRESS"
"RC_INPUT"
"RC_MACRO"
"RC_1 "
"RC_2 "
"RC_3 "
"RC_4 "
"RC_5 "
"RC_6 "
"RC_7 "
"RC_8 "
"RC_9 "
"RC_0 "
"RC_ASTERISK"
"RC_NUMBER"
"KP_LEFT"
"KP_UP"
"KP_OK"
"KP_RIGHT"
"KP_DOWN"
"KP_MENU"
"KP_POWER"
"KP_BACK"
"KP_OSD"

```
"KP_LENS"  
"KP_PATTERN"  
"KP_SHUTTER"  
"KP_INPUT"  
"KP_STANDBY"  
"TP_TOUCH"
```

This method has no return value.

led.activity

Activates the LEDS when enabled and restarts the LED timeout counter

MODELS

All

This method does not require any parameters.

This method has no return value.

led.list

Description not provided

MODELS

All

This method does not require any parameters.

Return value

Name	Type
leds	[string]

lightmeasurement.getlightoutput

Description not provided

MODELS

All

This method does not require any parameters.

Return value

Name	Type						
lumens	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

network.device.wifi.connect

Description not provided

MODELS

All

Parameters

Name	Type
ssid	string
security	enum
	Values "NONE" "WEP" "WPA" "WPA2"

This method has no return value.

network.device.wifi.getaccesspoints

Get list of available access points for this device

MODELS

All

This method does not require any parameters.

Return value

Name	Type																		
value	[{ }]																		
	{ }																		
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>SSID</td><td>string</td></tr><tr><td>Strength</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td>security</td><td>enum</td></tr><tr><td></td><td> Values "NONE" "WEP" "WPA" "WPA2" </td></tr></table>	Name	Type	SSID	string	Strength	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }	security	enum		Values "NONE" "WEP" "WPA" "WPA2"
Name	Type																		
SSID	string																		
Strength	{ }																		
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }												
Name	Type																		
base	int																		
hints	{ }																		
security	enum																		
	Values "NONE" "WEP" "WPA" "WPA2"																		

network.device.wifi.getsecurity

Get the security protocol for the specified access point

MODELS

 All

Parameters

Name	Type
ssid	string

Return value

Name	Type
security	enum
Values	
"NONE"	
"WEP"	
"WPA"	
"WPA2"	

network.device.wifi.requestscan

Request a rescan of access points.

MODELS

 All

This method does not require any parameters.

This method has no return value.

network.list

List of logical device id, e.g: 'wired1', 'wifi1'

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
devices	[string]

notification.caution.list

Retrieve all active notifications for the given severity

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
notifications	[{ }]
{ }	

Name	Type
severity	enum
Values	
"INFO"	
"CAUTION"	
"WARNING"	
"ERROR"	
id	string
code	string
timestamp	string
message	string
timeout	string
visibility	string
doc	string
actions	[enum]
Values	
"NONE"	
"OK"	
"CANCEL"	
"YES"	
"NO"	
"SUPPRESS"	

notification.dismiss

Dismiss the notification with the specified id

MODELS All

Parameters

Name	Type
id	string
response	enum
Values	
"NONE"	
"OK"	
"CANCEL"	
"YES"	
"NO"	
"SUPPRESS"	

This method has no return value.

notification.error.list

Retrieve all active notifications for the given severity

MODELS

All

This method does not require any parameters.

Return value

Name	Type																																				
notifications	[{ }]																																				
	{ }																																				
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>severity</td><td>enum</td></tr><tr><td></td><td><table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table></td></tr><tr><td>id</td><td>string</td></tr><tr><td>code</td><td>string</td></tr><tr><td>timestamp</td><td>string</td></tr><tr><td>message</td><td>string</td></tr><tr><td>timeout</td><td>string</td></tr><tr><td>visibility</td><td>string</td></tr><tr><td>doc</td><td>string</td></tr><tr><td>actions</td><td>[enum]</td></tr><tr><td></td><td><table><tr><th>Values</th></tr><tr><td>"NONE"</td></tr><tr><td>"OK"</td></tr><tr><td>"CANCEL "</td></tr><tr><td>"YES"</td></tr><tr><td>"NO"</td></tr><tr><td>"SUPPRESS"</td></tr></table></td></tr></table>	Name	Type	severity	enum		<table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table>	Values	"INFO"	"CAUTION"	"WARNING"	"ERROR"	id	string	code	string	timestamp	string	message	string	timeout	string	visibility	string	doc	string	actions	[enum]		<table><tr><th>Values</th></tr><tr><td>"NONE"</td></tr><tr><td>"OK"</td></tr><tr><td>"CANCEL "</td></tr><tr><td>"YES"</td></tr><tr><td>"NO"</td></tr><tr><td>"SUPPRESS"</td></tr></table>	Values	"NONE"	"OK"	"CANCEL "	"YES"	"NO"	"SUPPRESS"
Name	Type																																				
severity	enum																																				
	<table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table>	Values	"INFO"	"CAUTION"	"WARNING"	"ERROR"																															
Values																																					
"INFO"																																					
"CAUTION"																																					
"WARNING"																																					
"ERROR"																																					
id	string																																				
code	string																																				
timestamp	string																																				
message	string																																				
timeout	string																																				
visibility	string																																				
doc	string																																				
actions	[enum]																																				
	<table><tr><th>Values</th></tr><tr><td>"NONE"</td></tr><tr><td>"OK"</td></tr><tr><td>"CANCEL "</td></tr><tr><td>"YES"</td></tr><tr><td>"NO"</td></tr><tr><td>"SUPPRESS"</td></tr></table>	Values	"NONE"	"OK"	"CANCEL "	"YES"	"NO"	"SUPPRESS"																													
Values																																					
"NONE"																																					
"OK"																																					
"CANCEL "																																					
"YES"																																					
"NO"																																					
"SUPPRESS"																																					

notification.info.list

Retrieve all active notifications for the given severity

MODELS

All

This method does not require any parameters.

Return value

Name	Type				
notifications	[{ }]				
	{ }				
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>severity</td><td>enum</td></tr></table>	Name	Type	severity	enum
Name	Type				
severity	enum				

		Values
		"INFO"
		"CAUTION"
		"WARNING"
		"ERROR"
id	string	
code	string	
timestamp	string	
message	string	
timeout	string	
visibility	string	
doc	string	
actions	[enum]	
		Values
		"NONE"
		"OK"
		"CANCEL"
		"YES"
		"NO"
		"SUPPRESS"

notification.list

List all active notifications

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
notifications	[{ }]
	{ }
Name	Type
severity	enum
	</

message	string
timeout	string
visibility	string
doc	string
actions	[enum]

Values
"NONE"
"OK"
"CANCEL"
"YES"
"NO"
"SUPPRESS"

notification.listsuppressed

Get a list of suppressed notification codes

MODELS All

This method does not require any parameters.

Return value

Name	Type
list	[string]

notification.log

List received notifications

MODELS All

Parameters

Name	Type						
minimumseverity	enum						
	<table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table>	Values	"INFO"	"CAUTION"	"WARNING"	"ERROR"	
Values							
"INFO"							
"CAUTION"							
"WARNING"							
"ERROR"							
start	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						
count	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

base	int
hints	{ }

Name	Type
------	------

Return value

Name	Type
notifications	[{ }]

{ }

Name	Type
severity	enum

Values

- "INFO"
- "CAUTION"
- "WARNING"
- "ERROR"

id	string
code	string
timestamp	string
arguments	[string]
message	string
timeout	string
visibility	string
doc	string
actions	[enum]

Values

- "NONE"
- "OK"
- "CANCEL "
- "YES"
- "NO"
- "SUPPRESS"

notification.suppress

Add a notification code to suppress (log but do not show on the LCD/OSD)

MODELS All

Parameters

Name	Type
code	string

This method has no return value.

notification.unsuppress

No longer suppress a certain notification code

MODELS

All

Parameters

Name	Type
code	string

This method has no return value.

notification.unsuppressall

No longer suppress any notification codes

MODELS

All

This method does not require any parameters.

This method has no return value.

notification.warning.list

Retrieve all active notifications for the given severity

MODELS

All

This method does not require any parameters.

Return value

Name	Type																											
notifications	[{ }]																											
	{ }																											
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>severity</td><td>enum</td></tr><tr><td></td><td><table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table></td></tr><tr><td>id</td><td>string</td></tr><tr><td>code</td><td>string</td></tr><tr><td>timestamp</td><td>string</td></tr><tr><td>message</td><td>string</td></tr><tr><td>timeout</td><td>string</td></tr><tr><td>visibility</td><td>string</td></tr><tr><td>doc</td><td>string</td></tr><tr><td>actions</td><td>[enum]</td></tr></table>	Name	Type	severity	enum		<table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table>	Values	"INFO"	"CAUTION"	"WARNING"	"ERROR"	id	string	code	string	timestamp	string	message	string	timeout	string	visibility	string	doc	string	actions	[enum]
Name	Type																											
severity	enum																											
	<table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table>	Values	"INFO"	"CAUTION"	"WARNING"	"ERROR"																						
Values																												
"INFO"																												
"CAUTION"																												
"WARNING"																												
"ERROR"																												
id	string																											
code	string																											
timestamp	string																											
message	string																											
timeout	string																											
visibility	string																											
doc	string																											
actions	[enum]																											
	<table><tr><th>Values</th></tr></table>	Values																										
Values																												

"NONE "

"OK "

"CANCEL "

"YES "

"NO "

"SUPPRESS "

optics.autofocus.calibration.calibrate

Calibration should be done 2 times: with light content and with dark content. Call Calibrate only after this content has been displayed for a duration equal to the integration interval of the lens.

MODELS

 All

This method does not require any parameters.

This method has no return value.

optics.focus.calibrate

Calibrate motor

MODELS

 All

This method does not require any parameters.

This method has no return value.

optics.focus.runforward

Run forward

MODELS

 All

This method does not require any parameters.

This method has no return value.

optics.focus.runforwardtime

Run forward for X milliseconds

MODELS

 All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.focus.runreverse

Run reverse

MODELS All

This method does not require any parameters.

This method has no return value.

optics.focus.runreversetime

Run reverse for X milliseconds

MODELS All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.focus.stepforward

Step forward

MODELS All

Parameters

Name	Type
steps	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.focus.stepreverse

Step reverse

MODELS All

Parameters

Name	Type
steps	{ }

Name	Type
base	int
hints	{ }

Name	Type
------	------

This method has no return value.

optics.focus.stop

Stop

MODELS All

This method does not require any parameters.

This method has no return value.

optics.lens.list

List available lenses

MODELS All

This method does not require any parameters.

Return value

Name	Type
availablelenses	[{ }]

{ }

Name	Type
Id	string
Description	string

optics.lensshift.horizontal.calibrate

Calibrate motor

MODELS All

This method does not require any parameters.

This method has no return value.

optics.lensshift.horizontal.runforward

Run forward

MODELS

All

This method does not require any parameters.

This method has no return value.

optics.lensshift.horizontal.runforwardtime

Run forward for X milliseconds

MODELS

All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.lensshift.horizontal.runreverse

Run reverse

MODELS

All

This method does not require any parameters.

This method has no return value.

optics.lensshift.horizontal.runreversetime

Run reverse for X milliseconds

MODELS

All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.lensshift.horizontal.stepforward

Step forward

MODELS

All

Parameters

Name	Type
steps	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.lensshift.horizontal.stepreverse

Step reverse

MODELS

All

Parameters

Name	Type
steps	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.lensshift.horizontal.stop

Stop

MODELS

All

This method does not require any parameters.

This method has no return value.

optics.lensshift.vertical.calibrate

Calibrate motor

MODELS

All

This method does not require any parameters.

This method has no return value.

optics.lensshift.vertical.runforward

Run forward

MODELS All

This method does not require any parameters.

This method has no return value.

optics.lensshift.vertical.runforwardtime

Run forward for X milliseconds

MODELS All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.lensshift.vertical.runreverse

Run reverse

MODELS All

This method does not require any parameters.

This method has no return value.

optics.lensshift.vertical.runreversetime

Run reverse for X milliseconds

MODELS All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.lensshift.vertical.stepforward

Step forward

MODELS All

Parameters

Name	Type						
steps	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

This method has no return value.

optics.lensshift.vertical.stepreverse

Step reverse

MODELS All

Parameters

Name	Type						
steps	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

This method has no return value.

optics.lensshift.vertical.stop

Stop

MODELS All

This method does not require any parameters.

This method has no return value.

optics.shifftocenter

Shift lens to center of allowed shift range

MODELS All

This method does not require any parameters.

This method has no return value.

optics.shutter.getobjectpath

Get object path of shutter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
path	string

optics.shutter.getstate

Get state of shutter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
state	enum
Values	
"Stopped"	
"Running"	
"Calibrating"	

optics.shutter.toggle

Toggle shutter position

MODELS

 All

This method does not require any parameters.

This method has no return value.

optics.zoom.calibrate

Calibrate motor

MODELS

 All

This method does not require any parameters.

This method has no return value.

optics.zoom.runforward

Run forward

MODELS

All

This method does not require any parameters.

This method has no return value.

optics.zoom.runforwardtime

Run forward for X milliseconds

MODELS

All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.zoom.runreverse

Run reverse

MODELS

All

This method does not require any parameters.

This method has no return value.

optics.zoom.runreversetime

Run reverse for X milliseconds

MODELS

All

Parameters

Name	Type
milliseconds	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.zoom.stepforward

Step forward

MODELS

All

Parameters

Name	Type
steps	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.zoom.stepreverse

Step reverse

MODELS

All

Parameters

Name	Type
steps	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

optics.zoom.stop

Stop

MODELS

All

This method does not require any parameters.

This method has no return value.

peripheral.autodetect

Trigger auto detection of connected peripherals.

MODELS

All

This method does not require any parameters.

This method has no return value.

peripheral.list

List alignment tools.

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
modes	[string]

peripheral.ranging.getdistance

Distance in meter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type						
modes	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	float	hints	{ }
Name	Type						
base	float						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

profile.activatepreset

Activate the profile associated with the given preset.

MODELS

 All

Parameters

Name	Type						
preset	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

Return value

Name	Type
result	bool

profile.activateprofile

Activate the given profile

MODELS

All

Parameters

Name	Type
name	string

Return value

Name	Type
result	bool

profile.assignpreset

Assign a profile to a preset.

MODELS

All

Parameters

Name	Type						
preset	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
profile	string						

Return value

Name	Type
result	bool

profile.createprofile

Create a profile containing the current values of the listed domains.

MODELS

All

Parameters

Name	Type
profilename	string
domains	[string]

Return value

Name	Type
result	bool

profile.deleteprofile

Delete the given profile

MODELS

All

Parameters

Name	Type
name	string

Return value

Name	Type
result	bool

profile.getdomainsforprofile

Get the list of domains stored in a profile.

MODELS

All

Parameters

Name	Type
name	string

Return value

Name	Type
domains	[string]

profile.presetforprofile

Get the preset assignment for given profile. -1 means unassigned.

MODELS

All

Parameters

Name	Type
name	string

Return value

Name	Type						
result	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

profile.profileforpreset

Get the profile associated with the given preset assignment. Empty string means unassigned.

MODELS

 All

Parameters

Name	Type						
preset	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

Return value

Name	Type
result	string

remotecontrol.listsensors

Return a list of all the object names of the IR sensors

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
sensors	[string]

scheduler.create

Create a new action for scheduler.

MODELS

 All

Parameters

Name	Type
name	string

Return value

Name	Type
path	string

statistics.laser plate 1 bank 1 runtime.getname

Name of the counter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 1 bank 1 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 1 bank 2 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 1 bank 2 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	

"none"

"hours"

"minutes"

"seconds"

"number"

"percent"

statistics.laser plate 1 bank 3 runtime.getname

Name of the counter

MODELSAll

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 1 bank 3 runtime.getunit

Unit of measurements

MODELSAll

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none" "hours" "minutes" "seconds" "number" "percent"	

statistics.laser plate 2 bank 1 runtime.getname

Name of the counter

MODELSAll

This method does not require any parameters.

Return value

Name	Type
------	------

countername string

statistics.laser plate 2 bank 1 runtime.getunit

Unit of measurements

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 2 bank 2 runtime.getname

Name of the counter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 2 bank 2 runtime.getunit

Unit of measurements

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	

"seconds"
"number"
"percent"

statistics.laser plate 2 bank 3 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 2 bank 3 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 3 bank 1 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 3 bank 1 runtime.getunit

Unit of measurements

MODELS

All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 3 bank 2 runtime.getname

Name of the counter

MODELS

All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 3 bank 2 runtime.getunit

Unit of measurements

MODELS

All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 3 bank 3 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 3 bank 3 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 4 bank 1 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 4 bank 1 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 4 bank 2 runtime.getname

Name of the counter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 4 bank 2 runtime.getunit

Unit of measurements

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 4 bank 3 runtime.getname

Name of the counter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 4 bank 3 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laser plate 5 bank 1 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 5 bank 1 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum

Values
"none"
"hours"
"minutes"
"seconds"
"number"
"percent"

statistics.laser plate 5 bank 2 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 5 bank 2 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type							
unit	enum							
	<table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table>	Values	"none"	"hours"	"minutes"	"seconds"	"number"	"percent"
Values								
"none"								
"hours"								
"minutes"								
"seconds"								
"number"								
"percent"								

statistics.laser plate 5 bank 3 runtime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laser plate 5 bank 3 runtime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.laserruntime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.laserruntime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	

"minutes"
"seconds"
"number"
"percent"

statistics.listcounters

List all counter names

MODELS All

This method does not require any parameters.

Return value

Name	Type																													
counterlist	[{ }]																													
	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>name</td><td>string</td></tr><tr><td>value</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>unit</td><td>enum</td></tr><tr><td></td><td><table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table></td></tr></table></td></tr></table>	Name	Type	name	string	value	{ }		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>unit</td><td>enum</td></tr><tr><td></td><td><table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	unit	enum		<table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table>	Values	"none"	"hours"	"minutes"	"seconds"	"number"	"percent"
Name	Type																													
name	string																													
value	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>unit</td><td>enum</td></tr><tr><td></td><td><table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table></td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	unit	enum		<table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table>	Values	"none"	"hours"	"minutes"	"seconds"	"number"	"percent"								
Name	Type																													
base	int																													
hints	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																											
Name	Type																													
unit	enum																													
	<table><tr><th>Values</th></tr><tr><td>"none"</td></tr><tr><td>"hours"</td></tr><tr><td>"minutes"</td></tr><tr><td>"seconds"</td></tr><tr><td>"number"</td></tr><tr><td>"percent"</td></tr></table>	Values	"none"	"hours"	"minutes"	"seconds"	"number"	"percent"																						
Values																														
"none"																														
"hours"																														
"minutes"																														
"seconds"																														
"number"																														
"percent"																														

statistics.projectorruntime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.projectorruntime.getunit

Unit of measurements

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.systemtime.getname

Name of the counter

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.systemtime.getunit

Unit of measurements

MODELS

 All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

statistics.uptime.getname

Name of the counter

MODELS All

This method does not require any parameters.

Return value

Name	Type
countername	string

statistics.uptime.getunit

Unit of measurements

MODELS All

This method does not require any parameters.

Return value

Name	Type
unit	enum
Values	
"none"	
"hours"	
"minutes"	
"seconds"	
"number"	
"percent"	

system.activity

Signal user activity (resets timeout countdown timers)

MODELS All

This method does not require any parameters.

This method has no return value.

system.boards.getboardlist

Retrieve list of detected boards

MODELS All

This method does not require any parameters.

Return value

Name	Type
boards	[string]

system.boards.getmissingboardlist

Retrieve list of missing boards

MODELS All

This method does not require any parameters.

Return value

Name	Type
boards	[string]

system.date.getlocaldate

Returns the system date as local time.

MODELS All

This method does not require any parameters.

Return value

Name	Type																																																												
date	{ }																																																												
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>year</td><td>{ }</td></tr></table>	Name	Type	year	{ }																																																								
Name	Type																																																												
year	{ }																																																												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>month</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>day</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>hour</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table></td></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	month	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>day</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>hour</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	day	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>hour</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	hour	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																						
Name	Type																																																												
base	int																																																												
hints	{ }																																																												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																										
Name	Type																																																												
month	{ }																																																												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>day</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>hour</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	day	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>hour</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	hour	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																						
Name	Type																																																												
base	int																																																												
hints	{ }																																																												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																										
Name	Type																																																												
day	{ }																																																												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>hour</td><td>{ }</td></tr><tr><td></td><td><table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	hour	{ }		<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																						
Name	Type																																																												
base	int																																																												
hints	{ }																																																												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																										
Name	Type																																																												
hour	{ }																																																												
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }																																																						
Name	Type																																																												
base	int																																																												
hints	{ }																																																												
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																																																										
Name	Type																																																												

	<table><tr><th>Name</th><th>Type</th></tr><tr><td>minute</td><td>{ }</td></tr></table>	Name	Type	minute	{ }		
Name	Type						
minute	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>second</td><td>{ }</td></tr></table>	Name	Type	second	{ }		
Name	Type						
second	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

system.date.getsystemdate

Returns the system date as UTC time.

MODELS All

This method does not require any parameters.

Return value

	<table><tr><th>Name</th><th>Type</th></tr><tr><td>date</td><td>{ }</td></tr></table>	Name	Type	date	{ }		
Name	Type						
date	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>year</td><td>{ }</td></tr></table>	Name	Type	year	{ }		
Name	Type						
year	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>month</td><td>{ }</td></tr></table>	Name	Type	month	{ }		
Name	Type						
month	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>day</td><td>{ }</td></tr></table>	Name	Type	day	{ }		
Name	Type						
day	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>hour</td><td>{ }</td></tr></table>	Name	Type	hour	{ }		
Name	Type						
hour	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

minute	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						
second	{ }						
<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>		Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
<table><tr><th>Name</th><th>Type</th></tr></table>		Name	Type				
Name	Type						

system.getidentification

Description not provided

MODELS All

Parameters

Name	Type
identification	string

Return value

Name	Type
value	string

system.getidentifications

Description not provided

MODELS All

This method does not require any parameters.

Return value

Name	Type
identification	[{ }]

Name	Type
key	string
value	string

system.getsystemdate

Description not provided

MODELS All

Parameters

Name	Type
------	------

Return value

Name	Type
------	------

system.gotoeco

Set the device in the eco state

MODELS All

This method does not require any parameters.

This method has no return value.

system.gotoready

Set the device from standby state to ready state

MODELS All

This method does not require any parameters.

This method has no return value.

system.license.file.saveregistrationtousb

Save registration file to usb memory device

MODELS All

This method does not require any parameters.

This method has no return value.

system.license.file.uploadfromusb

Upload license from usb memory device

MODELS All

This method does not require any parameters.

This method has no return value.

system.listresetdomains

Returns the list of available reset domains

MODELS All

This method does not require any parameters.

Return value

Name	Type
domains	[enum]
Values	
"Dmx"	
"Environment"	
"Gpio"	
"Gsm"	
"Illumination"	
"ImageBlend"	
"ImageCapture"	
"ImageConnector"	
"ImageConvergence"	
"ImageDisplay"	
"ImageFeatures"	
"ImageOrientation"	
"ImageRealColor"	
"ImageResolution"	
"ImageSource"	
"ImageStereo"	
"ImageTestPattern"	
"ImageUniformity"	
"ImageWarp"	
"LightMeasurement"	
"LightSteering"	
"LinkedServices"	
"Macro"	
"MediaPlayer"	
"Network"	
"Optics"	
"Peripheral"	
"Profiles"	
"Scheduler"	
"Screen"	
"Snmp"	
"System"	
"Time"	
"User"	
"UserInterface"	

system.poweroff

Power off the unit

MODELS

All

This method does not require any parameters.

This method has no return value.

system.poweron

Power on the unit

MODELS

All

This method does not require any parameters.

This method has no return value.

system.reset

Asynchronously starts reset of selected domains. The completion of the domains are signalled by one ore more 'Performed'-signals. Subsequent calls to 'ResetAll' or 'Reset' will fail until all domains have completed.

MODELS

All

Parameters

Name	Type
domains	[enum]
Values	
"Dmx"	
"Environment"	
"Gpio"	
"Gsm"	
"Illumination"	
"ImageBlend"	
"ImageCapture"	
"ImageConnector"	
"ImageConvergence"	
"ImageDisplay"	
"ImageFeatures"	
"ImageOrientation"	
"ImageRealColor"	
"ImageResolution"	
"ImageSource"	
"ImageStereo"	
"ImageTestPattern"	
"ImageUniformity"	
"ImageWarp"	
"LightMeasurement"	

"LightSteering"
"LinkedServices"
"Macro"
"MediaPlayer"
"Network"
"Optics"
"Peripheral"
"Profiles"
"Scheduler"
"Screen"
"Sntp"
"System"
"Time"
"User"
"UserInterface"

This method has no return value.

system.resetall

Asynchronously starts reset of all domains. The completion of the domains are signalled by one ore more 'Performed'-signals. Subsequent calls to 'ResetAll' or 'Reset' will fail until all domains have completed.

MODELS All

This method does not require any parameters.

Return value

Name	Type
domains	[enum]
Values	
"Dmx"	
"Environment"	
"Gpio"	
"Gsm"	
"Illumination"	
"ImageBlend"	
"ImageCapture"	
"ImageConnector"	
"ImageConvergence"	
"ImageDisplay"	
"ImageFeatures"	
"ImageOrientation"	
"ImageRealColor"	
"ImageResolution"	

"ImageSource"
"ImageStereo"
"ImageTestPattern"
"ImageUniformity"
"ImageWarp"
"LightMeasurement"
"LightSteering"
"LinkedServices"
"Macro"
"MediaPlayer"
"Network"
"Optics"
"Peripheral"
"Profiles"
"Scheduler"
"Screen"
"Snmp"
"System"
"Time"
"User"
"UserInterface"

ui.leds.setstate

Set the state and color of the key LEDs

MODELS

All

Parameters

Name	Type
key	enum
Values	
"Back"	
"Down"	
"Input"	
"Left"	
"Lens"	
"Menu"	
"Ok"	
"OsdOff"	
"Pattern"	
"Right"	
"Up"	
state	enum

Values
"Off"
"White"
"Blue"

This method has no return value.

ui.settings.get

Get the value of the specified key

MODELS

 All

Parameters

Name	Type
key	string

Return value

Name	Type
value	string

ui.settings.getfonticons

Return a dictionary of icons for the specified category. Then icon is returned as a string containing the font set class name and the icon class name.

MODELS

 All

Parameters

Name	Type				
category	enum				
<table><tr><th>Values</th></tr><tr><td>"Source"</td></tr><tr><td>"Connector"</td></tr><tr><td>"TestPattern"</td></tr></table>		Values	"Source"	"Connector"	"TestPattern"
Values					
"Source"					
"Connector"					
"TestPattern"					

Return value

Name	Type						
dictionary	[{ }]						
{ }							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>		Name	Type	key	string	value	string
Name	Type						
key	string						
value	string						

ui.settings.geticons

Return a dictionary of icons for the specified category. The icon is return as a SVG sprite name.

MODELS

All

Parameters

Name	Type
category	enum
Values	
"Source"	
"Connector"	
"TestPattern"	

Return value

Name	Type						
dictionary	[{ }]						
{ }							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>		Name	Type	key	string	value	string
Name	Type						
key	string						
value	string						

ui.settings.keys

Return a list of all the keys

MODELS

All

This method does not require any parameters.

Return value

Name	Type
keys	[string]

ui.settings.list

Return a list of key/value pairs of all the settings

MODELS

All

This method does not require any parameters.

Return value

Name	Type						
dict	[{ }]						
{ }							
<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>		Name	Type	key	string	value	string
Name	Type						
key	string						
value	string						

ui.settings.remove

Remove the specified key and value

MODELS All

Parameters

Name	Type
key	string

This method has no return value.

ui.settings.set

Set the key to the specified value

MODELS All

Parameters

Name	Type
key	string
value	string

This method has no return value.

ui.togglelensmenu

Toggle lens menu on or off

MODELS All

This method does not require any parameters.

This method has no return value.

ui.togglemenu

Toggle menu on/off

MODELS All

This method does not require any parameters.

This method has no return value.

ui.toggleosd

Toggle on screen display on or off

MODELS All

This method does not require any parameters.

This method has no return value.

ui.togglepatternmenu

Toggle the pattern menu on or off

MODELS

 All

This method does not require any parameters.

This method has no return value.

ui.togglesourcemenu

Toggle the source menu on or off

MODELS

 All

This method does not require any parameters.

This method has no return value.

ui.togglestealthmode

This method is depreciated.

MODELS

 All

This method does not require any parameters.

This method has no return value.

ui.touchscreen.tapped

Called from touchpanel when the touchpanel is in Test mode

MODELS

 All

Parameters

Name	Type
x	{ }
Name Type	
base	int
hints	{ }
Name Type	
y	{ }
Name Type	
base	int
hints	{ }
Name Type	

This method has no return value.

user.admin.changegroup

Change the user group

MODELS All

Parameters

Name	Type
newgroup	string

This method has no return value.

user.admin.changepassword

Change the password

MODELS All

Parameters

Name	Type
newpassword	string

This method has no return value.

user.admin.changepincode

Change the 6-digit PIN code

MODELS All

Parameters

Name	Type						
newpincode	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

This method has no return value.

user.admin.changepublickey

Change the public key

MODELS All

Parameters

Name	Type
newpublickey	string

This method has no return value.

user.admin.changeusername

Change the username

MODELS All

Parameters

Name	Type
newusername	string

Return value

Name	Type
newuserslug	string

user.admin.disable

Disable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.admin.enable

Enable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.admin.invalidatetokens

Invalidate all tokens

MODELS All

This method does not require any parameters.

This method has no return value.

user.admin.logouteverywhere

Invalidate all tokens and log out of all active sessions

MODELS All

This method does not require any parameters.

This method has no return value.

user.admin.removepassword

Remove the password

MODELS All

This method does not require any parameters.

This method has no return value.

user.admin.removepincode

Remove the 6-digit PIN code

MODELS All

This method does not require any parameters.

This method has no return value.

user.currentuser.changegroup

Change the user group

MODELS All

Parameters

Name	Type
newgroup	string

This method has no return value.

user.currentuser.changepassword

Change the password

MODELS All

Parameters

Name	Type
newpassword	string

This method has no return value.

user.currentuser.changepincode

Change the 6-digit PIN code

MODELS All

Parameters

Name	Type						
newpincode	{ }						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>int</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	int	hints	{ }
Name	Type						
base	int						
hints	{ }						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

This method has no return value.

user.currentuser.changepublickey

Change the public key

MODELS All

Parameters

Name	Type
newpublickey	string

This method has no return value.

user.currentuser.changeusername

Change the username

MODELS All

Parameters

Name	Type
newusername	string

Return value

Name	Type
newuserslug	string

user.currentuser.disable

Disable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.currentuser.enable

Enable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.currentuser.invalidatetokens

Invalidate all tokens

MODELS All

This method does not require any parameters.

This method has no return value.

user.currentuser.logouteverywhere

Invalidate all tokens and log out of all active sessions

MODELS All

This method does not require any parameters.

This method has no return value.

user.currentuser.removepassword

Remove the password

MODELS All

This method does not require any parameters.

This method has no return value.

user.currentuser.removepincode

Remove the 6-digit PIN code

MODELS All

This method does not require any parameters.

This method has no return value.

user.poweruser.changegroup

Change the user group

MODELS All

Parameters

Name	Type
------	------

`newgroup` `string`

This method has no return value.

`user.poweruser.changepassword`

Change the password

MODELS

All

Parameters

Name	Type
<code>newpassword</code>	<code>string</code>

This method has no return value.

`user.poweruser.changepincode`

Change the 6-digit PIN code

MODELS

All

Parameters

Name	Type						
<code>newpincode</code>	<code>{ }</code>						
	<table><tr><th>Name</th><th>Type</th></tr><tr><td><code>base</code></td><td><code>int</code></td></tr><tr><td><code>hints</code></td><td><code>{ }</code></td></tr></table>	Name	Type	<code>base</code>	<code>int</code>	<code>hints</code>	<code>{ }</code>
Name	Type						
<code>base</code>	<code>int</code>						
<code>hints</code>	<code>{ }</code>						
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type				
Name	Type						

This method has no return value.

`user.poweruser.changepublickey`

Change the public key

MODELS

All

Parameters

Name	Type
<code>newpublickey</code>	<code>string</code>

This method has no return value.

`user.poweruser.changeusername`

Change the username

MODELS

All

Parameters

Name	Type
newusername	string

Return value

Name	Type
newuserslug	string

user.poweruser.disable

Disable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.poweruser.enable

Enable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.poweruser.invalidatetokens

Invalidate all tokens

MODELS All

This method does not require any parameters.

This method has no return value.

user.poweruser.logouteverywhere

Invalidate all tokens and log out of all active sessions

MODELS All

This method does not require any parameters.

This method has no return value.

user.poweruser.removepassword

Remove the password

MODELS All

This method does not require any parameters.

This method has no return value.

user.poweruser.removepincode

Remove the 6-digit PIN code

MODELS All

This method does not require any parameters.

This method has no return value.

user.resetadministrator

Reset the administrator account using the response provided by barco

MODELS All

Parameters

Name	Type
response	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

user.user.changegroup

Change the user group

MODELS All

Parameters

Name	Type
newgroup	string

This method has no return value.

user.user.changepassword

Change the password

MODELS All

Parameters

Name	Type
newpassword	string

This method has no return value.

user.user.changepincode

Change the 6-digit PIN code

MODELS All

Parameters

Name	Type
newpincode	{ }
Name	Type
base	int
hints	{ }
Name	Type

This method has no return value.

user.user.changepublickey

Change the public key

MODELS All

Parameters

Name	Type
newpublickey	string

This method has no return value.

user.user.changeusername

Change the username

MODELS All

Parameters

Name	Type
newusername	string

Return value

Name	Type
newuserslug	string

user.user.disable

Disable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.user.enable

Enable the user

MODELS All

This method does not require any parameters.

This method has no return value.

user.user.invalidatetokens

Invalidate all tokens

MODELS All

This method does not require any parameters.

This method has no return value.

user.user.logouteverywhere

Invalidate all tokens and log out of all active sessions

MODELS All

This method does not require any parameters.

This method has no return value.

user.user.removepassword

Remove the password

MODELS All

This method does not require any parameters.

This method has no return value.

user.user.removepincode

Remove the 6-digit PIN code

MODELS All

This method does not require any parameters.

This method has no return value.

Signals

Alphabetical list of all signals

loggedout

Will be raised whenever the user is logged out

MODELSAll

Name	Type	Description
reason	enum	
Values		
"Disable"		
"InternalError"		
"LogoutEverywhere"		
"Reauthentication"		
"Reboot"		
"Remove"		
"Request"		
"Reset"		
"Restore"		

modelupdated

Signals whenever functionality for an object appears or disappears

MODELSAll

Name	Type	Description
object	string	The object name (JSON-RPC dot-notation)
newobject	bool	True if functionality is added and false if functionality is removed
reason	string	The reason for this ModelUpdated signal, either an object was added, removed or changed.
accesslevel	enum	Minimum access level for this object
Values		
"UNAUTHENTICATED_END_USER"		
"END_USER"		
"POWER_USER"		
"ADMINISTRATOR"		
"SERVICE_PARTNER"		
"MANUFACTURING"		
"DEVELOPMENT"		
"INACCESSIBLE"		

dmx.shutdownrequest

Description not provided

MODELS All

This signal does contain any arguments.

image.connector.l1displayport.edid.listchanged

Will be raised when list of EDIDs changed

MODELS All

This signal does contain any arguments.

image.connector.l1hdbaset1.edid.listchanged

Will be raised when list of EDIDs changed

MODELS All

This signal does contain any arguments.

image.connector.l1hdbaset2.edid.listchanged

Will be raised when list of EDIDs changed

MODELS All

This signal does contain any arguments.

image.connector.l1hdmi.edid.listchanged

Will be raised when list of EDIDs changed

MODELS All

This signal does contain any arguments.

image.processing.blacklevel.file.listchanged

This signal is fired every time the return value of List() changes

MODELS All

This signal does contain any arguments.

image.processing.blend.file.listchanged

This signal is fired every time the return value of List() changes

MODELS All

This signal does contain any arguments.

image.processing.warp.file.listchanged

This signal is fired every time the return value of List() changes

MODELS

All

This signal does contain any arguments.

image.processing.warpgrid.changed

Fired when the grid changes, without the grid data as payload

MODELS

All

This signal does contain any arguments.

image.processing.warpgrid.gridchanged

Description not provided

MODELS

All

Name	Type	Description																												
grid	[{ }]																													
	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>x</td><td>{ }</td></tr></table>	Name	Type	x	{ }																									
Name	Type																													
x	{ }																													
	<table><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td></tr><tr><td>y</td><td>{ }</td><td></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table></td><td></td></tr><tr><td></td><td><table><tr><th>Name</th><th>Type</th></tr></table></td><td></td></tr></table>		<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }		<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	y	{ }			<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }			<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type	
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																							
Name	Type																													
base	float																													
hints	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																											
Name	Type																													
y	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>base</td><td>float</td></tr><tr><td>hints</td><td>{ }</td></tr></table>	Name	Type	base	float	hints	{ }																							
Name	Type																													
base	float																													
hints	{ }																													
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type																											
Name	Type																													

image.testpattern.added

Description not provided

MODELS

All

Name	Type	Description														
pattern	{ }	The ID of the new pattern that was added														
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>name</td><td>string</td></tr><tr><td>location</td><td>string</td></tr><tr><td>id</td><td>string</td></tr><tr><td>above</td><td>bool</td></tr><tr><td>internal</td><td>bool</td></tr><tr><td>properties</td><td>[{ }]</td></tr></table>	Name	Type	name	string	location	string	id	string	above	bool	internal	bool	properties	[{ }]	
Name	Type															
name	string															
location	string															
id	string															
above	bool															
internal	bool															
properties	[{ }]															
	{ }															
	<table><tr><th>Name</th><th>Type</th></tr></table>	Name	Type													
Name	Type															

key	string
value	string

image.testpattern.changed

Description not provided

MODELS All

Name	Type	Description						
id	string	The unique pattern id of the pattern that have changed						
properties	[{ }]							
	{ }							
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>	Name	Type	key	string	value	string
Name	Type							
key	string							
value	string							

image.testpattern.file.listchanged

This signal is fired every time the return value of List() changes

MODELS All

This signal does contain any arguments.

image.testpattern.removed

Description not provided

MODELS All

Name	Type	Description														
pattern	{ }	The ID of the pattern that was removed														
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>name</td><td>string</td></tr><tr><td>location</td><td>string</td></tr><tr><td>id</td><td>string</td></tr><tr><td>above</td><td>bool</td></tr><tr><td>internal</td><td>bool</td></tr><tr><td>properties</td><td>[{ }]</td></tr></table>	Name	Type	name	string	location	string	id	string	above	bool	internal	bool	properties	[{ }]
Name	Type															
name	string															
location	string															
id	string															
above	bool															
internal	bool															
properties	[{ }]															
	{ }															
		<table><tr><th>Name</th><th>Type</th></tr><tr><td>key</td><td>string</td></tr><tr><td>value</td><td>string</td></tr></table>	Name	Type	key	string	value	string								
Name	Type															
key	string															
value	string															

keydispatcher.keyevent

Raised when the user has pressed a physical key on an input device.

MODELS All

Name	Type	Description
------	------	-------------

key	string	The logical key name that was pressed
type	enum	Type of event: press, repeat, release, clicked. Values "PRESS" "RELEASE" "REPEAT" "CLICKED"
source	string	The source of the event, e.g remote control or keypad

network.added

Raised when a new device has been added

MODELS All

Name	Type	Description
id	string	The logical device id of the new device, e.g: 'wifi1'

network.removed

Raised when a device has been removed

MODELS All

Name	Type	Description
id	string	The logical device id of the device that was removed, e.g: 'wired2'

notification.caution.dismissed

Signal the dismissal of a notification of the given severity level

MODELS All

This signal does contain any arguments.

notification.caution.emitted

Signal the emission of a new notification of the given severity level

MODELS All

This signal does contain any arguments.

notification.dismissed

Description not provided

MODELS All

Name	Type	Description
id	string	The ID of the dismissed notification
response	enum	The reason or response for dismissing the notification Values "NONE"

"OK "

"CANCEL "

"YES "

"NO "

"SUPPRESS "

notification.emitted

Description not provided

MODELS All

Name	Type	Description							
notification	{ }	The new notification							
	<table><tr><th>Name</th><th>Type</th></tr><tr><td>severity</td><td>enum</td></tr></table>	Name	Type	severity	enum				
Name	Type								
severity	enum								
		<table><tr><th>Values</th></tr><tr><td>"INFO"</td></tr><tr><td>"CAUTION"</td></tr><tr><td>"WARNING"</td></tr><tr><td>"ERROR"</td></tr></table>	Values	"INFO"	"CAUTION"	"WARNING"	"ERROR"		
Values									
"INFO"									
"CAUTION"									
"WARNING"									
"ERROR"									
	id	string							
	code	string							
	timestamp	string							
	arguments	[string]							
	message	string							
	timeout	string							
	visibility	string							
	doc	string							
	actions	[enum]							
		<table><tr><th>Values</th></tr><tr><td>"NONE"</td></tr><tr><td>"OK"</td></tr><tr><td>"CANCEL"</td></tr><tr><td>"YES"</td></tr><tr><td>"NO"</td></tr><tr><td>"SUPPRESS"</td></tr></table>	Values	"NONE"	"OK"	"CANCEL"	"YES"	"NO"	"SUPPRESS"
Values									
"NONE"									
"OK"									
"CANCEL"									
"YES"									
"NO"									
"SUPPRESS"									

notification.error.dismissed

Signal the dismissal of a notification of the given severity level

MODELS All

This signal does contain any arguments.

notification.error.emitted

Signal the emission of a new notification of the given severity level

MODELS

 All

This signal does contain any arguments.

notification.info.dismissed

Signal the dismissal of a notification of the given severity level

MODELS

 All

This signal does contain any arguments.

notification.info.emitted

Signal the emission of a new notification of the given severity level

MODELS

 All

This signal does contain any arguments.

notification.warning.dismissed

Signal the dismissal of a notification of the given severity level

MODELS

 All

This signal does contain any arguments.

notification.warning.emitted

Signal the emission of a new notification of the given severity level

MODELS

 All

This signal does contain any arguments.

profile.profileactivated

Fired when a profile is activated

MODELS

 All

Name	Type	Description
name	string	The name of the activated profile

system.identificationchanged

Will be raised whenever an identification is changed

MODELS

 All

Name	Type	Description
identification	string	

system.license.licensechanged

Description not provided

MODELS

All

This signal does contain any arguments.

system.performed

Emitted when one or more domains have completed resetting. This signal might be emitted several times, every time with one or more domains, until all requested domains are done resetting. 'All domains' in this context is either the list of domains supplied to a call to 'Reset' or the list of domains returned from 'ResetAll'.

MODELS

All

Name	Type	Description
domains	[enum]	
		Values "Dmx" "Environment" "Gpio" "Gsm" "Illumination" "ImageBlend" "ImageCapture" "ImageConnector" "ImageConvergence" "ImageDisplay" "ImageFeatures" "ImageOrientation" "ImageRealColor" "ImageResolution" "ImageSource" "ImageStereo" "ImageTestPattern" "ImageUniformity" "ImageWarp" "LightMeasurement" "LightSteering" "LinkedServices" "Macro" "MediaPlayer" "Network" "Optics" "Peripheral" "Profiles" "Scheduler"

"Screen"
"Snmp"
"System"
"Time"
"User"
"UserInterface"

ui.settings.added

Fired when a new key/value pair was added

MODELS All

Name	Type	Description
key	string	The key name
value	string	The key value

ui.settings.changed

Fired when a key has an updated value

MODELS All

Name	Type	Description
key	string	The key name
value	string	The value of the key

ui.settings.removed

Fired when a key was removed

MODELS All

Name	Type	Description
key	string	The key name

Files

Alphabetical list of all file end points.

bio.logger.transfer

Download notification log files

MODELS All	
Upload	Download
No	Yes

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/bio/logger/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/bio/logger/transfer -Method Get -
OutFile logger.dat
```

image.connector.edid.transfer

Upload and download EDID files

MODELS All	
Upload	Download
Yes	Yes

Example of file upload using the `curl` program.

 

```
pulse:~$ curl -F file=@edid.dat http://192.168.1.100/api/image/connector/edid/transfer
```

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/connector/edid/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/connector/edid/transfer -
Method Get -OutFile edid.dat
```

image.processing.blacklevel.file.transfer

Upload and download black level correction file

MODELS

All

Upload

Download

Yes

Yes

Example of file upload using the `curl` program.

 

```
pulse:~$ curl -F file=@blacklevel.dat http://192.168.1.100/api/image/processing/blacklevel/file/transfer
```

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/processing/blacklevel/file/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/processing/blacklevel/file/transfer -Method Get -OutFile blacklevel.dat
```

image.processing.blend.file.transfer

Upload and download blend files

MODELS

All

Upload

Download

Yes

Yes

Example of file upload using the `curl` program.

 

```
pulse:~$ curl -F file=@blend.dat http://192.168.1.100/api/image/processing/blend/file/transfer
```

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/processing/blend/file/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/processing/blend/file/transfer -Method Get -OutFile blend.dat
```

image.processing.warp.file.transfer

Upload and download warp files

MODELS All	
Upload	Download
Yes	Yes

Example of file upload using the `curl` program.

 

```
pulse:~$ curl -F file=@warp.dat http://192.168.1.100/api/image/processing/warp/file/transfer
```

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/processing/warp/file/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/processing/warp/file/transfer -Method Get -OutFile warp.dat
```

image.source.preview.transfer

Download current source image

MODELS All	
Upload	Download
No	Yes

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/source/preview/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/source/preview/transfer -Method Get -OutFile preview.dat
```

image.source.transfer

Download current source image

MODELS All	
------------	--

Upload

Download

No

Yes

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/source/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/source/transfer -Method Get -
OutFile source.dat
```

image.testpattern.file.transfer

Upload and download test pattern images

MODELS

All

Upload

Download

Yes

Yes

Example of file upload using the `curl` program.

 

```
pulse:~$ curl -F file=@testpattern.dat http://192.168.1.100/api/image/testpattern/file/transfer
```

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/image/testpattern/file/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/image/testpattern/file/transfer -
Method Get -OutFile testpattern.dat
```

notification.logger.transfer

Download notification log files

MODELS

All

Upload

Download

No

Yes

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl`

use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/notification/logger/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/notification/logger/transfer -
Method Get -OutFile logger.dat
```

peripheral.camera.transfer

Download camera image file

MODELS

All

Upload

Download

No

Yes

Example of file download using the **curl** program on **Linux** and **Mac OS X**. Using the **-O** and **-J** option makes **curl** use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/peripheral/camera/transfer
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/peripheral/camera/transfer -Method
Get -OutFile camera.dat
```

system.license.file.registration

Download registration file

MODELS

All

Upload

Download

No

Yes

Example of file download using the **curl** program on **Linux** and **Mac OS X**. Using the **-O** and **-J** option makes **curl** use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/system/license/file/registration
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/system/license/file/registration -
Method Get -OutFile registration.dat
```

ui.capture.lcd

Download screen capture of LCD

MODELS

All

Upload

Download

No

Yes

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/ui/capture/lcd
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/ui/capture/lcd -Method Get -OutFile lcd.dat
```

ui.capture.osd

Download screen capture of OSD

MODELS

All

Upload

Download

No

Yes

Example of file download using the `curl` program on **Linux** and **Mac OS X**. Using the `-O` and `-J` option makes `curl` use the file name suggestion of the server and will overwrite a file if it already exists.

 

```
pulse:~$ curl -O -J http://192.168.1.100/api/ui/capture/osd
```

Example of file download using the **PowerShell** on **Windows**.



```
PS C:\Users\pulse> Invoke-WebRequest -Uri http://192.168.1.100/api/ui/capture/osd -Method Get -OutFile osd.dat
```

ui.splashscreen.file

Upload splash screen for LCD - 800x480 PNG,JPG

MODELS

All

Upload

Download

Yes

No

Example of file upload using the `curl` program.



```
pulse:~$ curl -F file=@splashscreen.dat http://192.168.1.100/api/ui/splashscreen/file
```

Property index

dmx.artnet	44	dmx.artnetnet	44
dmx.artnetuniverse	44	dmx.mode	44
dmx.monitor.channel01.function	45	dmx.monitor.channel01.offset	45
dmx.monitor.channel01.value	45	dmx.monitor.channel02.function	45
dmx.monitor.channel02.offset	46	dmx.monitor.channel02.value	46
dmx.monitor.connectionstate.active	46	dmx.shutdown	46
dmx.shutdowntimeout	47	dmx.startchannel	47
environment.alarmstate	47	firmware.component.color-sensor-boot.actualversion	47
firmware.component.color-sensor-boot.displayname	48	firmware.component.color-sensor-boot.error	48
	48	firmware.component.color-sensor-boot.progress	48
firmware.component.color-sensor-boot.mismatch	49	firmware.component.color-sensor-boot.status	49
firmware.component.color-sensor-boot.requiredversion	49	firmware.component.color-sensor-run.displayname	49
firmware.component.color-sensor-run.actualversion	49	firmware.component.color-sensor-run.mismatch	50
	50	firmware.component.color-sensor-run.requiredversion	50
firmware.component.color-sensor-run.error	50	firmware.component.cornet-fpga.actualversion	51
firmware.component.color-sensor-run.progress	51	firmware.component.cornet-fpga.error	51
firmware.component.color-sensor-run.status	51	firmware.component.cornet-fpga.progress	51
firmware.component.cornet-fpga.displayname	52	firmware.component.cornet-fpga.status	52
firmware.component.cornet-fpga.mismatch	52	firmware.component.formatter-blue-wqxga.displayname	52
firmware.component.cornet-fpga.requiredversion	52		53
firmware.component.formatter-blue-wqxga.actualversion	53	firmware.component.formatter-blue-wqxga.mismatch	53
firmware.component.formatter-blue-wqxga.error	53	firmware.component.formatter-blue-wqxga.requiredversion	54
firmware.component.formatter-blue-wqxga.progress	54	firmware.component.formatter-green-wqxga.actualversion	54
	54		54
firmware.component.formatter-blue-wqxga.status	55	firmware.component.formatter-green-wqxga.error	55
firmware.component.formatter-green-wqxga.displayname	55	firmware.component.formatter-green-wqxga.progress	55
	56	firmware.component.formatter-green-wqxga.status	56
firmware.component.formatter-green-wqxga.mismatch	56	firmware.component.formatter-red-wqxga.displayname	56
firmware.component.formatter-green-wqxga.requiredversion	56	firmware.component.formatter-red-wqxga.mismatch	57
	57	firmware.component.formatter-red-wqxga.requiredversion	57
firmware.component.formatter-red-wqxga.actualversion	57		57
firmware.component.formatter-red-wqxga.error	58	firmware.component.intel-i210.actualversion	58
firmware.component.formatter-red-wqxga.progress	58	firmware.component.intel-i210.error	58
	59	firmware.component.intel-i210.progress	59
firmware.component.formatter-red-wqxga.status	59	firmware.component.intel-i210.status	59
firmware.component.intel-i210.displayname	59	firmware.component.l1-vh-input-fpga.displayname	60
firmware.component.intel-i210.mismatch	59	firmware.component.l1-vh-input-fpga.mismatch	60
firmware.component.intel-i210.requiredversion	60	firmware.component.l1-vh-input-fpga.requiredversion	61
firmware.component.l1-vh-input-fpga.actualversion	60	firmware.component.l1-vh-input-hd-base-t-a.actualversion	61
	61		61
firmware.component.l1-vh-input-fpga.error	61	firmware.component.l1-vh-input-hd-base-t-a.error	62
firmware.component.l1-vh-input-fpga.progress	62	firmware.component.l1-vh-input-hd-base-t-a.progress	62
firmware.component.l1-vh-input-fpga.status	62	firmware.component.l1-vh-input-hd-base-t-a.status	62

Pulse API		Reference Guide	Version 2.2
firmware.component.l1-vh-input-hd-base-t-a.displayname	63	firmware.component.l1-vh-input-hd-base-t-b.displayname	63
firmware.component.l1-vh-input-hd-base-t-a.mismatch	63	firmware.component.l1-vh-input-hd-base-t-b.mismatch	64
firmware.component.l1-vh-input-hd-base-t-a.requiredversion	64	firmware.component.l1-vh-input-hd-base-t-b.requiredversion	65
firmware.component.l1-vh-input-hd-base-t-b.actualversion	65	firmware.component.l1-vh-input-power-sequencer.actualversion	65
firmware.component.l1-vh-input-hd-base-t-b.error	65	firmware.component.l1-vh-input-power-sequencer.error	66
firmware.component.l1-vh-input-hd-base-t-b.progress	66	firmware.component.l1-vh-input-power-sequencer.progress	66
firmware.component.l1-vh-input-hd-base-t-b.status	67	firmware.component.l1-vh-input-power-sequencer.status	67
firmware.component.l1-vh-input-power-sequencer.displayname	67	firmware.component.laser-control-board-boot.displayname	68
firmware.component.l1-vh-input-power-sequencer.mismatch	68	firmware.component.laser-control-board-boot.mismatch	69
firmware.component.l1-vh-input-power-sequencer.requiredversion	68	firmware.component.laser-control-board-boot.requiredversion	69
firmware.component.laser-control-board-boot.actualversion	69	firmware.component.laser-control-board-fpga.actualversion	70
firmware.component.laser-control-board-boot.error	70	firmware.component.laser-control-board-fpga.error	70
firmware.component.laser-control-board-boot.progress	71	firmware.component.laser-control-board-fpga.progress	71
firmware.component.laser-control-board-boot.status	71	firmware.component.laser-control-board-fpga.status	72
firmware.component.laser-control-board-fpga.displayname	72	firmware.component.laser-control-board-run.displayname	72
firmware.component.laser-control-board-fpga.mismatch	73	firmware.component.laser-control-board-run.mismatch	73
firmware.component.laser-control-board-fpga.requiredversion	73	firmware.component.laser-control-board-run.requiredversion	74
firmware.component.laser-control-board-run.actualversion	74	firmware.component.laser-driver-module-1-hdr.actualversion	74
firmware.component.laser-control-board-run.error	74	firmware.component.laser-driver-module-1-hdr.error	75
firmware.component.laser-control-board-run.progress	75	firmware.component.laser-driver-module-1-hdr.progress	76
firmware.component.laser-control-board-run.status	76	firmware.component.laser-driver-module-1-primary.displayname	76
firmware.component.laser-driver-module-1-hdr.displayname	76	firmware.component.laser-driver-module-1-primary.mismatch	77
firmware.component.laser-driver-module-1-hdr.mismatch	77	firmware.component.laser-driver-module-1-primary.requiredversion	78
firmware.component.laser-driver-module-1-hdr.requiredversion	77	firmware.component.laser-driver-module-1-secondary.actualversion	78
firmware.component.laser-driver-module-1-primary.actualversion	78	firmware.component.laser-driver-module-1-secondary.error	79
firmware.component.laser-driver-module-1-primary.error	79	firmware.component.laser-driver-module-1-secondary.progress	79
firmware.component.laser-driver-module-1-primary.progress	80	firmware.component.laser-driver-module-1-secondary.status	80
firmware.component.laser-driver-module-1-primary.status	80	firmware.component.laser-driver-module-1-slot-1.displayname	81
firmware.component.laser-driver-module-1-secondary.displayname	81	firmware.component.laser-driver-module-1-slot-1.mismatch	82
firmware.component.laser-driver-module-1-	82		

secondary.mismatch	82	firmware.component.laser-driver-module-1-slot-1.requiredversion	82
firmware.component.laser-driver-module-1-secondary.requiredversion	82		83
	83	firmware.component.laser-driver-module-1-slot-2.actualversion	83
firmware.component.laser-driver-module-1-slot-1.actualversion	83	firmware.component.laser-driver-module-1-slot-2.error	83
firmware.component.laser-driver-module-1-slot-1.error	84		84
	84	firmware.component.laser-driver-module-1-slot-2.progress	84
firmware.component.laser-driver-module-1-slot-1.progress	85	firmware.component.laser-driver-module-1-slot-2.status	85
firmware.component.laser-driver-module-1-slot-1.status	85	firmware.component.laser-driver-module-1-slot-3.displayname	86
	85		
firmware.component.laser-driver-module-1-slot-2.displayname	86	firmware.component.laser-driver-module-1-slot-3.mismatch	86
	86		87
firmware.component.laser-driver-module-1-slot-2.mismatch	87	firmware.component.laser-driver-module-1-slot-3.requiredversion	87
firmware.component.laser-driver-module-1-slot-2.requiredversion	87	firmware.component.laser-driver-module-1-slot-6.actualversion	87
	88		88
firmware.component.laser-driver-module-1-slot-3.actualversion	88	firmware.component.laser-driver-module-1-slot-6.error	88
firmware.component.laser-driver-module-1-slot-3.error	89		89
	89	firmware.component.laser-driver-module-1-slot-6.progress	89
firmware.component.laser-driver-module-1-slot-3.progress	89	firmware.component.laser-driver-module-1-slot-6.status	90
firmware.component.laser-driver-module-1-slot-3.status	90		90
	90	firmware.component.laser-driver-module-1-slot-7.displayname	91
firmware.component.laser-driver-module-1-slot-6.displayname	91	firmware.component.laser-driver-module-1-slot-7.mismatch	91
firmware.component.laser-driver-module-1-slot-6.mismatch	91	firmware.component.laser-driver-module-1-slot-7.requiredversion	91
	92		92
firmware.component.laser-driver-module-1-slot-6.requiredversion	92	firmware.component.laser-power-module-1-slot-8.actualversion	92
	92		93
firmware.component.laser-driver-module-1-slot-7.actualversion	93	firmware.component.laser-power-module-1-slot-8.error	93
firmware.component.laser-driver-module-1-slot-7.error	93	firmware.component.laser-power-module-1-slot-8.progress	94
	94		94
firmware.component.laser-driver-module-1-slot-7.progress	94	firmware.component.laser-power-module-1-slot-8.status	94
firmware.component.laser-driver-module-1-slot-7.status	95	firmware.component.mainboard-power-sequencer.displayname	95
	95		95
firmware.component.laser-power-module-1-slot-8.displayname	96	firmware.component.mainboard-power-sequencer.mismatch	96
firmware.component.laser-power-module-1-slot-8.mismatch	96	firmware.component.mainboard-power-sequencer.requiredversion	96
	96		97
firmware.component.laser-power-module-1-slot-8.requiredversion	97	firmware.component.peltier-control-board-boot.actualversion	97
firmware.component.mainboard-power-sequencer.actualversion	97	firmware.component.peltier-control-board-boot.error	97
	98		98
firmware.component.mainboard-power-sequencer.error	98	firmware.component.peltier-control-board-boot.progress	98
	98		98
firmware.component.mainboard-power-sequencer.progress	99	firmware.component.peltier-control-board-boot.status	99
	99		
firmware.component.mainboard-power-sequencer.status	99	firmware.component.peltier-control-board-run.displayname	99
	99		100
firmware.component.peltier-control-board-boot.displayname	100	firmware.component.peltier-control-board-run.mismatch	100
	100		101
firmware.component.peltier-control-board-boot.mismatch	101	firmware.component.peltier-control-board-run.requiredversion	101
	101		101

Pulse API		Reference Guide	Version 2.2
firmware.component.peltier-control-board-boot.requiredversion	101	firmware.component.peri-fpga.actualversion	102
	102	firmware.component.peri-fpga.error	102
firmware.component.peltier-control-board-run.actualversion	102	firmware.component.peri-fpga.progress	102
firmware.component.peltier-control-board-run.error	103	firmware.component.peri-fpga.status	103
	103	firmware.component.power-distribution-board-boot-2.displayname	103
firmware.component.peltier-control-board-run.progress	104		104
	104	firmware.component.power-distribution-board-boot-2.mismatch	104
firmware.component.peltier-control-board-run.status	104		105
firmware.component.peri-fpga.displayname	104	firmware.component.power-distribution-board-boot-2.requiredversion	105
	105		105
firmware.component.peri-fpga.mismatch	105	firmware.component.power-distribution-board-fcb-fpga.actualversion	105
firmware.component.peri-fpga.requiredversion	106		106
firmware.component.power-distribution-board-boot-2.actualversion	106	firmware.component.power-distribution-board-fcb-fpga.error	106
	107		107
firmware.component.power-distribution-board-boot-2.error	107	firmware.component.power-distribution-board-fcb-fpga.progress	107
	107	firmware.component.power-distribution-board-fcb-fpga.status	107
firmware.component.power-distribution-board-boot-2.progress	107		108
	108	firmware.component.power-distribution-board-lsr-fantable-ac.displayname	108
firmware.component.power-distribution-board-boot-2.status	108		109
	109	firmware.component.power-distribution-board-lsr-fantable-ac.mismatch	109
firmware.component.power-distribution-board-fcb-fpga.displayname	109		110
	109	firmware.component.power-distribution-board-lsr-fantable-ac.requiredversion	110
firmware.component.power-distribution-board-fcb-fpga.mismatch	110		110
	110	firmware.component.power-distribution-board-lsr-fantable-on.actualversion	110
firmware.component.power-distribution-board-fcb-fpga.requiredversion	110		111
firmware.component.power-distribution-board-lsr-fantable-ac.actualversion	111	firmware.component.power-distribution-board-lsr-fantable-on.error	111
	111		112
firmware.component.power-distribution-board-lsr-fantable-ac.error	112	firmware.component.power-distribution-board-lsr-fantable-on.progress	112
	112		113
firmware.component.power-distribution-board-lsr-fantable-ac.progress	113	firmware.component.power-distribution-board-lsr-fantable-on.status	113
	114		114
firmware.component.power-distribution-board-lsr-fantable-ac.status	114	firmware.component.power-distribution-board-lsr-fantable-sb.displayname	114
	114		114
firmware.component.power-distribution-board-lsr-fantable-on.displayname	115	firmware.component.power-distribution-board-lsr-fantable-sb.mismatch	115
	115		116
firmware.component.power-distribution-board-lsr-fantable-on.mismatch	116	firmware.component.power-distribution-board-lsr-fantable-sb.requiredversion	116
	116		117
firmware.component.power-distribution-board-lsr-fantable-on.requiredversion	116	firmware.component.power-distribution-board-lsr-fantable-tc.actualversion	117
	117		117
firmware.component.power-distribution-board-lsr-fantable-sb.actualversion	117	firmware.component.power-distribution-board-lsr-fantable-tc.error	117
	118		118
firmware.component.power-distribution-board-lsr-fantable-sb.error	118	firmware.component.power-distribution-board-lsr-fantable-tc.progress	118
	119		119
firmware.component.power-distribution-board-lsr-fantable-sb.progress	119	firmware.component.power-distribution-board-lsr-fantable-tc.status	119
	120		119
firmware.component.power-distribution-board-lsr-fantable-sb.status	120	firmware.component.power-distribution-board-mcb-fpga.displayname	120
	120		120
firmware.component.power-distribution-board-lsr-fantable-tc.displayname	121	firmware.component.power-distribution-board-mcb-fpga.mismatch	121
	121		121
firmware.component.power-distribution-board-lsr-fantable-tc.mismatch	121	firmware.component.power-distribution-board-mcb-fpga.requiredversion	122
	122		122
firmware.component.power-distribution-board-lsr-fantable-tc.requiredversion	122	firmware.component.power-distribution-board-run-2.actualversion	122
	122		123
firmware.component.power-distribution-board-	123	firmware.component.power-distribution-board-run-2.error	123

Pulse API		Reference Guide	Version 2.2
mcb-fpga.actualversion	123	firmware.component.power-distribution-board-run-2.progress	124
firmware.component.power-distribution-board-mcb-fpga.error	124	firmware.component.power-distribution-board-run-2.status	125
firmware.component.power-distribution-board-mcb-fpga.progress	125	firmware.component.switch-mode-power-supply.actualversion	126
firmware.component.power-distribution-board-mcb-fpga.status	126	firmware.component.switch-mode-power-supply.error	127
firmware.component.power-distribution-board-run-2.displayname	127	firmware.component.switch-mode-power-supply.progress	127
firmware.component.power-distribution-board-run-2.mismatch	127	firmware.component.switch-mode-power-supply.status	128
firmware.component.power-distribution-board-run-2.requiredversion	128	firmware.component.u-boot-img.displayname	130
firmware.component.status	129	firmware.component.u-boot-img.mismatch	130
firmware.component.switch-mode-power-supply.displayname	130	firmware.component.u-boot-img.requiredversion	131
firmware.component.switch-mode-power-supply.mismatch	131	firmware.component.u-boot-spl.actualversion	131
firmware.component.switch-mode-power-supply.requiredversion	132	firmware.component.u-boot-spl.error	132
firmware.component.u-boot-img.actualversion	132	firmware.component.u-boot-spl.progress	132
firmware.component.u-boot-img.error	133	firmware.component.u-boot-spl.status	133
firmware.component.u-boot-img.progress	134	firmware.component.wobulator-boot.displayname	134
firmware.component.u-boot-img.status	134	firmware.component.wobulator-boot.mismatch	135
firmware.component.u-boot-spl.displayname	135	firmware.component.wobulator-boot.requiredversion	135
firmware.component.u-boot-spl.mismatch	136	firmware.component.wobulator-run.actualversion	136
firmware.component.u-boot-spl.requiredversion	136	firmware.component.wobulator-run.error	138
firmware.component.wobulator-boot.actualversion	138	firmware.component.wobulator-run.progress	140
firmware.component.wobulator-boot.error	140	firmware.component.wobulator-run.status	141
firmware.component.wobulator-boot.progress	141	firmware.eula.requiretermagreement	141
firmware.component.wobulator-boot.status	142	firmware.firmwareversion	142
firmware.component.wobulator-run.displayname	146	firmware.status	146
firmware.component.wobulator-run.mismatch	146	gsm.iccid	146
firmware.component.wobulator-run.requiredversion	147	gsm.pinstate	147
firmware.error	147	illumination.clo.allowed	148
firmware.eula.termagreement	152	illumination.clo.enable	152
firmware.progress	152	illumination.clo.setpoint	152
gsm.data.status	153	illumination.laser.opticalalignment	153
gsm.imei	157	illumination.laser.timcuring.state	153
gsm.signalstrength	158	illumination.laser.whitepoint.tracking.progress	157
illumination.clo.availability	158	illumination.laser.whitepoint.tracking.state	158
illumination.clo.scale	159	illumination.sources.laser.controlledbyclo	159
illumination.clo.state	163	illumination.sources.laser.maxpower	159
illumination.laser.timcuring.cureavailable	163	illumination.sources.laser.power	163
illumination.laser.whitepoint.tracking.cancellable	164	illumination.sources.laser.status	164
illumination.laser.whitepoint.tracking.result	164	image.blackcontentdetection.dimminginterval	164
illumination.sources.laser.actualpower	165	image.blackcontentdetection.sampleinterval	165
	169	image.blackcontentdetection.threshold	169
	169	image.capture.corrected	170
	170	image.color.p7.custom.bluegain	170

Pulse API		Reference Guide	Version 2.2
illumination.sources.laser.ispowerlimited	170	image.color.p7.custom.bluex	174
illumination.sources.laser.minpower	175	image.color.p7.custom.cmyreadable	175
illumination.sources.laser.powerlimitreason	175	image.color.p7.custom.cyangain	175
illumination.state	176	image.color.p7.custom.cyanx	176
image.blackcontentdetection.enable	180	image.color.p7.custom.gainsavailable	180
image.blackcontentdetection.state	180	image.color.p7.custom.greenlum	181
image.brightness	181	image.color.p7.custom.greeny	181
image.capture.enable	181	image.color.p7.custom.magentagain	185
image.color.p7.custom.bluelum	186	image.color.p7.custom.magentax	186
image.color.p7.custom.bluey	186	image.color.p7.custom.mode	187
image.color.p7.custom.cmywritable	187	image.color.p7.custom.redgain	188
image.color.p7.custom.cyanlum	188	image.color.p7.custom.redx	188
image.color.p7.custom.cyany	189	image.color.p7.custom.rgbcmYGainsAvailable	189
image.color.p7.custom.greengain	189	image.color.p7.custom.target	189
image.color.p7.custom.greenx	190	image.color.p7.custom.whitegainavailable	190
image.color.p7.custom.luminancesavailable	190	image.color.p7.custom.whitemode	191
image.color.p7.custom.magentalum	191	image.color.p7.custom.whitetemperatureavailable	191
image.color.p7.custom.magentay	191	image.color.p7.custom.whitex	192
image.color.p7.custom.modes	192	image.color.p7.custom.yellowgain	192
image.color.p7.custom.redlum	193	image.color.p7.custom.yellowx	193
image.color.p7.custom.redy	193	image.color.p7.native.c1	193
image.color.p7.custom.rgbwritable	193	image.color.p7.native.c2	194
image.color.p7.custom.whitegain	194	image.color.p7.native.list	194
image.color.p7.custom.whitelum	195	image.color.p7.native.normal.c1available	195
image.color.p7.custom.whitetemperature	196	image.color.p7.native.normal.c2available	196
image.color.p7.custom.whitewritable	196	image.color.p7.native.normal.whiteavailable	196
image.color.p7.custom.whitey	197	image.color.p7.native.whiteavailable	197
image.color.p7.custom.yellowlum	197	image.connector.l1displayport.capabilities	198
image.color.p7.custom.yellowy	198	image.connector.l1displayport.colorprimariesavailable	198
image.color.p7.native.c1available	199	image.connector.l1displayport.detectedsignal	199
image.color.p7.native.c2available	200	image.connector.l1displayport.name	200
image.color.p7.native.normal.c1	200	image.connector.l1hdbaset1.capabilities	201
image.color.p7.native.normal.c2	201	image.connector.l1hdbaset1.colorprimariesavailable	202
image.color.p7.native.normal.rgbw	202	image.connector.l1hdbaset1.detectedsignal	202
image.color.p7.native.rgbw	202	image.connector.l1hdbaset1.name	203
image.color.rgbmode.rgbmode	203	image.connector.l1hdbaset2.capabilities	203
image.connector.l1displayport.colorprimaries	204	image.connector.l1hdbaset2.colorprimariesavailable	204
image.connector.l1displayport.colorspace	204	image.connector.l1hdbaset2.detectedsignal	205
image.connector.l1displayport.edid.selected	205	image.connector.l1hdbaset2.name	205
image.connector.l1displayport.signalrange	206	image.connector.l1hdmi.capabilities	206
image.connector.l1hdbaset1.colorprimaries	206	image.connector.l1hdmi.colorprimariesavailable	206
image.connector.l1hdbaset1.colorspace	207	image.connector.l1hdmi.detectedsignal	207
image.connector.l1hdbaset1.edid.selected	207	image.connector.l1hdmi.name	208
image.connector.l1hdbaset1.signalrange	208	image.connector.l1sdi.capabilities	209

Pulse API		Reference Guide	Version 2.2
image.connector.l1hdbaset2.colorprimaries	209	image.connector.l1sdia.colorprimariesavailable	209
image.connector.l1hdbaset2.colorspace	210	image.connector.l1sdia.detectedsignal	210
image.connector.l1hdbaset2.edid.selected	211	image.connector.l1sdia.signalrange	211
image.connector.l1hdbaset2.signalrange	211	image.connector.l1sdib.colorprimaries	211
image.connector.l1hdmi.colorprimaries	212	image.connector.l1sdib.colorspace	212
image.connector.l1hdmi.colorspace	212	image.connector.l1sdib.name	213
image.connector.l1hdmi.edid.selected	213	image.connector.l1sdic.capabilities	214
image.connector.l1hdmi.signalrange	214	image.connector.l1sdic.colorprimariesavailable	215
image.connector.l1sdia.colorprimaries	215	image.connector.l1sdic.detectedsignal	215
image.connector.l1sdia.colorspace	216	image.connector.l1sdic.signalrange	216
image.connector.l1sdia.name	216	image.connector.l1sdid.colorprimaries	217
image.connector.l1sdib.capabilities	217	image.connector.l1sdid.colorspace	217
image.connector.l1sdib.colorprimariesavailable	218	image.connector.l1sdid.name	218
image.connector.l1sdib.detectedsignal	218	image.contrast	219
image.connector.l1sdib.signalrange	219	image.convergence.green	220
image.connector.l1sdic.colorprimaries	220	image.display.desireddisplaymode	220
image.connector.l1sdic.colorspace	221	image.display.frequency	221
image.connector.l1sdic.name	222	image.dynamiccontrast.level	222
image.connector.l1sdid.capabilities	222	image.gammatype	222
image.connector.l1sdid.colorprimariesavailable	223	image.orientation	223
image.connector.l1sdid.detectedsignal	223	image.processing.blacklevel.basicblacklevel.bottom	223
image.connector.l1sdid.signalrange	223	image.processing.blacklevel.basicblacklevel.left	224
image.convergence.blue	224	image.processing.blacklevel.basicblacklevel.right	224
image.convergence.red	224	image.processing.blacklevel.bluegain	225
image.display.displaymode	225	image.processing.blacklevel.file.selected	225
image.display.synchronouslock	225	image.processing.blacklevel.linkrgb	226
image.gamma	226	image.processing.blend.basicblend.bottom	226
image.intensity	226	image.processing.blend.basicblend.left	227
image.orientationmode	227	image.processing.blend.basicblend.top	227
image.processing.blacklevel.basicblacklevel.enable	228	image.processing.blend.file.maxselected	228
image.processing.blacklevel.basicblacklevel.level	228	image.processing.blend.scurve	228
image.processing.blacklevel.basicblacklevel.top	229	image.processing.transportdelay.desired	229
image.processing.blacklevel.file.enable	229	image.processing.warp.bow.bottomleftu	229
image.processing.blacklevel.greengain	230	image.processing.warp.bow.bottomrightu	230
image.processing.blacklevel.redgain	230	image.processing.warp.bow.enable	230
image.processing.blend.basicblend.enable	230	image.processing.warp.bow.topleftu	231
image.processing.blend.basicblend.right	231	image.processing.warp.bow.toprightu	231
image.processing.blend.file.enable	231	image.processing.warp.enable	232
image.processing.blend.file.selected	232	image.processing.warp.file.selected	232
image.processing.transportdelay.actual	232	image.processing.warp.fourcorners.bottomright	232
image.processing.transportdelay.minimum	233	image.processing.warp.fourcorners.screenheight	233
image.processing.warp.bow.bottomleftv	233	image.processing.warp.fourcorners.topleft	234
image.processing.warp.bow.bottomrightv	234	image.processing.warp.position.enable	234
image.processing.warp.bow.symmetric	235	image.processing.warp.position.vertical	235

Pulse API		Reference Guide	Version 2.2
image.processing.warp.bow.topleftv	235	image.processing.warp.warpstatusdescription	235
image.processing.warp.bow.toprightv	236	image.processing.warp.zoom.factor	236
image.processing.warp.file.enable	236	image.resolution.alpha.size	236
image.processing.warp.fourcorners.bottomleft	237	image.resolution.beta.size	237
image.processing.warp.fourcorners.enable	237	image.resolution.osd.size	238
image.processing.warp.fourcorners.screenwidth	238	image.resolution.processing.size	238
image.processing.warp.fourcorners.topright	238	image.resolution.size	239
image.processing.warp.position.horizontal	239	image.sharpness	239
image.processing.warp.warpstatus	239	image.source.l1displayport.capabilities	240
image.processing.warp.zoom.enable	240	image.source.l1displayport.name	240
image.processing.warp.zoom.size	241	image.source.l1hdbaset1.layout	241
image.resolution.availableresolutions	241	image.source.l1hdbaset2.capabilities	242
image.resolution.display.size	242	image.source.l1hdbaset2.name	242
image.resolution.output.size	242	image.source.l1hdmi.layout	243
image.resolution.resolution	243	image.source.l1quadsdi.capabilities	243
image.saturation	244	image.source.l1quadsdi.name	244
image.source.enable	244	image.source.l1sdia.layout	244
image.source.l1displayport.layout	245	image.source.l1sdib.capabilities	245
image.source.l1hdbaset1.capabilities	245	image.source.l1sdib.name	245
image.source.l1hdbaset1.name	246	image.source.l1sdic.layout	246
image.source.l1hdbaset2.layout	246	image.source.l1sdid.capabilities	246
image.source.l1hdmi.capabilities	246	image.source.l1sdid.name	247
image.source.l1hdmi.name	247	image.stereo.darktime	247
image.source.l1quadsdi.layout	248	image.stereo.glassync.delaymaximum	248
image.source.l1sdia.capabilities	248	image.stereo.glassync.invert	248
image.source.l1sdia.name	248	image.testpattern.available	249
image.source.l1sdib.layout	249	image.testpattern.iterator.calibration.size	249
image.source.l1sdic.capabilities	250	image.testpattern.iterator.iris.size	250
image.source.l1sdic.name	250	image.testpattern.iterator.optics.size	250
image.source.l1sdid.layout	251	image.testpattern.iterator.shift.size	251
image.source.preview.enable	251	image.testpattern.iterator.stereo.size	251
image.stereo.glassync.delay	251	image.testpattern.show	252
image.stereo.glassync.delayminimum	252	image.window.main.position	252
image.stereo.swapframepair	252	image.window.main.size	253
image.testpattern.iterator.calibration.list	253	image.window.main.sourceresolutionlimit	253
image.testpattern.iterator.iris.list	253	iot.certificate.applicable	254
image.testpattern.iterator.optics.list	254	iot.certificate.valid	254
image.testpattern.iterator.shift.list	255	iot.cloudservices.enable	255
image.testpattern.iterator.stereo.list	255	iot.proxy.enabled	255
image.testpattern.selected	255	iot.proxy.port	256
image.window.main.maximumsourceresolution	256	iot.requiretermagreement	256
image.window.main.scalingmode	256	keydispatcher.repeatdelay	257
image.window.main.source	257	network.device.hdbaset.carrier	257
image.window.main.sourcevalidity	257	network.device.hdbaset.devicetype	258

Pulse API		Reference Guide	Version 2.2
iot.certificate.available	258	network.device.hdbaset.ip4config	258
iot.cloudservices.connectionstate	258	network.device.hdbaset.ip6config	259
iot.proxy.address	259	network.device.hdbaset.speed	259
iot.proxy.password	259	network.device.hdbaset.stateinfo	260
iot.proxy.username	260	network.device.lan.configuration	260
iot.termagreement	260	network.device.lan.hwaddress	261
keydispatcher.repeatinterval	261	network.device.lan.ip4configmanual	261
network.device.hdbaset.configuration	262	network.device.lan.ip6configmanual	262
network.device.hdbaset.hwaddress	262	network.device.lan.state	262
network.device.hdbaset.ip4configmanual	263	network.device.mobile.apn	263
network.device.hdbaset.ip6configmanual	263	network.device.mobile.devicetype	263
network.device.hdbaset.state	263	network.device.mobile.ip4config	264
network.device.lan.carrier	264	network.device.mobile.ip6config	264
network.device.lan.devicetype	265	network.device.mobile.state	265
network.device.lan.ip4config	265	network.device.wifi.accesspoints	265
network.device.lan.ip6config	265	network.device.wifi.configuration	266
network.device.lan.speed	266	network.device.wifi.enable	266
network.device.lan.stateinfo	266	network.device.wifi.ip4config	266
network.device.mobile.configuration	267	network.device.wifi.ip6config	267
network.device.mobile.hwaddress	267	network.device.wifi.maximumbitrate	267
network.device.mobile.ip4configmanual	268	network.device.wifi.state	268
network.device.mobile.ip6configmanual	268	network.device.wifi.strength	269
network.device.mobile.stateinfo	269	network.hostname	269
network.device.wifi.bitrate	270	notification.count	270
network.device.wifi.devicetype	270	optics.autofocus.calibration.calibratedlight	270
network.device.wifi.hwaddress	271	optics.autofocus.integrationinterval	271
network.device.wifi.ip4configmanual	271	optics.autofocus.maxzoom	272
network.device.wifi.ip6configmanual	272	optics.autofocus.minzoom	272
network.device.wifi.ssid	272	optics.focus.enabled	273
network.device.wifi.stateinfo	273	optics.focus.limits.reverse	273
network.domain	274	optics.focus.minposition	274
network.version	274	optics.focus.safetocalibrate	274
optics.autofocus.calibration.calibrateddark	274	optics.focus.state	275
optics.autofocus.enabled	275	optics.irisoperation	275
optics.autofocus.maxfocus	275	optics.lens.lens	275
optics.autofocus.minfocus	276	optics.lenspresent	276
optics.focus.calibrationstate	276	optics.lensshift.horizontal.enabled	276
optics.focus.limits.forward	277	optics.lensshift.horizontal.limits.reverse	277
optics.focus.maxposition	277	optics.lensshift.horizontal.minposition	277
optics.focus.position	277	optics.lensshift.horizontal.safetocalibrate	278
optics.focus.safetooperate	278	optics.lensshift.horizontal.state	278
optics.focus.target	278	optics.lensshift.vertical.calibrationstate	279
optics.irisoperationavailable	279	optics.lensshift.vertical.limits.forward	279
optics.lens.metadata	279	optics.lensshift.vertical.maxposition	280

Pulse API		Reference Guide	Version 2.2
optics.lensshift.horizontal.calibrationstate	280	optics.lensshift.vertical.position	280
optics.lensshift.horizontal.limits.forward	280	optics.lensshift.vertical.safetoooperate	281
optics.lensshift.horizontal.maxposition	281	optics.lensshift.vertical.target	281
optics.lensshift.horizontal.position	281	optics.shutter.target	282
optics.lensshift.horizontal.safetoooperate	282	optics.zoom.enabled	282
optics.lensshift.horizontal.target	283	optics.zoom.limits.reverse	283
optics.lensshift.vertical.enabled	283	optics.zoom.minposition	283
optics.lensshift.vertical.limits.reverse	284	optics.zoom.safetocalibrate	284
optics.lensshift.vertical.minposition	284	optics.zoom.state	284
optics.lensshift.vertical.safetocalibrate	285	peripheral.camera.enable	285
optics.lensshift.vertical.state	285	peripheral.camera.exposuretime	285
optics.shutter.position	285	peripheral.camera.exposuretimeminimum	286
optics.zoom.calibrationstate	286	peripheral.camera.initialised	286
optics.zoom.limits.forward	286	peripheral.ranging.lastdistance	287
optics.zoom.maxposition	287	peripheral.xlrfont.enable	287
optics.zoom.position	287	profile.domains	287
optics.zoom.safetoooperate	287	profile.profiles	288
optics.zoom.target	288	remotecontrol.broadcastaddress	288
peripheral.camera.exposuremode	288	remotecontrol.sensors.front.name	288
peripheral.camera.exposuretimemaximum	289	remotecontrol.sensors.rear.name	289
peripheral.camera.exposuretimestep	289	remotecontrol.sensors.side.name	289
peripheral.camera.resolution	289	scheduler.actiontypes	290
peripheral.ranging.sampletype	290	scheduler.scheduledactions	291
peripheral.xlrfont.outputvoltage	291	screen.hdrboost	291
profile.presetassignments	291	screen.screenaspect	291
remotecontrol.address	292	statistics.laser plate 1 bank 1 runtime.value	292
remotecontrol.sensors.front.enable	292	statistics.laser plate 1 bank 3 runtime.value	293
remotecontrol.sensors.rear.enable	293	statistics.laser plate 2 bank 2 runtime.value	293
remotecontrol.sensors.side.enable	293	statistics.laser plate 3 bank 1 runtime.value	294
scheduler.actions	294	statistics.laser plate 3 bank 3 runtime.value	294
scheduler.pause	294	statistics.laser plate 4 bank 2 runtime.value	294
scheduler.transitions	295	statistics.laser plate 5 bank 1 runtime.value	295
screen.luminance	295	statistics.laser plate 5 bank 3 runtime.value	296
screen.screensize	296	statistics.projectorruntime.value	296
statistics.laser plate 1 bank 2 runtime.value	296	statistics.uptime.value	297
statistics.laser plate 2 bank 1 runtime.value	297	system.colorwheel	297
statistics.laser plate 2 bank 3 runtime.value	297	system.date.availabletimezones	297
statistics.laser plate 3 bank 2 runtime.value	298	system.date.ntp.enabled	298
statistics.laser plate 4 bank 1 runtime.value	298	system.date.timezone	298
statistics.laser plate 4 bank 3 runtime.value	298	system.eco.enable	299
statistics.laser plate 5 bank 2 runtime.value	299	system.firmwareversion	299
statistics.laserruntime.value	299	system.license.applicable	299
statistics.systemtime.value	300	system.license.options	300
system.articlenumber	300	system.license.register.remaining	301

system.colorwheelname	301	system.license.register.valid	301
system.date.ntp.connected	301	system.modelname	301
system.date.ntp.server	302	system.on.timeout.duration	302
system.eco.available	302	system.operationalmodes.availablemodes	302
system.familyname	302	system.ready.timeout.duration	303
system.initialstate		system.resetprogress	
system.license.available		system.resetstatus	
system.license.register.mandatory		system.standby.available	
system.license.register.total		system.standby.timeout.duration	
system.license.valid		system.state	
system.name		ui.access.enduser	
system.on.timeout.enable		ui.backlight.timeout	
system.operationalmodes.mode		ui.language	
system.ready.timeout.enable		ui.layer.advancedblend.enable	
system.resetresult		ui.layer.basicblacklevel.color	
system.serialnumber		ui.layer.basicblacklevel.selection	
system.standby.enable		ui.layer.basicblend.color	
system.standby.timeout.enable		ui.layer.basicblend.selection	
system.targetstate		ui.layer.blank.border	
ui.backlight.state		ui.layer.blank.enable	
ui.hasstealthmode		ui.layer.blank.show	
ui.layer.advancedblend.drawing		ui.layer.blank.text	
ui.layer.advancedblend.palette		ui.layer.fourcorner.enable	
ui.layer.basicblacklevel.enable		ui.layer.fourcorner.lines	
ui.layer.basicblacklevel.selectioncolor		ui.layer.fourcorner.selectioncolor	
ui.layer.basicblend.enable		ui.layer.grid.enable	
ui.layer.basicblend.selectioncolor		ui.layer.grid.mark	
ui.layer.blank.bordercolor		ui.layer.grid.showlines	
ui.layer.blank.icon		ui.length	
ui.layer.blank.showtext		ui.luminance	
ui.layer.fourcorner.cornercolor		ui.menuposition	
ui.layer.fourcorner.linecolor		ui.messageposition	
ui.layer.fourcorner.selection		ui.osd	
ui.layer.grid.color		ui.poweroffhint	
ui.layer.grid.lines		ui.sourcemenu	
ui.layer.grid.points		ui.sourcesignalposition	
ui.layer.grid.showpoints		ui.splashscreen.timeoutseconds	
ui.lensmenu		ui.temperature	
ui.menu		ui.touchscreen.mode	
ui.message		user.admin.activesessioncount	
ui.minimize		user.admin.group	
ui.patternmenu		user.admin.username	
ui.showmessage		user.authenticationrequired	
ui.sourcesignal		user.currentuser.activesessioncount	

ui.splashscreen.show	user.currentuser.group
ui.stealthmode	user.currentuser.username
ui.theme	user.list
ui.touchscreen.status	user.poweruser.activesessioncount
user.admin.enabled	user.poweruser.group
user.admin.publickey	user.poweruser.username
user.admin.userslug	user.resetadministratorchallenge
user.availablegroups	user.user.enabled
user.currentuser.enabled	user.user.publickey
user.currentuser.publickey	user.user.userslug
user.currentuser.userslug	
user.pincodeauthenticationenabled	
user.poweruser.enabled	
user.poweruser.publickey	
user.poweruser.userslug	
user.user.activesessioncount	
user.user.group	
user.user.username	

Method index

.authenticate	304	.generatenewtoken	304
.getchallenge	304	.introspect	304
.login	305	.logout	305
dmx.listchannels	305	dmx.listmodes	305
environment.getalarminfo	306	environment.getcontrolblocks	306
illumination.clo.engage	308	image.color.p7.custom.copypresettocustom	308
image.color.p7.custom.resetpreset	308	image.color.p7.custom.resettonative	308
image.color.rgbmode.nextrgbmode	309	image.connector.l1displayport.edid.list	309
image.connector.l1hdbaset1.edid.list	309	image.connector.l1hdbaset2.edid.list	309
image.connector.l1hdmi.edid.list	310	image.connector.list	310
image.display.listdisplaymodes	310	image.list	311
image.listgammatypes	311	image.processing.blacklevel.basicblacklevel.getblacklevelarea	311
image.processing.blacklevel.basicblacklevel.getwarpedblacklevelarea	314	image.processing.blacklevel.file.delete	317
image.processing.blacklevel.file.list	318	image.processing.blend.basicblend.getblendarea	318
image.processing.blend.basicblend.getwarpedblendarea	323	image.processing.blend.file.delete	328
image.processing.blend.file.list	329	image.processing.warp.file.delete	329
image.processing.warp.file.list	329	image.processing.warp.fourcorners.getscaledcorners	329
image.processing.warp.warpscaledpoints	331	image.processing.warpgrid.getgrid	332
image.processing.warpgrid.getgridsize	333	image.processing.warpgrid.getscaledgrid	333
image.resolution.list	334	image.source.l1displayport.listconnectors	334
image.source.l1hdbaset1.listconnectors	335	image.source.l1hdbaset2.listconnectors	336
image.source.l1hdmi.listconnectors	337	image.source.l1quadsdi.listconnectors	338
image.source.l1sdi.listconnectors	338	image.source.l1sdib.listconnectors	339
image.source.l1sdi.listconnectors	340	image.source.l1sdi.listconnectors	341
image.source.list	342	image.stereo.listdarktime	342
image.testpattern.file.delete	342	image.testpattern.file.list	343
image.testpattern.getproperties	343	image.testpattern.iterator.calibration.listpossiblepatterns	343
image.testpattern.iterator.calibration.next	343	image.testpattern.iterator.iris.listpossiblepatterns	344
image.testpattern.iterator.iris.next	344	image.testpattern.iterator.optics.listpossiblepatterns	344
image.testpattern.iterator.optics.next	344	image.testpattern.iterator.shift.listpossiblepatterns	344
image.testpattern.iterator.shift.next	345	image.testpattern.iterator.stereo.listpossiblepatterns	345
image.testpattern.iterator.stereo.next	345	image.testpattern.list	345
image.testpattern.listinternal	346	image.testpattern.setproperties	346
image.testpattern.toggle	347	image.window.list	347
keydispatcher.postevent	347	led.activity	349
led.list	349	lightmeasurement.getlightoutput	349
network.device.wifi.connect	349	network.device.wifi.getaccesspoints	350
network.device.wifi.getsecurity	350	network.device.wifi.requestscan	351
network.list	351	notification.caution.list	351
notification.dismiss	352	notification.error.list	352
notification.info.list	353	notification.list	354
notification.listsuppressed	355	notification.log	355

Pulse API	Reference Guide		Version 2.2
notification.suppress	356	notification.unsuppress	356
notification.unsuppressall	357	notification.warning.list	357
optics.autofocus.calibration.calibrate	358	optics.focus.calibrate	358
optics.focus.runforward	358	optics.focus.runforwardtime	358
optics.focus.runreverse	359	optics.focus.runreversetime	359
optics.focus.stepforward	359	optics.focus.stepreverse	359
optics.focus.stop	360	optics.lens.list	360
optics.lensshift.horizontal.calibrate	360	optics.lensshift.horizontal.runforward	360
optics.lensshift.horizontal.runforwardtime	361	optics.lensshift.horizontal.runreverse	361
optics.lensshift.horizontal.runreversetime	361	optics.lensshift.horizontal.stepforward	361
optics.lensshift.horizontal.stepreverse	362	optics.lensshift.horizontal.stop	362
optics.lensshift.vertical.calibrate	362	optics.lensshift.vertical.runforward	363
optics.lensshift.vertical.runforwardtime	363	optics.lensshift.vertical.runreverse	363
optics.lensshift.vertical.runreversetime	363	optics.lensshift.vertical.stepforward	364
optics.lensshift.vertical.stepreverse	364	optics.lensshift.vertical.stop	364
optics.shiftocenter	364	optics.shutter.getobjectpath	365
optics.shutter.getstate	365	optics.shutter.toggle	365
optics.zoom.calibrate	365	optics.zoom.runforward	365
optics.zoom.runforwardtime	366	optics.zoom.runreverse	366
optics.zoom.runreversetime	366	optics.zoom.stepforward	366
optics.zoom.stepreverse	367	optics.zoom.stop	367
peripheral.autodetect	367	peripheral.list	368
peripheral.ranging.getdistance	368	profile.activatepreset	368
profile.activateprofile	368	profile.assignpreset	369
profile.createprofile	369	profile.deleteprofile	369
profile.getdomainsforprofile	370	profile.presetforprofile	370
profile.profileforpreset	370	remotecontrol.listsensors	371
scheduler.create	371	statistics.laser plate 1 bank 1 runtime.getname	371
statistics.laser plate 1 bank 1 runtime.getunit	372	statistics.laser plate 1 bank 2 runtime.getname	372
statistics.laser plate 1 bank 2 runtime.getunit	372	statistics.laser plate 1 bank 3 runtime.getname	373
statistics.laser plate 1 bank 3 runtime.getunit	373	statistics.laser plate 2 bank 1 runtime.getname	373
statistics.laser plate 2 bank 1 runtime.getunit	374	statistics.laser plate 2 bank 2 runtime.getname	374
statistics.laser plate 2 bank 2 runtime.getunit	374	statistics.laser plate 2 bank 3 runtime.getname	375
statistics.laser plate 2 bank 3 runtime.getunit	375	statistics.laser plate 3 bank 1 runtime.getname	375
statistics.laser plate 3 bank 1 runtime.getunit	375	statistics.laser plate 3 bank 2 runtime.getname	376
statistics.laser plate 3 bank 2 runtime.getunit	376	statistics.laser plate 3 bank 3 runtime.getname	377
statistics.laser plate 3 bank 3 runtime.getunit	377	statistics.laser plate 4 bank 1 runtime.getname	377
statistics.laser plate 4 bank 1 runtime.getunit	377	statistics.laser plate 4 bank 2 runtime.getname	378
statistics.laser plate 4 bank 2 runtime.getunit	378	statistics.laser plate 4 bank 3 runtime.getname	378
statistics.laser plate 4 bank 3 runtime.getunit	379	statistics.laser plate 5 bank 1 runtime.getname	379
statistics.laser plate 5 bank 1 runtime.getunit	379	statistics.laser plate 5 bank 2 runtime.getname	380
statistics.laser plate 5 bank 2 runtime.getunit	380	statistics.laser plate 5 bank 3 runtime.getname	380
statistics.laser plate 5 bank 3 runtime.getunit	381	statistics.laserruntime.getname	381
statistics.laserruntime.getunit	381	statistics.listcounters	382

Pulse API	Reference Guide		Version 2.2
statistics.projectorruntime.getname	382	statistics.projectorruntime.getunit	382
statistics.systemtime.getname	383	statistics.systemtime.getunit	383
statistics.uptime.getname	384	statistics.uptime.getunit	384
system.activity	384	system.boards.getboardlist	384
system.boards.getmissingboardlist	385	system.date.getlocaldate	385
system.date.getsystemdate	386	system.getidentification	387
system.getidentifications	387	system.getsystemdate	387
system.gotoeco	388	system.gotoready	388
system.license.file.saveregistrationtousb	388	system.license.file.uploadfromusb	388
system.listresetdomains	388	system.poweroff	389
system.poweron	390	system.reset	390
system.resetall	391	ui.leds.setstate	392
ui.settings.get	393	ui.settings.getfonticons	393
ui.settings.geticons	393	ui.settings.keys	394
ui.settings.list	394	ui.settings.remove	394
ui.settings.set	395	ui.togglelensmenu	395
ui.togglemenu	395	ui.toggleosd	395
ui.togglepatternmenu	396	ui.togglesourcemenu	396
ui.togglestealthmode	396	ui.touchscreen.tapped	396
user.admin.changegroup	397	user.admin.changepassword	397
user.admin.changepincode	397	user.admin.changepublickey	397
user.admin.changeusername	398	user.admin.disable	398
user.admin.enable	398	user.admin.invalidatetokens	398
user.admin.logouteverywhere	398	user.admin.removepassword	399
user.admin.removepincode	399	user.currentuser.changegroup	399
user.currentuser.changepassword	399	user.currentuser.changepincode	399
user.currentuser.changepublickey	400	user.currentuser.changeusername	400
user.currentuser.disable	400	user.currentuser.enable	400
user.currentuser.invalidatetokens	401	user.currentuser.logouteverywhere	401
user.currentuser.removepassword	401	user.currentuser.removepincode	401
user.poweruser.changegroup	401	user.poweruser.changepassword	402
user.poweruser.changepincode	402	user.poweruser.changepublickey	402
user.poweruser.changeusername	402	user.poweruser.disable	403
user.poweruser.enable	403	user.poweruser.invalidatetokens	403
user.poweruser.logouteverywhere	403	user.poweruser.removepassword	403
user.poweruser.removepincode	404	user.resetadministrator	404
user.user.changegroup	404	user.user.changepassword	404
user.user.changepincode	405	user.user.changepublickey	405
user.user.changeusername	405	user.user.disable	405
user.user.enable	406	user.user.invalidatetokens	406
user.user.logouteverywhere	406	user.user.removepassword	406
user.user.removepincode	406		

Signal index

loggedout	408	modelupdated	408
dmx.shutdownrequest	408	image.connector.l1displayport.edid.listchanged	409
image.connector.l1hdbaset1.edid.listchanged	409	image.connector.l1hdbaset2.edid.listchanged	409
image.connector.l1hdmi.edid.listchanged	409	image.processing.blacklevel.file.listchanged	409
image.processing.blend.file.listchanged	409	image.processing.warp.file.listchanged	409
image.processing.warpgrid.changed	410	image.processing.warpgrid.gridchanged	410
image.testpattern.added	410	image.testpattern.changed	411
image.testpattern.file.listchanged	411	image.testpattern.removed	411
keydispatcher.keyevent	411	network.added	412
network.removed	412	notification.caution.dismissed	412
notification.caution.emitted	412	notification.dismissed	412
notification.emitted	413	notification.error.dismissed	413
notification.error.emitted	413	notification.info.dismissed	414
notification.info.emitted	414	notification.warning.dismissed	414
notification.warning.emitted	414	profile.profileactivated	414
system.identificationchanged	414	system.license.licensechanged	414
system.performed	415	ui.settings.added	416
ui.settings.changed	416	ui.settings.removed	416

File index

bio.logger.transfer	417	image.connector.edid.transfer	417
image.processing.blacklevel.file.transfer	418	image.processing.blend.file.transfer	418
image.processing.warp.file.transfer	418	image.source.preview.transfer	419
image.source.transfer	419	image.testpattern.file.transfer	420
notification.logger.transfer	420	peripheral.camera.transfer	421
system.license.file.registration	421	ui.capture.lcd	422
ui.capture.osd	422	ui.splashscreen.file	422