

# APPLICATION PROGRAM INTERFACE MANUAL

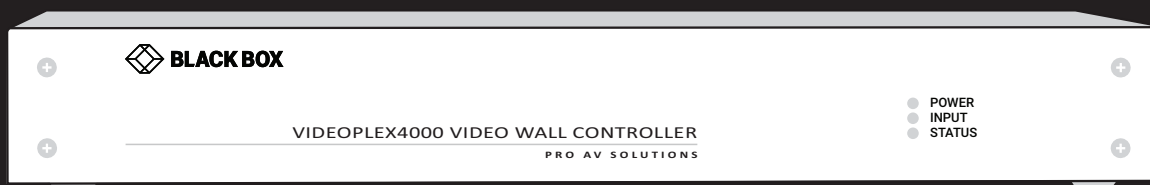
---

VSC-VPLEX4000

# VIDEOPLEX 4000 API MANUAL

---

24/7 TECHNICAL SUPPORT AT 1.877.877.2269 OR VISIT [BLACKBOX.COM](http://BLACKBOX.COM)



# TABLE OF CONTENTS

<b>1. INTRODUCTION</b> .....	<b>3</b>
1.1 VideoPlex4000 API .....	3
1.2 Rest API .....	3
<b>2. DEVICE PROPERTIES</b> .....	<b>4</b>
2.1 Serial Number .....	4
2.2 System Info .....	4
2.3 System Health .....	5
2.4 Friendly Name .....	6
2.5 Network Settings .....	7
2.6 Password .....	8
2.7 Smoothness .....	9
2.8 No Signal Behavior .....	10
2.9 Preferred Input .....	11
2.10 Active Input .....	12
2.11 Save Settings .....	13
2.12 Reset Settings .....	13
<b>3. INPUT PROPERTIES</b> .....	<b>15</b>
3.1 Input Current Timings .....	15
3.2 Input Preferred Timings .....	16
3.3 Input EDID .....	18
3.4 Input EDID Create .....	19
3.5 Input EDID Reset .....	20
3.6 Input HDCP Enable .....	21
3.7 Input HDCP Status .....	22
3.8 Input Maximum Link Rate .....	23
<b>4. OUTPUT PROPERTIES</b> .....	<b>25</b>
4.1 Output Crop .....	25
4.2 Output Transform .....	26
4.3 Output Enable .....	28
4.4 Output Current Timings .....	29
4.5 Output Preferred Timings .....	30
4.6 Output Default Timings .....	31
4.7 Output Timing Source .....	33
4.8 Output Current Timing Source .....	34
4.9 Output Image Source .....	35
4.10 Output Genlock Source .....	36
4.11 Output Genlock Status .....	37
4.12 Output HDCP Status .....	38
4.13 Output Link Training Result .....	38
<b>5. DISCLAIMER/TRADEMARKS</b> .....	<b>40</b>
5.1 Disclaimer .....	40
5.2 Trademarks Used in this Manual .....	40



# CHAPTER 1: INTRODUCTION

## 1.1 VIDEOPLEX4000 API

This document describes how to interact with the VideoPlex4000 over Ethernet using the REST API.

## 1.2 REST API

The VideoPlex4000 can be configured by issuing HTTP GET and POST requests over Ethernet. HTTP is a standard protocol which is supported natively by many different programming languages. As such, no Black Box proprietary libraries or drivers are needed to communicate with the VideoPlex4000 using this interface.

All GET requests and POST responses are encoded in JSON, which is a flexible and compact format also supported by many programming languages. For GET requests, the contents of the JSON response depends on the API call (see reference). If an error occurs something similar to the following will be returned:

```
{ "Success":false, "ErrorMessage":"Invalid input", "ErrorCode":-9 }
```

In this case, the error code is -9, which resolves to the string "Invalid input". This is a common error that will occur if a GET request is issued on a API call that requires an input number but none is given or is out of range.

Likewise, for POST responses, errors are indicated as above. In the case of success, the following response will be returned:

```
{ "Success":true }
```

Regardless of error or success, the HTTP status code returned is always 200; the body contents must be parsed to detect errors.

The functions generally fall into one of 3 categories: Device, Input and Output. The Device functions are settings that affect the device as a whole, the Input and Output functions can be set on a per input or output basis.

If a password is set on the device, then all POST requests must have a "Password" member that matches the password that has been set. See Password.cgx for more details.

## CHAPTER 2: DEVICE PROPERTIES

### 2.1 SERIAL NUMBER

Get Serial Number

GET/SerialNumber.cgx

Get the unique serial number of the device.

Example URI

GET /SerialNumber.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "SerialNumber": "123456789000DA"  
}
```

### 2.2 SYSTEM INFO

Get System Info

GET/SystemInfo.cgx

Get the firmware, config, hardware and FPGA version numbers of the device, along with the build date and time of the application and bootloader.

Example URI

GET /SystemInfo.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "FirmwareVersion": 261,  
  "ConfigVersion": 256,  
  "HardwareVersion": 1,  
  "FPGAVersion": 5,  
  "ApplicationBuildDate": "Apr 20 2016",  
  "ApplicationBuildTime": "13:12:50",
```



## CHAPTER 2: DEVICE PROPERTIES

```
"BootloaderBuildDate": "Apr 20 2016",  
"BootloaderBuildTime": "14:55:01"  
}
```

### 2.3 SYSTEM HEALTH

Get System Health

GET/SystemHealth.cgx

Get power supply state and temperature.

Example URI

GET /SystemHealth.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "PowerSupply": {  
    "IO": true,  
    "1.8V": true,  
    "1.2V": true  
  },  
  "Temperature": {  
    "Average": 34.312,  
    "Maximum": 39,  
    "Minimum": 23  
  }  
}
```

## CHAPTER 2: DEVICE PROPERTIES

### 2.4 FRIENDLY NAME

Friendly name is a user-configurable name used to identify the device. This is the name that will appear in Windows Explorer Network section under Other Devices.

Get Friendly Name

GET/FriendlyName.cgx

Example URI

GET /FriendlyName.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "FriendlyName": "VideoPlex4000"  
}
```

Set Friendly Name

POST/FriendlyName.cgx

Example URI

POST /FriendlyName.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "FriendlyName": "My Fx4"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```



## CHAPTER 2: DEVICE PROPERTIES

### 2.5 NETWORK SETTINGS

#### Get Network Settings

GET/NetworkSettings.cgx

Get the current network settings. If the device is using a static IP address, then the values returned are those configured manually. If the device is set up to use DHCP, then the values returned are those obtained from the DHCP server. The MAC address is unaffected by the IP configuration and is simply returned for information.

Example URI

GET /NetworkSettings.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "MACAddress": "00:55:DA:40:02:DA",  
  "DHCP": false,  
  "IPAddress": "192.168.1.2",  
  "Gateway": "192.168.1.1",  
  "Subnet": "255.255.255.0"  
}
```

#### Set Network Settings

POST/NetworkSettings.cgx

For dynamic IP address assignment set DHCP to true, the other fields are then not required. For static IP address configuration set DHCP to false and specify an IPAddress, Gateway and Subnet.

Example URI

POST /NetworkSettings.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "DHCP": false,  
  "IPAddress": "192.168.1.2",
```

## CHAPTER 2: DEVICE PROPERTIES

```
"Gateway": "192.168.1.1",  
"Subnet": "255.255.255.0"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 2.6 PASSWORD

Set Password

POST/Password.cgx

Password protect the VideoPlex4000. Once a password is set, all future post requests must contain a "Password" member that matches the password that has been set. If the password does not match, ERR\_INVALID\_PASSWORD will be returned. To change an existing password, the old password must be provided as a "Password" member and the new password provided in "NewPassword". If the password is forgotten, it must be reset via the USB interface. To remove password protection, set an empty string.

Example URI

POST /Password.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "NewPassword": "secret"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8





## CHAPTER 2: DEVICE PROPERTIES

Body

```
{  
  "Success": true  
}
```

### 2.7 SMOOTHNESS

Get and set the scaling optimization for video or text.

Set Smoothness to 0 for text optimization. Set Smoothness to non-zero for video optimization.

Get Smoothness

GET/Smoothness.cgx

Example URI

GET /Smoothness.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Smoothness": 0  
}
```

Set Smoothness

POST/Smoothness.cgx

Example URI

POST /Smoothness.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Smoothness": 0  
}
```

## CHAPTER 2: DEVICE PROPERTIES

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 2.8 NO SIGNAL BEHAVIOR

Get and set the no signal behavior of the outputs if no inputs are connected.

Set to true if outputs should not output a signal if no inputs are connected.

Get No Signal Behavior

GET/OutputNoSignal.cgx

Example URI

GET /OutputNoSignal.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "NoSignal": true  
}
```

Set No Signal Behavior

POST/OutputNoSignal.cgx

Example URI

POST /OutputNoSignal.cgx

Request

Show Headers

Content-Type: application/jso



## CHAPTER 2: DEVICE PROPERTIES

Body

```
{  
  "NoSignal": true  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 2.9 PREFERRED INPUT

#### Get Preferred Input

GET/PreferredInput.cgx

Get which input is preferred. Note that if the selected input has no signal, it may fall back to using a different input. ActiveInput.cgx can be used to get the currently selected input.

Example URI

GET /PreferredInput.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Input": 0  
}
```

#### Set Preferred Input

POST/PreferredInput.cgx

Select which input is preferred as the active source. If the selected input is not available, it will fall back to inputs in the following order: 2, 0, 1. A value of null can be used to indicate no preferred input.

## CHAPTER 2: DEVICE PROPERTIES

Example URI

POST /PreferredInput.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Input": 0  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 2.10 ACTIVE INPUT

Get Active Input

GET /ActiveInput.cgx

Get which input is currently being used as the source. If none of the inputs have a signal, then a value of null will be returned.

Example URI

GET /ActiveInput.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Input": 0  
}
```



## CHAPTER 2: DEVICE PROPERTIES

### 2.11 SAVE SETTINGS

Save Settings

POST/SaveSettings.cgx

Commit settings to flash memory, this must be called so settings are restored upon device reboot.

WARNING: Don't call this too frequently because there is a limited number of write operations support for the lifetime of the device. The VideoPlex4000 supports a minimum of 10K, typically 100K.

Example URI

POST /SaveSettings.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 2.12 RESET SETTINGS

Reset Settings

POST/ResetSettings.cgx

Reset all settings, apart from the network settings unless specified. To reset network settings at the same time, ResetIP must be true.

WARNING: The default network settings have DHCP enabled. If ResetIP is true and a static IP address was configured, this will be reset to DHCP enabled and your current IP address may change.

Example URI

POST /ResetSettings.cgx

## CHAPTER 2: DEVICE PROPERTIES

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "ResetIP": false,  
  "ResetFriendlyName": true  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```



## CHAPTER 3: INPUT PROPERTIES

### 3.1 INPUT CURRENT TIMINGS

Get Input Current Timings

GET/InputCurrentTimings.cgx{?input}

Get the current timings of the specified input. For an SDI input, an extra option of SDILevel is available to indicate level A or level B.

Example URI

GET/InputCurrentTimings.cgx?input=0

URI Parameters

Hide

input number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{
  "HorFrequency": 67432,
  "VerFrequency": 59939,
  "PixelClock": 148351478,
  "Flags": 0,
  "HorAddrTime": 1920,
  "HorRightBorder": 0,
  "HorFrontPorch": 44,
  "HorSyncTime": 88,
  "HorBackPorch": 148,
  "HorLeftBorder": 0,
  "VerAddrTime": 1080,
  "VerBottomBorder": 0,
  "VerFrontPorch": 4,
  "VerSyncTime": 5,
  "VerBackPorch": 36,
  "VerTopBorder": 0,
  "SDILevel": "A"
}
```

## CHAPTER 3: INPUT PROPERTIES

### 3.2 INPUT PREFERRED TIMINGS

Get and set the input preferred timings.

#### Get Input Preferred Timings

GET/InputPreferredTimings.cgx{?input}

Example URI

GET/InputPreferredTimings.cgx?input=0

URI Parameters

Hide input number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{
  "HorFrequency": 67432,
  "VerFrequency": 59939,
  "PixelClock": 148351478,
  "Flags": 0,
  "HorAddrTime": 1920,
  "HorRightBorder": 0,
  "HorFrontPorch": 44,
  "HorSyncTime": 88,
  "HorBackPorch": 148,
  "HorLeftBorder": 0,
  "VerAddrTime": 1080,
  "VerBottomBorder": 0,
  "VerFrontPorch": 4,
  "VerSyncTime": 5,
  "VerBackPorch": 36,
  "VerTopBorder": 0
}
```





## CHAPTER 3: INPUT PROPERTIES

### Set Input Preferred Timings

POST/InputPreferredTimings.cgx

Example URI

POST /InputPreferredTimings.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Input": 0,  
  "HorFrequency": 67432,  
  "VerFrequency": 59939,  
  "PixelClock": 148351478,  
  "Flags": 0,  
  "HorAddrTime": 1920,  
  "HorRightBorder": 0,  
  "HorFrontPorch": 44,  
  "HorSyncTime": 88,  
  "HorBackPorch": 148,  
  "HorLeftBorder": 0,  
  "VerAddrTime": 1080,  
  "VerBottomBorder": 0,  
  "VerFrontPorch": 4,  
  "VerSyncTime": 5,  
  "VerBackPorch": 36,  
  "VerTopBorder": 0  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

## CHAPTER 3: INPUT PROPERTIES

### 3.3 INPUT EDID

Read or write the entire EDID for an input. The contents of the EDID are Base64 encoded.

#### Read Input EDID

GET/InputEDID.cgx{?input}

#### Example URI

GET /InputEDID.cgx?input=0

#### URI Parameters

Hide input number (required) Example: 0

#### Response 200

#### Show Headers

Content-Type: application/json; charset=utf-8

#### Body

```
{  
  "EDID": "AP"  
}
```

#### Write Input EDID

POST/InputEDID.cgx

#### Example URI

POST /InputEDID.cgx

#### Request

#### Show Headers

Content-Type: application/json

#### Body

```
{  
  "Input": 0,  
  "EDID": "AP"  
}
```



## CHAPTER 3: INPUT PROPERTIES

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 3.4 INPUT EDID CREATE

POST/InputEDIDCreate.cgx

Create an EDID for a specified input. This will create an entire EDID and will not reuse any part of the existing EDID.

Extension must be "None" for no extension, "CEA-861" for a CEA-861 extension or "DisplayID" for a DisplayID extension.

Mode must be "HD" for up to and including 1920 x 1080 modes, "UHD" for up to and including 3840 x 2160 modes, "4k" for up to and including 4096 x 2160 modes and "Custom" to set a custom mode.

Name is optional but must be no more than 13 characters. It can be used by the source to identify what it is plugged in to.

If using a custom mode, the "Custom" object is mandatory. Otherwise, it will be ignored.

Example URI

POST /InputEDIDCreate.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Input": 0,  
  "Extension": "CEA-861",  
  "Mode": "Custom",  
  "Name": "CustomHDEDID",  
  "Custom": {  
    "Timings": {  
      "PixelClock": 148351478,  
      "Flags": 0,  
      "HorAddrTime": 1920,  
      "HorRightBorder": 0,  
      "HorFrontPorch": 44,  
      "HorSyncTime": 88,  
    }  
  }  
}
```

## CHAPTER 3: INPUT PROPERTIES

```
"HorBackPorch": 148,  
"HorLeftBorder": 0,  
"VerAddrTime": 1080,  
"VerBottomBorder": 0,  
"VerFrontPorch": 4,  
"VerSyncTime": 5,  
"VerBackPorch": 36,  
"VerTopBorder": 0  
},  
"SupportYCbCr4:2:0": true,  
"SupportYCbCr4:2:2": false,  
"SupportYCbCr4:4:4": true  
}  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 3.5 INPUT EDID RESET

POST/InputEDIDReset.cgx

Reset the EDID to default for a specified input or all inputs. To reset a single input's EDID, Input must be a number. To reset all input EDIDs, Input must be "All".

Example URI

POST /InputEDIDReset.cgx

Request

Show Headers

Content-Type: application/json



## CHAPTER 3: INPUT PROPERTIES

Body

```
{  
  "Input": 0  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 3.6 INPUT HDCP ENABLE

Enable or disable HDCP on an input, in some cases it may be beneficial to disable it to avoid the source attempting and potentially failing to use HDCP when it is not necessary.

**Get Input HDCP Enable**

GET/InpuHDCPEnable.cgx{?input}

Example URI

GET /InputHDCPEnable.cgx?input=0

URI Parameters

Hide input number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Enable":true,  
}
```

## CHAPTER 3: INPUT PROPERTIES

### Set Input HDCP Enable

POST/InputHDCPEnable.cgx

Example URI

POST /InputHDCPEnable.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Input":0,  
  "Enable":true,  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 3.7 INPUT HDCP STATUS

Get the status of HDCP of an input.

Status will be "Unknown", "Unencrypted", "Encrypted", "Error" or "Failure".

Get Input HDCP Status

GET/InputHDCPStatus.cgx{?input}

Example URI

GET /InputHDCPStatus.cgx?input=0

URI Parameters

Hide input number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8



## CHAPTER 3: INPUT PROPERTIES

Body

```
{  
  "Status": "Encrypted",  
}
```

### 3.8 INPUT MAXIMUM LINK RATE

Set the maximum link rate of the input. LinkRate must be "RBR", "HBR" or "HBR2".

**Get Input Maximum Link Rate**

GET/InputMaxLinkRate.cgx{?input}

Example URI

GET /InputMaxLinkRate.cgx?input=0

URI Parameters

Hide input number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "LinkRate": "HBR2",  
}
```

**Set Input Maximum Link Rate**

POST/InputMaxLinkRate.cgx

Example URI

POST /InputMaxLinkRate.cgx

Request

Show Headers

Content-Type: application/json

## CHAPTER 3: INPUT PROPERTIES

Body

```
{  
  "Input":0,  
  "LinkRate":"HBR2",  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```





## CHAPTER 4: OUTPUT PROPERTIES

### 4.1 OUTPUT CROP

The cropping region represents the subregion of the input that is to be displayed on the output.

#### Get Output Crop

GET/OutputCrop.cgx{?output}

Example URI

GET /OutputCrop.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "X0": 0,  
  "Y0": 0,  
  "X1": 1,  
  "Y1": 1  
}
```

#### Set Output Crop

POST/OutputCrop.cgx

Example URI

POST /OutputCrop.cgx

Request

Show Headers

Content-Type: application/json

## CHAPTER 4: OUTPUT PROPERTIES

Body

```
{  
  "Output": 0,  
  "X0": 0,  
  "Y0": 0,  
  "X1": 1,  
  "Y1": 1  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 4.2 OUTPUT TRANSFORM

Get and set the flip and rotation. Rotation must be 0, 90, 180 or 270. Flip must be "None", "Horizontal" or "Vertical".

This function sets the rotation for a specific output. This is useful when monitors have been physically rotated, the output can be rotated in the opposite direction to maintain the original orientation of the source. The rotation is in degrees clockwise.

For example, for an output displaying to a monitor that has been rotated 90 degrees anti-clockwise, use Rotation 90.

The Horizontal and Vertical flips are useful for projectors; they raster the horizontal or vertical in the opposite direction.

When a flip and rotation is applied simultaneously, the flip transformation is applied before the rotation.

Note that flipping both the Horizontal and Vertical is equivalent to a rotation of 180 degrees.

#### Get Output Transform

GET/OutputTransform.cgx{?output}

Example URI

GET /OutputTransform.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200



## CHAPTER 4: OUTPUT PROPERTIES

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Rotation": 0,  
  "Flip": "None"  
}
```

**Set Output Transform**

POST/OutputTransform.cgx

Example URI

POST /OutputTransform.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Output": 0,  
  "Rotation": 0,  
  "Flip": "None"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

## CHAPTER 4: OUTPUT PROPERTIES

### 4.3 OUTPUT ENABLE

Enable or disable outputs.

#### Get Output Enable

GET/OutputEnable.cgx{?output}

Example URI

GET /OutputEnable.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Enable": true  
}
```

#### Set Output Enable

POST/OutputEnable.cgx

Example URI

POST /OutputEnable.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Output":0,  
  "Enable":true,  
}
```



## CHAPTER 4: OUTPUT PROPERTIES

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 4.4 OUTPUT CURRENT TIMINGS

Get Output Current Timings

GET/OutputCurrentTimings.cgx{?output}

Get the current output timings.

Example URI

GET /OutputCurrentTimings.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "HorFrequency": 67432,  
  "VerFrequency": 59939,  
  "PixelClock": 148351478,  
  "Flags": 0,  
  "HorAddrTime": 1920,  
  "HorRightBorder": 0,  
  "HorFrontPorch": 44,  
  "HorSyncTime": 88,  
  "HorBackPorch": 148,  
  "HorLeftBorder": 0,  
  "VerAddrTime": 1080,
```

## CHAPTER 4: OUTPUT PROPERTIES

```
"VerBottomBorder": 0,  
"VerFrontPorch": 4,  
"VerSyncTime": 5,  
"VerBackPorch": 36,  
"VerTopBorder": 0  
}
```

### 4.5 OUTPUT PREFERRED TIMINGS

Get Output Preferred Timings

GET/OutputPreferredTimings.cgx{?output}

Get the output's preferred timings.

Example URI

GET /OutputPreferredTimings.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "HorFrequency": 67432,  
  "VerFrequency": 59939,  
  "PixelClock": 148351478,  
  "Flags": 0,  
  "HorAddrTime": 1920,  
  "HorRightBorder": 0,  
  "HorFrontPorch": 44,  
  "HorSyncTime": 88,  
  "HorBackPorch": 148,  
  "HorLeftBorder": 0,  
  "VerAddrTime": 1080,  
  "VerBottomBorder": 0,  
  "VerFrontPorch": 4,  
  "VerSyncTime": 5,  
}
```



## CHAPTER 4: OUTPUT PROPERTIES

```
"VerBackPorch": 36,  
"VerTopBorder": 0  
}
```

### 4.6 OUTPUT DEFAULT TIMINGS

Get and set the output's default timings. For an SDI output, an extra option of SDILevel is available to select level A or level B.

#### Get Output Default Timings

```
GET/OutputDefaultTimings.cgx{?output}
```

#### Example URI

```
GET /OutputDefaultTimings.cgx?output=0
```

#### URI Parameters

Hide output number (required) Example: 0

#### Response 200

##### Show Headers

```
Content-Type: application/json; charset=utf-8
```

##### Body

```
{  
  "HorFrequency": 67432,  
  "VerFrequency": 59939,  
  "PixelClock": 148351478,  
  "Flags": 0,  
  "HorAddrTime": 1920,  
  "HorRightBorder": 0,  
  "HorFrontPorch": 44,  
  "HorSyncTime": 88,  
  "HorBackPorch": 148,  
  "HorLeftBorder": 0,  
  "VerAddrTime": 1080,  
  "VerBottomBorder": 0,  
  "VerFrontPorch": 4,  
  "VerSyncTime": 5,  
  "VerBackPorch": 36,  
  "VerTopBorder": 0,  
  "SDILevel": "A"  
}
```

## CHAPTER 4: OUTPUT PROPERTIES

### Set Output Default Timings

POST/OutputDefaultTimings.cgx

Example URI

POST /OutputDefaultTimings.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Output": 0,  
  "HorFrequency": 67432,  
  "VerFrequency": 59939,  
  "PixelClock": 148351478,  
  "Flags": 0,  
  "HorAddrTime": 1920,  
  "HorRightBorder": 0,  
  "HorFrontPorch": 44,  
  "HorSyncTime": 88,  
  "HorBackPorch": 148,  
  "HorLeftBorder": 0,  
  "VerAddrTime": 1080,  
  "VerBottomBorder": 0,  
  "VerFrontPorch": 4,  
  "VerSyncTime": 5,  
  "VerBackPorch": 36,  
  "VerTopBorder": 0,  
  "SDILevel": "A"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8





## CHAPTER 4: OUTPUT PROPERTIES

Body

```
{  
  "Success": true  
}
```

### 4.7 OUTPUT TIMING SOURCE

Get and set the timing source. Source must be one of the following:

“Monitor”: Use the connected monitor’s preferred timings

“Default”: Use the “Default” timing specified by OutputDefaultTimings

#### Get Output Timing Source

GET/OutputTimingSource.cgx{?output}

Example URI

GET /OutputTimingSource.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Source": "Monitor"  
}
```

#### Set Output Timing Source

POST/OutputTimingSource.cgx

Example URI

POST /OutputTimingSource.cgx

Request

Show Headers

Content-Type: application/json

## CHAPTER 4: OUTPUT PROPERTIES

Body

```
{  
  "Output": 0,  
  "Source": "Monitor"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 4.8 OUTPUT CURRENT TIMING SOURCE

Get the current timing source. Source will be one of the following:

"Monitor": Use the connected monitor's preferred timings

"Default": Use the "Default" timing specified by OutputDefaultTimings

#### Get Output Timing Source

GET/OutputCurrentTimingSource.cgx{?output}

Example URI

GET /OutputCurrentTimingSource.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Source": "Monitor"  
}
```

## CHAPTER 4: OUTPUT PROPERTIES

### 4.9 OUTPUT IMAGE SOURCE

Get and set the image source. Must be "InputCapture", "Colorbars", "Greyscale", "MovingColorbars", "MovingGreyscale" or "FlatColor".

#### Get Output Image Source

GET/OutputImageSource.cgx{?output}

Example URI

GET /OutputImageSource.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{
  "Source": "InputCapture"
}
```

#### Set Output Image Source

POST/OutputImageSource.cgx

Example URI

POST /OutputImageSource.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{
  "Output": 0,
  "Source": "InputCapture"
}
```

## CHAPTER 4: OUTPUT PROPERTIES

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 4.10 OUTPUT GENLOCK SOURCE

Get and set the source for the output to genlock to. Source must be one of the following:

“None”: Do not try and genlock this output to any source

“Sync”: Try and genlock this output to the SYNC input

“Input”: Try and genlock to the current input

“Input0”: Try and genlock to input 0

“Input1”: Try and genlock to input 1

“Input2”: Try and genlock to input 2

**Get Output Genlock Source**

GET/OutputGenlockSource.cgx{?output}

Example URI

GET /OutputGenlockSource.cgx?output=0

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "GenlockSource": "Sync"  
}
```



## CHAPTER 4: OUTPUT PROPERTIES

### Set Output Genlock Source

POST/OutputGenlockSource.cgx

Example URI

POST /OutputGenlockSource.cgx

Request

Show Headers

Content-Type: application/json

Body

```
{  
  "Output": 0,  
  "GenlockSource": "Sync"  
}
```

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Success": true  
}
```

### 4.11 OUTPUT GENLOCK STATUS

Output Get Genlock Status

GET/OutputGenlockStatus.cgx

Get the genlock status. Will be "None", "Sync", "Input", "Input0", "Input1" or "Input2".

Example URI

GET /OutputGenlockStatus.cgx

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

## CHAPTER 4: OUTPUT PROPERTIES

Body

```
{  
  "GenlockStatus": "None"  
}
```

### 4.12 OUTPUT HDCP STATUS

Get the status of HDCP of an output.

Status will be "Unknown", "Unencrypted", "Encrypted", "Error" or "Failure".

Get Output HDCP Status

```
GET/OutputHDCPStatus.cgx{?input}
```

Example URI

```
GET /OutputHDCPStatus.cgx?input=0
```

URI Parameters

Hide input number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "Status": "Encrypted",  
}
```

### 4.13 OUTPUT LINK TRAINING RESULT

```
GET/OutputLinkTraining.cgx{?output}
```

For a DisplayPort output, get the link training result.

LinkRate will be "None", "RBR", "HBR" or "HBR2". LaneCount will be 0, 1, 2 or 4.

Example URI

```
GET /OutputLinkTraining.cgx?output=0
```



## CHAPTER 4: OUTPUT PROPERTIES

URI Parameters

Hide output number (required) Example: 0

Response 200

Show Headers

Content-Type: application/json; charset=utf-8

Body

```
{  
  "LinkRate": "HBR2"  
  "LaneCount": 4  
}
```

## CHAPTER 5: DISCLAIMER/TRADEMARKS

### 5.1 DISCLAIMER

Black Box Corporation shall not be liable for damages of any kind, including, but not limited to, punitive, consequential or cost of cover damages, resulting from any errors in the product information or specifications set forth in this document and Black Box Corporation may revise this document at any time without notice.

### 5.2 TRADEMARKS USED IN THIS MANUAL

Black Box and the Black Box logo type and mark are registered trademarks of Black Box Corporation.

Any other trademarks mentioned in this manual are acknowledged to be the property of the trademark owners.











**NEED HELP?  
LEAVE THE TECH TO US**

---

**LIVE 24/7  
TECHNICAL  
SUPPORT**

---

**1.877.877.2269**

