

Technical Reference

020-102207-08

Spyder X80 Serial Commands

NOTICES

COPYRIGHT AND TRADEMARKS

Copyright © 2020 Christie Digital Systems USA Inc. All rights reserved.

All brand names and product names are trademarks, registered trademarks or trade names of their respective holders.

GENERAL

Every effort has been made to ensure accuracy, however in some cases changes in the products or availability could occur which may not be reflected in this document. Christie reserves the right to make changes to specifications at any time without notice. Performance specifications are typical, but may vary depending on conditions beyond Christie's control such as maintenance of the product in proper working conditions. Performance specifications are based on information available at the time of printing. Christie makes no warranty of any kind with regard to this material, including, but not limited to, implied warranties of fitness for a particular purpose. Christie will not be liable for errors contained herein or for incidental or consequential damages in connection with the performance or use of this material. Canadian manufacturing facility is ISO 9001 and 14001 certified.

WARRANTY

Products are warranted under Christie's standard limited warranty, the complete details of which are available by contacting your Christie dealer or Christie. In addition to the other limitations that may be specified in Christie's standard limited warranty and, to the extent relevant or applicable to your product, the warranty does not cover:

- a) Problems or damage occurring during shipment, in either direction.
- b) Problems or damage caused by combination of a product with non-Christie equipment, such as distribution systems, cameras, DVD players, etc., or use of a product with any non-Christie interface device.
- c) Problems or damage caused by misuse, improper power source, accident, fire, flood, lightning, earthquake, or other natural disaster.
- d) Problems or damage caused by improper installation/alignment, or by equipment modification, if by other than Christie service personnel or a Christie authorized repair service provider.
- e) Problems or damage caused by use of a product on a motion platform or other movable device where such product has not been designed, modified or approved by Christie for such use.
- f) Except where the product is designed for outdoor use, problems or damage caused by use of the product outdoors unless such product is protected from precipitation or other adverse weather or environmental conditions and the ambient temperature is within the recommended ambient temperature set forth in the specifications for such product.
- g) Defects caused by normal wear and tear or otherwise due to normal aging of a product.

The warranty does not apply to any product where the serial number has been removed or obliterated. The warranty also does not apply to any product sold by a reseller to an end user outside of the country where the reseller is located unless (i) Christie has an office in the country where the end user is located or (ii) the required international warranty fee has been paid.

The warranty does not obligate Christie to provide any on site warranty service at the product site location.

PREVENTATIVE MAINTENANCE

Preventative maintenance is an important part of the continued and proper operation of your product. Failure to perform maintenance as required, and in accordance with the maintenance schedule specified by Christie, will void the warranty.


REGULATORY

The product has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the product is operated in a commercial environment. The product generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of the product in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at the user's own expense.

CAN ICES-3 (A) / NMB-3 (A)

이 기기는 업무용(A급)으로 전자파적합등록을 한 기기이오니 판매자 또는 사용자는 이점을 주의하시기 바라며, 가정 외의 지역에서 사용하는 것을 목적으로 합니다.

ENVIRONMENTAL

The product is designed and manufactured with high-quality materials and components that can be recycled and reused. This symbol  means that electrical and electronic equipment, at their end-of-life, should be disposed of separately from regular waste. Please dispose of the product appropriately and according to local regulations. In the European Union, there are separate collection systems for used electrical and electronic products. Please help us to conserve the environment we live in!

Revision History

Date	Version	Revision	Description
10/12/2020	8	1	<p>Added the RAS command.</p> <p>Added Frame Rates returned by RSF.</p> <p>Added format notations for commands.</p> <p>Corrected the syntax for AIR.</p> <p>Deleted the SIP command from the documentation since it was not fully supported by the software.</p>
9/10/2019	7	1	<p>Documentation Update</p> <p>Corrected an argument value for MVPR.</p> <p>Added Aux Mode support for OCM.</p> <p>Updated documentation for OCM.</p>
8/15/2019	6	1	<p>Documentation Correction</p> <p>ICR - Corrected the ConnectorType values in documentation.</p>
6/6/2019	5	1	<p>Added the following command:</p> <ul style="list-style-type: none"> • ASC: Advance Script Cue <p>Added documentation for the following commands:</p> <ul style="list-style-type: none"> • AIR-Aspect Insensitive Resize • DSL-Detailed Source List • ICS-Input Config Raster Size • OCC-Output Config Connection • OCU-Output Config Undo/Cancel • RIF-Request Image File • RPM-Request PixelSpace Mappings • RSCC-Request Script CueData Count • RSCD-Request Script CueData Details • RSEC-Request Script Element Count • RSF-Request System FrameRate • SIP-Set Input Properties • SLR-Slide Layout Recall • SRS-Stop Running Scripts • SWA-Swap Layers • TPC-Test Pattern Clear • TPL-Test Pattern Load
3/29/2019	5	1	<p>Added the following commands:</p> <ul style="list-style-type: none"> • IGP: Input Get Properties • ISP: Input Set Properties • KGP: Keyframe Get Properties • KSP: Keyframe Set Properties • OGP: Output Get Properties • OSP: Output Set Properties • COI: Capture Output Image • CII: Capture Input Image • CLI: Capture Layer Image

			OCF: Added 2 new Timing arguments (1 and 2).
10/25/2018	4	1	<p>Added the following commands:</p> <ul style="list-style-type: none"> • MVAC: Multi-Viewer Assign Content • MVAS: Multi-Viewer Add Single • MVCA: Multi-Viewer Clear All • MVKF: Multi-Viewer Set Keyframe properties • MVPL: Multi-Viewer Preset Learn • MVPR: Multi-Viewer Preset Recall • MVQO: Multi-Viewer Query Output • MVST- Multi-Viewer Set Titling
10/20/2018	3	1	<p>Added the following commands:</p> <ul style="list-style-type: none"> • OCM: to include Multiviewer • RPN: Request Page Names • RRD: Request register details
10/9/2018	2	1	<p>Added the following command:</p> <ul style="list-style-type: none"> • KTP: Keyframe Transparency Apply

Content

Communicating with Spyder.....	8
Serial connectivity.....	8
Ethernet connectivity.....	8
Command format notations.....	8
Command considerations.....	9
Command processor responses.....	10
Serial API Commands.....	11
AIR-Aspect Insensitive Resize.....	11
ARL-Apply Register to Layer.....	12
ARO-Aspect Ratio Offset.....	13
ASC-Advance Script Cue.....	14
BLD-Load Still in Background.....	15
BPL-Basic Preset Learn.....	16
BPR-Basic Preset Recall.....	17
BTR-Transition Background.....	18
CII-Capture Input Image.....	19
CLI-Capture Layer Image.....	20
COI-Capture Output Image.....	21
CRP-Crop Layer.....	22
CSO-Clear Still on Output.....	23
DCK-Delete Command Key.....	24
DMB-Device Mixer Bus Apply.....	25
DMT-Device Mixer Transition.....	26
DSL-Detailed Source List.....	27
FKR-Function Key Recall.....	28
FRZ-Freeze Layer.....	29
ICK-Input Color Key.....	30
ICL-Input Config Learn.....	31
ICR-Input Configuration Recall.....	32
ICS-Input Config Raster Size.....	33
IGP - Input Get Properties.....	34
ILA-Input Level Adjust.....	35
ILK-Input Luminance Key.....	36
IRA-Input Config Raster.....	37
ISP - Input Set Properties.....	38
KBD-Border Adjust.....	39
KGP - KeyFrame Get Properties.....	40
KPS-Layer Position Change.....	41
KSH-Shadow Adjust.....	42
KSP - Keyframe Set Properties.....	43
KSZ-Layer Size Change.....	44
KTL-Treatment Learn.....	45
KTP-Keyframe Transparency Apply.....	46
KTR-Treatment Recall.....	47
LAC-Layer Alignment Control.....	48

LAP–Layer Assign pixelspace	51
LCC–Layer Clone Control.....	52
LCK–Learn Command Key	53
LSO–Load Still on Output	54
LSP–Layer Size and Position Change.....	55
MVAC - Multi-Viewer Assign Content.....	56
MVAS – Input, Output, PGM and PVW	57
MVCA - Multi-Viewer Clear All	58
MVKF - Multi-Viewer Set Keyframe Properties	59
MVPL - Multi-Viewer Preset Learn	60
MVPR - Multi-Viewer Preset Recall	61
MVQO - Multi-Viewer Query Output	62
MVST - Multi-Viewer Set Titling.....	63
OCB–Output Configuration Blending	64
OCC–Output Config Connection	65
OCF–Output Configuration Format.....	66
OCM–Output Configuration Mode	67
OCR–Output Configuration Rotation.....	69
OCS–Output Configuration Save	70
OCU–Output Config Undo/Cancel.....	71
OFZ–Freeze Output	72
OGP – Output Get Properties	73
OSP – Output Set Properties.....	74
QRC–Query Router Crosspoint	75
RAR–Request Aspect Ratio	76
RAS–Request Active Sources	77
RBL–Request Basic Preset List	78
RCR–Router Crosspoint Recall.....	79
RCS–Request Connection Status	80
RIF–Request Image File.....	81
RLC–Request Layer Count	82
RLK–Request Layer KeyFrame	83
RLS–Request Layer Source.....	84
RPD–Request pixelspace Definitions	85
RPM–Request PixelSpace Mappings	86
RPN–Request Page Names	87
RPS–Request I/O Processor Status	88
RRC–Request Register Count	89
RRD–Request Register Detail	90
RRL–Request Register List.....	91
RSC–Recall Script Cue	92
RSCC–Request Script CueData Count	93
RSCD - Request Script CueData Details.....	94
RSEC–Request Script Element Count.....	95
RSF–Request System FrameRate	96
RSN–Request Source Name.....	97

SAV-Force Server Save	98
SCL-Clear Still on Layer.....	99
SCR-Script Cue Request	100
SDN-Restart Spyder Server	101
SLD-Load Still on Layer	102
SLR-Slide Layout Recall	103
SRS-Stop Running Scripts	104
SRA-Source Apply.....	105
SWA- Swap Layers	106
TPC-Test Pattern Clear.....	107
TPL-Test Pattern Load.....	108
TRN-Transition Layers	110
ZPA-Zoom/Pan Adjust	111

Communicating with Spyder

Understand the information and procedures for communicating with Spyder from a remote location. Connectivity is available using an RJ-45 (Ethernet) connection or a 9-pin RS232 serial connection to the frame. Both the Ethernet and serial interfaces respond to the same string commands and can do so concurrently.

Serial connectivity

One of the three RS232 serial ports available on the back of the Spyder frame can be configured to accept external control commands by using Spyder Studio.

For more information on configuring Spyder Studio for external control, see the *Spyder Studio User Manual* (P/N: 020-102205-XX).



When using serial control, terminate each command with a carriage return.

Ethernet connectivity

To control the Spyder frame, send the commands listed in this manual in a UDP packet sent to port 11116 on the frame; no configuration is required.

Each message sent to Spyder over UDP must be precluded by a 10 byte message header, provided in the table below.

Index	Character
0	s
1	p
2	y
3	d
4	e
5	r
6	0x00 (hex)
7	0x00 (hex)
8	0x00 (hex)
9	0x00 (hex)

Command format notations

This document presents commands using the following conventional notations:

- Required parameters are enclosed in angle brackets (< >).
- Optional parameters are enclosed in square brackets ([]).
- Spaces separate the name and all arguments.
- Ellipsis (...) is used to indicate additional arguments of the same syntax may be included.

Example command notation:

```
Command Name <Required Arguments> [Optional Arguments]
```


Command considerations

Note the following about the commands:

- When specifying layer IDs, the IDs start at 2. Layer ID 0 and 1 are reserved for future commands using the two background layers of the system.
- Replace the string argument spaces with the three character ASCII string %20 to create a valid command, as the space is used as the argument delimiter.
- Do not put an argument delimiter (space character) between the header and the external control command.

Command processor responses

A response is returned for every command sent to the Spyder system.

If multiple values are returned in a single response, the argument delimiter is an ASCII space character and argument values containing a space are converted to the three character ASCII string: %20.

The first response argument is always the error code for the command, which lets the user know if a command was successfully processed. The table below displays the various responses returned by the Spyder.

Response	Response name	Description
0	Success	The command was successfully processed.
1	Empty	The data requested is not available.
2	Header	An invalid command was specified.
3	Argument count	The command is missing the required minimum number of
4	Argument value	One or more arguments of the command were invalid.
5	Execution	An error occurred while processing the command. For details, check the Alert Viewer in Spyder Studio.
6	Checksum	Reserved.

Serial API Commands

The Spyder commands can be used to modify product settings.

AIR-Aspect Insensitive Resize

Resizes one or more layers independently horizontally or vertically.

Syntax

```
AIR <SizingType> <TargetDimension> <NewSize> <LayerIDs>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<SizingType>	Allows window size to be set directly or be offset in size by a specified number of pixels	0 = Absolute Size (In Pixels) 1 = Relative Size (In Pixels)
<TargetDimension>	Allows the window size change to be applied to the width or height of the window	0 = Adjust Width (Horizontal Size) 1 = Adjust Height (Vertical Size)
<NewSize>	Pixel value to set	New size (In Pixels)
<LayerID(s)>	One or more layers to apply the sizing change to	Layer ID(s) to set size on

ARL–Apply Register to Layer

Applies a specified register to one or more layers.

This is useful for recalling register data types such as sources or treatments to layers.

Syntax

```
ARL <RegisterType> <RegisterID> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<RegisterType>	Sets the register type.	0 (0x30) = Effect 1 (0x31) = Playitem 2 (0x32) = Not used 3 (0x33) = Not used 4 (0x34) = Command key/script 5 (0x35) = Treatment 6 (0x36) = Source 7 (0x37) = Function key 8 (0x38) = Not used 9 (0x39) = Not used 10 (0x31 0x30) = Still image
<RegisterID>	Sets the register ID to recall.	—
<Layer1>...<LayerX>	Sets the IDs of the layers to recall.	—

ARO–Aspect Ratio Offset

Adjusts the aspect ratio offset parameter in the KeyFrame of the specified layers.

This command can also be used to set the total aspect ratio for a layer or adjust the KeyFrame aspect ratio offset directly.

Syntax

```
ARO <Type> <FloatingPoint> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Type>	Determines the type of aspect ratio being set.	t = Sets the total aspect ratio o = Sets the KeyFrame aspect ratio offset a = Adjusts the existing KeyFrame aspect ratio offset
<FloatingPoint>	Sets the floating point aspect ratio value.	—
<Layer1>...<LayerX>	Sets the IDs of the layers to apply the aspect ratio offset to.	—

ASC-Advance Script Cue

Advances the current script cue by steps.

Syntax

ASC [<Steps>]

Response

<Result Code>

Refer to *Command processor responses*, page 13, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Steps>	Number of steps (optional)	<i>None (no number)</i> – advances the current script cue by one. <i>n</i> - to go forward <i>-n</i> - to go backward (default is 1)

BLD–Load Still in Background

Loads a file as the background of either the current or next layer.

Syntax

```
BLD <Filename> <PixelspaceID> <Layer>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Filename>	Specifies the name of the file to load. If the image is not the same size as the pixelspace it is being loaded on, the image is automatically scaled to fit into it.	—
<PixelspaceID>	Sets the ID of the pixelspace to load onto.	—
<Layer>	Determines what layer to load the file onto.	0 = Loads onto the next background layer 1 = Loads onto the current background layer

BPL–Basic Preset Learn

Stores the current screen layout to a specified preset ID.

Syntax

BPL <Preset> [<Duration>]

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Preset>	Determines the preset ID to save the current window position and KeyFrame information to.	—
<Duration>	Determines how long, in frames per second, a layer takes to transition from one KeyFrame to another. (Optional)	60 (Default)

BPR—Basic Preset Recall

Recalls an existing preset from the frame.

Syntax

```
BPR <Preset> [<Duration>]
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Preset>	Determines the preset ID to recall from the server.	—
<Duration>	Determines how long, in frames per second, a layer takes to transition from one KeyFrame to another. (Optional)	60 (Default)

BTR–Transition Background

Transitions the background layers across all pixelspaces. The pixelspace backgrounds cannot be transitioned individually.

Syntax

```
BTR <Duration>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Duration>	Determines how long, in frames per second, a background layer takes to transition.	—

CII–Capture Input Image

Captures the specified input image to the specified file name.

Syntax

```
CII <InputID> <FileName>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<InputID>	Specifies the input ID to capture.	Zero-based
<FileName>	File name for the captured image including extension.	*.bmp, *.png, *.jpg, or *.tif

CLI-Capture Layer Image

Captures the input image currently associated with a specified Layer ID to a specified file name. Note: if a still image is loaded on a layer, it will not be captured and instead the last input selected to that layer will be captured.

Syntax

```
CLI <LayerID> <FileName>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LayerID>	Specifies the layer ID to capture.	
<FileName>	File name for the captured image including extension.	*.bmp, *.png, *.jpg, or *.tif

COI–Capture Output Image

Captures the specified output image to the specified file name.

Syntax

```
COI <OutputID> <FileName>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Specifies the output ID to capture.	Zero-based
<FileName>	File name for the captured image including extension.	*.bmp, *.png, *.jpg, or *.tif

CRP–Crop Layer

Sets the left, right, top, and bottom cropping for the content of the specified layers.

Syntax

```
CRP <Left> <Right> <Top> <Bottom> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Left>	Sets the amount to crop from the left side of the	0.0 = No crop
<Right>	Sets the amount to crop from the right side of the	1.0 = 100% crop
<Top>	Sets the amount to crop from the top of the layer.	
<Bottom>	Sets the amount to crop from the bottom of the	
<Layer1>....<LayerX>	Sets the IDs of the layers to crop.	—

CSO—Clear Still on Output

Clears an image currently loaded on a specified output ID.

Syntax

```
CSO <OutputID>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the ID of the output to clear the image on.	—

DCK–Delete Command Key

Deletes an existing command key by either a register ID or script ID.

Syntax

DCK <ID> [<Type>]

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<ID>	Sets the ID to delete.	—
<Type>	Sets the type of ID. (Optional)	S = ScriptID (Default) R = RegisterID

DMB–Device Mixer Bus Apply

Allows the active bus to be selected for one or more control devices defined in the system. This command is intended for devices configured as mixers.

Syntax

DMB <Duration> <Bus> <DeviceIndex1> [<DeviceIndex2>]...

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Duration>	Sets the transition duration to be applied to layers coming on or off of the program, in frames per second.	—
<Bus>	Sets the bus state to apply to the device.	OFF = Both layers for a device are off the screen PVW = The program layer of the device is in preview and the preview layer is off the screen PGM = Both the program and preview layers are visible in their respective pixelspaces.
<DeviceIndex1>....<DeviceIndexX>	Indicates the target device indexes.	Zero-based

DMT–Device Mixer Transition

Sets the automatic transition for a mixer one or more devices. This command is intended for devices configured as mixers.

Syntax

```
DMT <Duration> <DeviceID1> [<DeviceID2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Duration>	Sets the transition duration to be applied, in frames per second.	1 = Forces a cut transition
<DeviceID1>...<DeviceIDX>	Indicates the target device indexes.	Zero-based

DSL-Detailed Source List

Returns a detailed source list.

Syntax

DSL

Response

Lists count and six information fields for each source:

<Count> <RegisterID> <SourceName> <HActive> <VActive> <Connector> <Thumbnail>

Arguments

None.

FKR—Function Key Recall

Recalls a single function key defined in Spyder.

Note: To access registers on pages above the first page, add the page number * 1000 to the register ID to be recalled. For example, to recall a function key at register ID 2 on page 3, send a register ID of 3002.

Syntax

```
FKR <ID> <Layer1> [<Layer2>]... <IDType>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<ID>	Sets the ID of the function key.	—
<Layer1>....<LayerX>	Sets the IDs of the layers for the relative function keys. (Optional)	—
<IDType>	Defines the ID type being recalled. If this argument is not specified, the function key ID is used. (Optional)	F = Function key ID (Default) R = RegisterID

FRZ—Freeze Layer

Freezes or unfreezes one or more layers.

Syntax

```
FRZ <Freeze> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Freeze>	Enables or disables freezing the specified layers.	0 = Disables freezing the specified layers 1 = Enables freezing the specified layers
<Layer1>...<LayerX>	Sets the IDs of the layers to freeze or unfreeze.	—

ICK—Input Color Key

Enables or disables and adjusts the color key for the specified layers.

If this command is used, the ILK—Input Luminance Key command is automatically disabled.

Syntax

```
ICK <KeyEnabled> <ColorRed> <ColorGreen> <ColorBlue> <RangeRed> <RangeGreen>
<RangeBlue> <ColorGain> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<KeyEnabled>	Enables or disables the color key adjustment. If disabling the color key, all key-specific arguments are not used and must be set	0 = Disables the color key adjustment 1 = Enables the color key adjustment
<ColorRed>	Sets the color-red parameter for the specified layers.	0 to 255
<ColorGreen>	Sets the color-green parameter for the specified layers.	
<ColorBlue>	Sets the color-blue parameter for the specified layers.	
<RangeRed>	Sets the range-red parameter for the specified layers.	
<RangeGreen>	Sets the range-green parameter for the specified layers.	
<RangeBlue>	Sets the range-blue parameter for the specified layers.	
<ColorGain>	Sets the color gains for the specified	0 to 512
<Layer1>....<LayerX>	Sets the IDs of the layers to apply the settings to.	—

Related information

ILK—Input Luminance Key (page 36)

ICL–Input Config Learn

Saves the input configuration for a layer ID to a configuration ID for later recall.

Syntax

```
ICL <Input> <LayerID>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Input>	Sets the input configuration ID to save to.	—
<LayerID>	Determines the layer ID to save the input configuration from.	—

ICR–Input Configuration Recall

Loads a previously saved input configuration onto the video source of a specified layer ID.

Syntax

```
ICR <ConfigurationID> <LayerID> <ConnectorType>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<ConfigurationID>	Sets the ID number to store the current input configuration of the layer ID to.	-1 = Forces an auto-synch
<LayerID>	Sets the layer ID to apply the input configuration to.	—
<ConnectorType>	If <ConfigurationID> is set to -1, sets the connector type to be switched to and auto-synched on a layer.	0 = HDMI 1 = Display Port 2 = SDI

ICS-Input Config Raster Size

Sets the raster size for the input config of a layer.

Syntax

```
ICS <LayerID> <Raster Edge> <Delta>
```

Response

None.

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Layer ID>	Layer ID to apply input raster change to.	Numeric layer index
<Raster Edge>	Side of input raster to adjust.	Top, Left, Bottom, Right
<Delta>	Value in pixels to shift the raster.	Numeric value

IGP – Input Get Properties

Returns all input config properties on the specified object. The properties are formatted in a list.

Syntax

IGP <Layer ID>

Response

<Property Name> <Property Type> <Current Value>

Arguments

Argument	Description	Values
<Layer ID>	Determines the properties of the source that the layer is using.	2 - 26

ILA—Input Level Adjust

Adjusts brightness, hue, contrast, saturation, and gamma on any layer in the system.

Syntax

```
ILA <Brightness> <Contrast> <Hue> <Saturation> <Gamma> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Brightness>	Adjusts the brightness on the specified layers.	0.0 to 2.0
<Contrast>	Adjusts the contrast on the specified layers.	0.0 to 2.5
<Hue>	Adjusts the hue on the specified layers.	-180 to 180
<Saturation>	Adjusts the saturation on the specified layers.	0.0 to 2.0
<Gamma>	Adjusts the gamma on the specified layers. (Optional) The value must be provided using a decimal (dot)	0.0 to 3.0
<Layer1>....<LayerX>	Sets the IDs of the layers to adjust.	—

ILK—Input Luminance Key

Enables or disables and adjusts the luminance key for the specified layers.

If this command is used, the ICK—Input Color Key command is automatically disabled.

Syntax

```
ILK <KeyEnabled> <Clip> <Gain> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<KeyEnabled>	Enables or disables the color key adjustment. If disabling the luminance key, all key-specific arguments are not used and must be set to zero.	0 = Disables the luminance key adjustment 1 = Enables the luminance key adjustment
<Clip>	Sets the clip for the specified layers.	0 to 512
<Gain>	Sets the gains for the specified layers.	
<Layer1>....<LayerX>	Sets the IDs of the layers to apply the settings to.	—

Related information

ICK—Input Color Key (page 30)

IRA–Input Config Raster

Allows the video shown inside of a window to be repositioned on a per edge basis.

Additionally, use this command to perform an auto raster (analog only), in which a specified layer inspects the video content currently in the layer and repositions the input video automatically.

Syntax

```
IRA <LayerID> <Edge> <Delta>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LayerID>	Sets the ID of the layer to adjust.	—
<Edge>	Sets the edge to adjust.	L = Left R = Right T = Top B = Bottom A = AutoRaster
<Delta>	Sets the number of pixels to move.	negative number = Moves the video edge inward positive number = Moves the video edge outward

ISP – Input Set Properties

Sets the named properties on the input config object for the specified layer.

Syntax

```
ISP <LayerID> [<PropertyName> <PropertyValue>]
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Layer ID>	Determines the source config object on the layer.	2 - 26

KBD–Border Adjust

Adjusts one or more border properties on a layer.

Syntax

KBD <ID> <BorderThickness> <Red> <Green> <Blue> <H-BevelOffset> <V-BevelOffset> <InsideSoftness>

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<ID>	Sets the ID of the layer to adjust.	—
<BorderThickness>	Determines the thickness of the border. Set a negative number for outside softness.	-255 to 255
<Red>	Sets the red parameter for the RGB color value. If specifying the red parameter, the blue and green parameters must also be set.	0 to 255
<Green>	Sets the green parameter for the RGB color value. If specifying the green parameter, the blue and red parameters must also be set.	
<Blue>	Sets the blue parameter for the RGB color value. If specifying the blue parameter, the red and green parameters must also be set.	
<H-BevelOffset>	Sets the horizontal bevel offset.	
<V-BevelOffset>	Sets the vertical bevel offset.	
<InsideSoftness>	Sets the sharpness of the inside edge of the border.	

KGP – KeyFrame Get Properties

Returns all KeyFrame properties for the specified object. The properties are formatted in a list.

Syntax

KGP <LayerID>

Response

<Property Name> <Property Type> <Current Value>

Arguments

Argument	Description	Values
<Layer ID>	Determines the properties of the source that the layer is using.	2 - 26

KPS–Layer Position Change

Sets the horizontal and vertical position of one or more layers. Note the following about positions:

- Positions are mapped in pixels, relative to the top-left pixelspace corner associated with the layer.
- Position changes can be relative to the layer's current position or can be an absolute position setting.

Syntax

KPS <Position> <Horizontal> <Vertical> <Layer1> [<Layer2>]...

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Position>	Determines if the position is absolute or relative to the current position of the layer.	0 = Absolute position in pixels 1 = Relative position in pixels
<Horizontal>	Sets the horizontal position.	—
<Vertical>	Sets the vertical position.	—
<Layer1>...<LayerX>	Sets the IDs of the layers.	—

KSH–Shadow Adjust

Adjusts the shadow parameters for a layer.

Syntax

```
KSH <ID> <Horizontal> <Vertical> <Size> <Transparency> <Softness>
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<ID>	Sets the ID of the layer to adjust.	—
<Horizontal>	Sets the horizontal position of the shadow.	0 to 255
<Vertical>	Sets the vertical position of the shadow.	
<Size>	Sets the size of the shadow.	
<Transparency>	Sets the transparency of the shadow.	
<Softness>	Sets the outside softness of the shadow.	

KSP – Keyframe Set Properties

Sets the named properties on the Keyframe object for a specified layer.

Syntax

```
KSP <LayerID> [<PropertyName> <PropertyValue>]
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Layer ID>	Determines the source config object on the layer.	2 - 26

KSZ–Layer Size Change

Sets the horizontal size of one or more specified layers.

The vertical size adjusts automatically to ensure the layer's aspect ratio.

Syntax

```
KSZ <Horz> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Horz>	Sets the horizontal size, in pixels, of the specified layers.	—
<Layer1>....<LayerX>	Sets the IDs of the layers to size.	—

KTL–Treatment Learn

Learns the KeyFrame attributes for a specified layer.

Syntax

KTL <Treatment ID> <Layer> [Treatment Name] <Argument3>...<Argument12>

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Treatment ID>	Sets the treatment ID.	-1 = Next available treatment ID
<Layer>	Specifies the layer ID to learn from.	—
[Treatment Name]	Optional name for the treatment	Text value used as treatment name. Do Not use pure numerical names for the treatment, as this may cause parsing errors on the server.
<Argument3>...<Argument12>	Learns KeyFrame attributes.	Argument3 = Learns the position attribute Argument4 = Learns the crop attribute Argument5 = Learns the clone attribute Argument6 = Learns the border attribute Argument7 = Learns the shadow attribute Argument8 = Learns the A/R attribute Argument9 = Learns the Pan/Zoom attribute Argument10 = Learns the Duration attribute Argument11 = Learns Transparency attribute Argument12 = Learns Size attribute

KTP–Keyframe Transparency Apply

Sets KeyFrame transparency value on layers.

Syntax

KTP <transparency Value> <LayerID(s)>

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Transparency Value>	0 - 255.	
<LayerID (s)>	Which layer or layers to apply transparency value.	Layer IDs are 2 based. For example, Layer 1 is ID 2, Layer 5 is ID 6.

KTR–Treatment Recall

Recalls a treatment to one or more layers.

Syntax

```
KTR <TreatmentID> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<TreatmentID>	Sets the ID of treatment to recall.	—
<Layer1>...<LayerX>	Sets the IDs to recall the treatment to.	—

LAC–Layer Alignment Control

Allows one or more layer KeyFrames to be manipulated by the Spyder Studio to achieve a specified alignment effect.

The alignment effects are the same as the ones available from the simulator control of Spyder Studio.

Syntax

```
LAC <EffectID> <RecallDuration> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Arguments	Description	Values	Minimum layers
<EffectID>	Sets the alignment effect.	0 (Align Bottom) = Repositions specified layers vertically to make the bottom edge of the layers match the first specified layer	2
		1 (Align Center) = Repositions specified layers horizontally to make the center position of all layers match the first specified layer	
		2 (Align Left) = Repositions specified layers horizontally to make the left edge of all layers match the first specified layer	
		3 (Align Middle) = Repositions specified layers vertically to make the center position of all layers match the first specified layer	
		4 (Align Right) = Repositions all specified layers horizontally to make the right edge of all layers match the first specified layer	
		5 (Align Top) = Repositions specified layers vertically to make the top edge of layers match the first specified layer	
		6 (Center Horizontal) = Repositions all specified layers as a group so that layers are centered horizontally in their current pixelspace	1
		7 (Center Vertical) = Repositions all specified layers as a group so that layers are centered vertically in their current pixelspace	
		8 (Horizontal Decrement) = Decreases the horizontal spacing between two or more layers	2
9 (Horizontal Increment) = Increases the horizontal spacing between two or more layers			

Arguments	Description	Values	Minimum layers
		10 (Make Horizontal Equal) = Makes the horizontal spacing between all specified layers equal to the spacing between the first and second specified layers	3
		11 (Make Same Height) = Makes all specified layers the same height as the first specified layer, maintaining aspect ratio in all layers	2
		12 (Make Same Width) = Makes all specified layers the same height as the first specified layer, maintaining aspect ratio in all layers	
		13 (Make Vertically Equal) = Makes the vertical spacing between all specified layers equal to the spacing between the first and second specified layers	3
		14 (Remove Horizontal Spacing) = Removes spacing between specified layers, causing them to be horizontally stacked in the order specified	2
		15 (Remove Vertical Spacing) = Removes spacing between specified layers, causing them to be vertically stacked in the order specified	
		16 (Size to Display Height) = Resizes specified layers to fill their respective pixelspace vertically; no adjustment is made to the horizontal position	1
		17 (Size to Display Width) = Resizes specified layers to fill their respective pixelspace horizontally, and centers the input vertically	
		18 (Snap to Bottom) = Repositions specified layers vertically to align bottom edges of layers with the bottom of their pixelspaces	
		19 (Snap Left) = Repositions the specified layers horizontally to align left edges of layers with the left edge of their pixelspaces	
		20 (Snap Right) = Repositions the specified layers horizontally to align right edges of layers with the right edge of their pixelspaces	
		21 (Snap Top) = Repositions the specified layers vertically to align top edges of the layers with the top edge of their pixelspaces	
		22 (Stack Horizontal) = Makes all layers the same height, centers them vertically with the first specified layer, and positions them in a horizontal array running to the right of the first layer	2

Arguments	Description	Values	Minimum layers
		23 (Stack Vertical) = Makes all layers the same width, centers them horizontally with the first specified layer, and positions them in a vertical array running downward starting at the first layer's position	
		24 (Swap Windows) = Swaps horizontal position and size between the two specified layers	
<RecallDuration>	Sets the duration for the alignment recall, in frames per second.	—	—
<Layer1>...<Layer X>	Sets the IDs of the layers to crop.	—	—

LAP–Layer Assign pixelspace

Associates a layer with a particular pixelspace.

Syntax

```
LAP <ID> <Visibility> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Syntax

```
LCC <LayerID> <Mode> [<Offset>] <OffsetType>
```

Arguments

Argument	Description	Values
<ID>	Sets the pixelspace to associate the layer with.	—
<Visibility>	Hides or shows the layers at the time this command is applied. Making the layer visible allows additional commands to be sent to configure the layer before it is transitioned	0 = Hides the layers (Default) 1 = Makes the layers visible
<Layer1>...<LayerX>	Sets the IDs of the layers to associate with the pixelspace.	—

Related information

TRN–Transition Layers (page 106)

LCC–Layer Clone Control

Adjusts the clone property on the KeyFrame of a specified layer.

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LayerID>	Sets the ID of the layer that the KeyFrame is on.	—
<Mode>	Sets the clone mode. If the mode is specified with no offset value, the clone offset value stored in the layer's current KeyFrame is used.	0 = Off 1 = Offset 2 = Mirror
<Offset>	Sets the offset distance in relative coordinates or the pixel value for the clone position. (Optional)	—
<OffsetType>	Determines the type of clone offset.	0 = Absolute (in pixels) 1 = Relative coordinate (Default)

LCK—Learn Command Key

Learns a new command key from one of the Spyder Studio interfaces.

Syntax

```
LCK <LearnAs> <Name> <RegisterID> <LearnFrom> <LearnMixers>
```

Response

```
<Result Code> <CommandKey ID> <Script ID>
```

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LearnAs>	Determines if the command key is learned as absolute or relative.	0 = Absolute 1 = Relative
<Name>	Sets the name of the command key to be learned.	—
<RegisterID>	Sets the ID of the register to learn the command to.	—
<LearnFrom>	Determines what the command key is learned from.	1 = Preview only 2 = Program only 3 = Both
<LearnMixers>	Determines if the command key is learned as a mixer.	0 = False 1 = True

LSO—Load Still on Output

Loads an unscaled still image directly onto a Spyder output for the purpose of loading custom test patterns.

Syntax

```
LSO <Filename> <OutputID>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Filename>	Sets the file of the image to load. The filename must exist in the Stills directory on the Spyder server, which can be accessed using FTP or the Spyder Studio software.	—
<OutputID>	Sets the ID of the output to load the image on.	—

LSP–Layer Size and Position Change

Sets the size and horizontal and vertical position of one or more layers.

In cases where the size and position are being modified simultaneously for an application, Christie recommends using this command instead of the individual layer size and position commands (KPS and KSZ).

Syntax

```
LSP <Position> <Horizontal> <Vertical> <Size> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Position>	Determines the position.	0 = Absolute position 1 = Relative position to the current position of the layer
<Horizontal>	Sets the horizontal position in pixels.	—
<Vertical>	Sets the vertical position in pixels.	—
<Size>	Sets the horizontal layer size in pixels.	—
<Layer1>...<LayerX>	Sets the IDs of the layers.	—

Related information

KPS–Layer Position Change (page 40)

KSZ–Layer Size Change (page 43)

MVAC - Multi-Viewer Assign Content

Assigns an input to an existing input view on a Multiviewer.

Syntax

```
MVAC <OutputID> <ViewID> <InputID>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
OutputID	Sets the ID of the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16
ViewID	ID of Multiviewer input view.	ID based upon user defined in the Multiviewer properties
InputID	ID of the Input to assign.	0 Based ID

MVAS – Input, Output, PGM and PVW

Adds any View to the Multiviewer.

Syntax

```
MVAS <OutputID> <ViewType> <ContentID> <Left> <Top> <Width> <Height>
[<BorderThickness> <BorderColorRed> <BorderColorGreen> <BorderColorBlue>]
```

Syntax for Custom PGM/PVW

```
MVAS <OutputID> <ViewType> <-1> <Left> <Top> <Width> <Height> <SectionLeft>
<SectionTop> <SectionWidth> <SectionHeight> [<BorderThickness> <BorderColorRed>
<BorderColorGreen> <BorderColorBlue>]
```

Response

```
<Result Code> <ViewID>
```

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16
<ViewType>	Sets the Multiviewer element type.	0 = PGM 1 = PVW 2 = Input 3 = Output
<ContentID>	Assigns the desired element ID to use.	PGM and PVW Content ID can be found in the system patch section by clicking on the desired pixelspace. Input and output ID's are 0 based.
<Left> <Top>	Positioning of the view type.	Coordinates based off of Multiviewer output resolution.
<Width> <Height>	Size of the view type.	Desired view resolution
Optional: <BorderThickness>	Gives the view a boarder around it.	0-100
Optional: <BorderColorRed> <BorderColorGreen> <BorderColorBlue>	Defines a color for the border.	0-255

MVCA - Multi-Viewer Clear All

Clears all view types on desired Multiviewer output.

Syntax

MVCA <OutputID>

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16

MVKF - Multi-Viewer Set Keyframe Properties

Sets the position, size, border thickness and color of the View Type.

Syntax

```
MVKF <OutputID> <ViewID> <Left> <Top> <Width> <Height> <BorderThickness>
<BorderColorRed> <BorderColorGreen> <BorderColorBlue>]
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
OutputID	Sets the ID of the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16
ViewID	ID of Multiviewer View.	ID based upon user defined value in the Multiviewer properties.
<Left> <Top>	Position of the View.	Dependent on the size of the Multiviewer output resolution. 0x0 is the top left position.
<Width> <Height>	Size of the View	Desired view resolution.
<BorderThickness>	Gives the view a boarder around it.	0-100
<BorderColorRed> <BorderColorGreen> <BorderColorBlue>	Defines a color for the boarder.	0-255

MVPL - Multi-Viewer Preset Learn

Learns a current viewing Multiviewer preset.

Syntax

```
MVPL <OutputID><PresetName>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
OutputID	Sets the ID of the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16
<PresetName>	The name of the desired preset to learn.	A unique preset name.

MVPR - Multi-Viewer Preset Recall

Recalls desired Multiviewer preset to any Multiviewer output.

Syntax

```
MVPR <OutputID> <PresetType> [<PresetName>]
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
OutputID	Sets the ID of the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16
PresetType	Type of Multiviewer Preset.	0 = OpMon 1 = SourceMon 2 = Custom
<PresetName>	The current name of the desired custom preset to learn.	

MVQO - Multi-Viewer Query Output

Queries the current output ID shown on the Multiviewer.

Syntax

MVQO <OutputID>

Response

<Result Code> <BackgroundColorRed> <BackgroundColorGreen> <BackgroundColorBlue> <View Count> [<ViewID> <ViewType> <ContentId> <Left> <Top> <Width> <Height> <BorderThickness> <BorderColorRed> <BorderColorGreen> <BorderColorBlue>]...

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16

MVST - Multi-Viewer Set Titling.

Adds the names and numbers to the view elements on the Multi-viewer.

Syntax for PGM/PVW

MVST <OutputID> <ViewID> <ShowPixelSpaceName>

Syntax for Input

MVST <OutputID> <ViewID> <ShowInputNumber><ShowSourceName>

Syntax for Output

MVST <OutputID> <ViewID> <ShowOutputNumber><ShowOutputName>

The rest of arguments for all types: [<ShowCustomLabel> [<CustomText>] <LabelPosition> <FontSize> <Bold> <Italic> <Separator> <ForegroundColorRed> <ForegroundColorGreen> <ForegroundColorBlue> <BackgroundColorRed> <BackgroundColorGreen> <BackgroundColorBlue>]

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Note: <CustomText> is required when <ShowCustomLabel> is enabled.

Arguments

Argument	Description	Values
<OutputID>	Sets the output.	3 = output 4 7 = output 8 11 = output 12 15 = output 16
ViewID	ID of Multiviewer View.	ID based upon user defined value in the Multiviewer properties.
LabelPosition	Position of the view labels.	0 = TopIn 1 = BottomIn 2 = LeftIn 3 = RightIn 4 = TopOut 5 = BottomOut 6 = LeftOut 7 = RightOut

OCB–Output Configuration Blending

Enables or disables the blending on a Spyder output.

Additionally, the blend width, type, and curve parameters can be adjusted.

Syntax

```
OCB <OutputID> <Edge> <Enable> <BlendWidth> <Mode> <Curve1> <Curve2>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Output>	Sets the ID of the output to configure.	Zero-based
<Edge>	Sets the edge of the blending area of the output.	L = Left R - Right
<Enable>	Enables or disables the blending of the output.	0 = Disables the blending of the output 1 = Enables the blending of the output
<BlendWidth>	Sets the width of the blend in pixels.	—
<Mode>	Sets the mode of the blend.	Bezier Gamma Velocity
<Curve1>	Sets the curve parameter 1.	0.000 to 1.000
<Curve2>	Sets the curve parameter 2.	

OCC-Output Config Connection

Enables and disabled connector types for a specified output.

Note: the Spyder X80 platform does not allow enabling multiple connector ports at the same time, and when executing this command against X80 the specified connector type will always be set to the active connector type, regardless of the enabled parameter

Syntax

```
OCC <Output ID> <Connector Type> <Enabled>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Output ID>	Output index to query.	Numeric output index
<Connector Type>	Output connector type to set.	0: Analog 1: DVI 2: SDI 3: Composite / S-Video 4: HDMI 5: Display Port
<Enabled>	Enables / Disables the specified connector type. Note: this flag is not supported by all hardware platforms (X80), in which case the connector type set will always be treated as enabled when this API call is made.	0: Disabled 1: Enabled

OCF–Output Configuration Format

Sets the video format being output by a specified output.

Currently this command supports configuring VESA output formats only. Additionally, Spyder outputs support specific output refresh rates, and in cases where an unsupported refresh rate is specified, the closest available refresh value is used.

Syntax

```
OCF <OutputID> <HActive> <VActive> <RefreshRate> <Interlaced> [<Timing>]
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the output ID.	Zero-based
<HActive>	Sets the horizontal active resolution.	—
<VActive>	Sets the vertical active resolution.	—
<RefreshRate>	Sets the refresh rate. Supports floating point	—
<Interlaced>	Sets the interlaced format.	0 = Non-interlaced 1 = Interlaced
<Timing>	Sets the reduced blanking timing.	0 = Off 1 = Reduced Level 1 2 = Reduced Level 2

OCM–Output Configuration Mode

Sets the output mode for a specified output.

The arguments for this command vary depending on the mode being applied.

Syntax for Normal Mode

```
OCM <OutputID> <Mode> [<HStart> <VStart>]
```

Response for Normal Mode

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments for Normal mode

Argument	Description	Values
<OutputID>	Sets the ID of the output.	Zero-based
<Mode>	Sets the output mode to Normal.	Normal
<HStart>	Sets the horizontal starting position for the output.	—
<VStart>	Sets the vertical starting position for the output.	—

Syntax for Multiviewer Mode

```
OCM <OutputID> <Mode> [<PresetName>]
```

Response for Multiviewer Mode

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments for Multiviewer Mode

Argument	Description	Values
<OutputID>	Sets the ID of the output.	3, 7, 11, 15 (zero based, 4 th port only)
<Mode>	Sets the output mode to Multiview.	Multiview
<PresetName>	The current name of the desired custom preset to load.	—

Syntax for Scaled Mode

```
OCM <OutputID> <Mode> [<PixelspaceID>]
```

Response Scaled Mode

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments for Scaled Mode

Argument	Description	Values
<OutputID>	Sets the ID of the output.	Zero-based
<Mode>	Sets the output mode to Scaled.	Scaled
<PixelspaceID>	Sets the program pixelspace to Scaled on. If a preview ID is supplied, its associated program pixelspace is applied.	—

Syntax for OpMon Mode

OCM <OutputID> <Mode> [<PixelspaceID>]

Response for OpMon Mode

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments for OpMon Mode

Argument	Description	Values
<OutputID>	Sets the ID of the output.	Zero-based
<Mode>	Sets the output mode to OpMon (X20) or Multiviewer with OpMon preset type (X80).	OpMon
<PixelspaceID>	Optional. Sets the program pixelspace to focus OpMon. If a preview ID is supplied, its associated program pixelspace is applied. Default: ALL PGM/PVW.	—

Syntax for SourceMon Mode

OCM <OutputID> <Mode>

Response for SourceMon Mode

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments for SourceMon Mode

Argument	Description	Values
<OutputID>	Sets the ID of the output.	Zero-based
<Mode>	Sets the output mode to SourceMon (X20) or Multiviewer with SourceMon preset type (X80).	SourceMon

Syntax for AUX Mode

OCM <OutputID> <Mode> [<SourceName> <BackgroundColorRed> <BackgroundColorGreen> <BackgroundColorBlue>]

Response for AUX Mode

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments for AUX Mode

Argument	Description	Values
<OutputID>	Sets the ID of the output.	Zero-based
<Mode>	Sets the output mode to Aux.	Aux
<SourceName>	Source name from Sources list.	—
<BackgroundColorRed> <BackgroundColorGreen> <BackgroundColorBlue>	Defines a color for the background.	0-255

OCR–Output Configuration Rotation

Sets the rotation parameter of a specified output.

Rotation is not supported on all output module types. For additional information, contact Christie Technical Support.

Syntax

```
OCR <OutputID> <RotationAngle>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the output ID.	Zero-based
<RotationAngle>	Sets the angle of the rotation in 90 degree increments.	0 90 180 270

OCS–Output Configuration Save

Forces the active configuration for an output to be saved to the persistent storage on Spyder.

To prevent changes from being lost when the system is restarted, call this command after making output configuration adjustments.

Syntax

```
OCS <OutputID>
```

Response

```
<Result Code>
```

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<OutputID>	Sets the ID of the output to save.	Zero-based

OCU-Output Config Undo/Cancel

Cancels any unsaved modifications to the output configuration. It is equivalent to clicking the cancel button on the output config panel in the Spyder software client.

Syntax

```
OCU <Output ID>
```

Response

```
<Result Code>
```

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Output ID>	Output ID to cancel changes on.	Numeric output index

OFZ–Freeze Output

Freezes or unfreezes one or more outputs.

This command is only compatible with universal outputs. DX4 outputs do not support individual freeze or unfreeze functionality.

Syntax

```
OFZ <Freeze> <Output1> [<Output2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Freeze>	Enables or disables freezing the specified layers.	0 = Disables freezing the output 1 = Enables freezing the output
<Output1>....<OutputX>	Sets the IDs of the outputs to freeze or unfreeze.	—

OGP – Output Get Properties

Returns all output properties for the specified object. The properties are formatted in a list.

Syntax

```
OGP <LayerID>
```

Response

```
<Property Name> <Property Type> <Current Value>
```

Arguments

Argument	Description	Values
<Layer ID>	Determines the properties of the source that the layer is using.	2 - 26

OSP – Output Set Properties

Sets named properties on the output config object for the specified output.

Syntax

```
OSP <Output ID> [<PropertyName> <PropertyValue>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Output ID>	Determines the output config object ID.	0 - 15

QRC–Query Router Crosspoint

Queries one or all router outputs for their currently connected input. Note the following about inputs and outputs:

- For level controlled routers, the input returned is the input connected to the output on the level configured to be controlled by Spyder.
- If no specific output is supplied, the inputs for all outputs are returned.
- Inputs and outputs returned are zero indexed. For example, output 1 on the router is returned as a zero (0).
- An output with no input connected (disconnected) returns -1 for the input.

Syntax

```
QRC <RouterID> [<OutputID>]
```

Response

```
<Result Code> <Router ID> <Output>:<Input> [<Output>:<Input>]
```

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<RouterID>	Sets the router ID to query.	—
<OutputID>	Sets the router output to query the status for.	—

RAR–Request Aspect Ratio

Requests the aspect ratio of a defined source.

Syntax

RAR <Layer>

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
RAR <Layer>	Sets the ID of the layer or the source name to request the aspect ratio	—

RAS–Request Active Sources

Allows for lightweight status request for sources visible in program and/or preview PixelSpaces. Useful for showing active status for Spyder sources in a third-party user interface, or it can be used with an external system to trigger a tally device.

Syntax

```
RAS [<PixelSpace ID> ...]
```

Response

```
<Result Code> <Count> [<Source Name> <Bus> ...]
```

Arguments

Argument	Description	Values
<PixelSpace ID>	Sets the PixelSpace ID(s) to query the active sources.	Default = all PixelSpaces

Returned values

Respon	Description	Values
Source	String name of source.	
Bus	Indicates if the specified source is visible on a Program PixelSpace, a Preview PixelSpace, or both.	0 = Preview 1 = Program 2 = Both

RBL–Request Basic Preset List

Returns a list of preset names and their associated IDs.

As preset lists can be long, additional arguments are available to request portions of the list at a time.

Syntax

RBL [<StartIndex>] [<MaxCount>] [<Chars>]

Response

<Result Code> <Return Count> [<Preset1 ID> <Preset1 Name>]...

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<StartIndex>	Determines which index to begin returning the list of preset names	—
<MaxCount>	Sets the maximum number of registers to return.	—
<Chars>	Sets the number of characters to truncate names to.	—

RCR–Router Crosspoint Recall

Switches crosspoints on a router connected to the Spyder frame.

The <output> and <input> arguments can be repeated as many times as required to stack a series of switches into a single command. If the router and the Spyder control protocol for the router support stack and trigger switching, Spyder uses this functionality automatically when sending a command with multiple switch assignments.

Syntax

```
RCR <RouterID> [<Switch>] <Output> <Input>
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<RouterID>	Sets ID of the router to switch.	—
<Switch>	Determines which patch is switched. If this argument is not specified, the logical patch is switched. (Optional) A logical output refers to the router patch configured from Spyder Studio, stored as part	L = Switch logical Output (Default) P = Switch Physical Outputs
<Output>	Sets the output to switch.	Zero-based
<Input>	Sets the input to switch.	Zero-based

RCS–Request Connection Status

Allows for the current connected or disconnected status of a specified input to be queried.

When called, the system polls the current connector type of the specified input in an attempt to determine if a video source is connected. Do not call this command too frequently (more than once per second) as system performance may degrade.

Syntax

RCS <LayerID>

Response

<Result Code> <Layer ID> <Current Connector Type> <Connection Status>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LayerID>	Sets the ID of the layer to check the connection status on.	—

Returned values

Response	Values
Connection status	0 = Disconnected 1 = Connected 2 = Unknown
Current connector type	0 = HDMI 1 = Display Port 2 = SDI

RIF-Request Image File

Allows an image file to be returned from the internal still image folder on the Spyder system. Image files are always normalized to a .PNG format, regardless of their original format. The dimensions of the image may additionally be downscaled to a specified maximum width or height.

Note: The byte formatting returned from this command is an ASCII encoding of hexadecimal numbers. For example, the following four bytes '0x00 0xAB 0xAC 0xFF' would be returned as the string literal '00ABACFF'.

Note: This command should be used sparingly, and it is recommended that the client device cache images locally to avoid regular re-downloading when possible. Heavy use of this command, particularly when image scaling is in use, may have a performance effect on the Spyder server.

Syntax

RIF <Image File Name (No Path)> [Max Width or Height]

Response

<ASCII encoded hexadecimal characters>

Arguments

Argument	Description	Values
<FileName>	File name of the image being requested. Do not include file path.	String filename
[Maximum width or height]	Optionally, specify maximum width or height of the image. Image scaling will maintain aspect ratio. Useful for grabbing thumbnails.	Numeric value

RLC—Request Layer Count

Retrieves the logical layer count within the connected Spyder system.

Syntax

RLC

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
RLC	Retrieves the logical layer count within the connected Spyder system. The logical layer count includes two background layers in Spyder. In most cases, applications should subtract two from the response.	—

RLK—Request Layer KeyFrame

Retrieves the KeyFrame values for a layer ID.

Christie recommends implementing the code to handle additional values which may be appended to the end of the responses in future versions.

Syntax

RLK <LayerID>

Response

<Result Code> <Relative HPosition> <Relative VPosition> <X Position> <Y Position>
 <Width> <Height> <Border Thickness> <Border Red> <Border Green> <Border Blue> <Border
 HBezel Offset> <Border VBezel Offset> <Border Inside Softness> <Border Outside
 Softness> <OutSide Edges> <Shadow HOffset> <Shadow VOffset> <Shadow HSize> <Shadow
 Softness> <Shadow Transparency> <Clone Mode> <Clone Offset> <Left Crop> <Right Crop>
 <Top Crop> <Bottom Crop> <Crop Anchor> <AR Offset> <Zoom> <HPan> <VPan> <Pixelspace
 ID> <Transparency>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LayerID>	Specifies the ID of the layer to retrieve the KeyFrame values	—

Returned values

Response	Values
<Outside Edges>	Provided as a hexadecimal value, with the lower four-bits indicating specific edges enabled. <ul style="list-style-type: none"> • 0x01—Top enabled • 0x02—Bottom enabled • 0x04—Left enabled • 0x08—Right enabled
<Clone Mode>	0 = Off 1 = Offset 2 = Mirror
<Crop Anchor>	0 = Input center 1 = Window center

RLS–Request Layer Source

Retrieves the current source name and associated source register ID loaded on a specified layer. Note the following about returned results:

- If no source is currently assigned to the specified layer, an empty result code is returned with no parameters.
- If a source is assigned to the specified layer but no corresponding register can be found, 1 is returned for the register ID.

Syntax

RLS <LayerID>

Response

<Result Code> <Source Name> <Source Register ID>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<LayerID>	Specifies the ID of the layer to retrieve the information for.	—

RPD–Request pixelpspace Definitions

Returns a list of all the pixelpspaces currently defined in the system.

Syntax

```
<Result Code> <Count> [<ID> <Name> <Current Background> <Next Background> <X
Position> <YPosition> <Width> <Height> <RenewalGroup ID>]...
```

Response

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
RPD	Returns a list of all the pixelpspaces currently defined in the	—

RPM-Request PixelSpace Mappings

This command returns the mappings between preview and program PixelSpace IDs. PixelSpaces with no associated preview / program PixelSpace will not be included in the response.

Syntax

RPM

Response

<Result Code> <Count> [<ProgramID> <PreviewID> <Preview Scale Ratio>]...

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

None.

RPN-Request Page Names

This command returns the names and page numbers for a specified register list.

Syntax

RPN <RegisterType>

Response

<Result Code> <Register Count> [<Register Page ID> <Register Page Name>]...

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Argument	Description	Values
<RegisterType>	Sets the register type.	0 (0x30) = Effect 1 (0x31) = Playitem 2 (0x32) = Not used 3 (0x33) = Not used 4 (0x34) = Command key/script 5 (0x35) = Treatment 6 (0x36) = Source 7 (0x37) = Function key 8 (0x38) = Not used 9 (0x39) = Not used 10 (0x31 0x30) = Still image

RPS–Request I/O Processor Status

Queries the current state of the I/O processor.

Asynchronous operations, such as still loading, are processed serially as background tasks in Spyder. You can query the current state of the I/O processor to use in external state logic such as needing to wait for the image load commands to complete.

The status returned includes a numeric value from 0 to 101 giving the percent progress as well as an ASCII string with a generic description of the task being processed. When idle, this request returns a progress/message response of 0/<empty string> or 101/Ready.

Syntax

RPS [<Chars>]

Response

<Result Code> <Progress> <Status Message>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Chars>	Sets the number of characters to truncate names to.	—

RRC–Request Register Count

Returns the number of registers for a specified data type, optionally on a specific page.

Syntax

RRC <RegisterType> [<PageNumber>]

Response

<Result Code> <Register Count>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<RegisterType>	Sets the register type.	0 (0x30) = Effect 1 (0x31) = Playitem 2 (0x32) = Not used 3 (0x33) = Not used 4 (0x34) = Command key/script 5 (0x35) = Treatment 6 (0x36) = Source 7 (0x37) = Function key 8 (0x38) = Not used 9 (0x39) = Not used 10 (0x31 0x30) = Still image
<PageNumber>	Determines the page number to return.	Zero-based -1 = Return all pages

RRD-Request Register Detail

Returns detailed information specific to the data stored within a specified register type and ID. The format of the response data is dependent on the register type requested, and the individual response formats register types supported under this command are documented below. Data associated with register types may be enhanced in the future, and callers supporting this function should implement their code to handle additional information which may be appended to the end of the responses in future versions of the Spyder server.

Syntax

RRD <RegisterType> <Register ID>



Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Response (Cmd Keys)

<Result Code> <Script ID> <0=Absolute/1= Relative> <Cue Count>

Response (Treatments)

<Result Code> <Size Enabled (1/0) > <Position Enabled (1/0)> <Border Enabled (1/0)> <Shadow Enabled (1/0)> <Clone Enabled (1/0)> <Crop Enabled (1/0)> <A/R Enabled (1/0)> <Pan/Zoom Enabled (1/0)> <Duration (1/0)>

Response (Source)

<Result Code> <Source Name> <Router ID> <Router Input> <Input Config ID> <Preferred Treatment ID> <Preferred Layer ID> <HActive> <VActive> <Frame Rate>

Response (Function Key)

<Result Code> <Function Key ID> <Function Key Type (string name)> <User Defined Name>

Response (Still)

<Result Code> <FileName> <Exists at server (0/1)> <Width> <Height> <File Size (bytes)>

Argument	Description	Values
<RegisterType>	Sets the register type	0 (0x30) = Effect 1 (0x31) = Playitem 2 (0x32) = Not used 3 (0x33) = Not used 4 (0x34) = Command key/script 5 (0x35) = Treatment 6 (0x36) = Source 7 (0x37) = Function key 8 (0x38) = Not used 9 (0x39) = Not used 10 (0x31 0x30) = Still image
<RegisterID>		

RRL–Request Register List

Returns a list of registers and their associated IDs for a specified data type.

As register lists can be long, additional arguments are available to request portions of the list at a time.

Syntax

```
RRL <RegisterType> [<PageNumber>] [<StartIndex>] [<MaxCount>] [<Chars>]
```

Response

```
<Result Code> <Return Count> [<Register1 ID> <Register1 Name>]...
```

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<RegisterType>	Sets the register type.	0 (0x30) = Effect 1 (0x31) = Playitem 2 (0x32) = Not used 3 (0x33) = Not used 4 (0x34) = Command key/script 5 (0x35) = Treatment 6 (0x36) = Source 7 (0x37) = Function key 8 (0x38) = Not used 9 (0x39) = Not used 10 (0x31 0x30) = Still image
<PageNumber>	Sets the page number to return. (Optional)	Zero-based index -1 = All pages
<StartIndex>	Sets the index to begin returning.	—
<MaxCount>	Sets the maximum number of registers to return.	—
<Chars>	Sets the number of characters to truncate register names to.	Positive integer

RSC–Recall Script Cue

Recalls an existing script at a specific cue. To build scripts, use Spyder Studio.

Syntax

```
RSC <Script> <Cue> [<Type>]
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Script>	Sets the ID of the script being recalled.	—
<Cue>	Sets what script cue to recall.	—
[Type]	Defines the ID type being recalled being recalled in the <Script> argument. (Optional) To access registers on pages above the first page, add the page number * 1000 to the register ID to be recalled.	S = ScriptID (Default) R = RegisterID

RSCC–Request Script CueData Count

Returns the number of cues in a specified command key script.

Syntax

```
RSCC <Script ID> [Script or Register ID]
```

Response

<Number of cues in the requested script>

Arguments

Argument	Description	Values
<Script / Register ID>	ID to query.	Numeric script or register ID
[ID Type]	Specifies if the ID is a script or register ID.	S: Script ID (Default) R: Register ID

RSCD - Request Script CueData Details

Returns details on a specified script cue. Note that control clients should format their code to support additional returned values in the response.

Syntax

RSCC <Script ID> [Script or Register ID]

Response

<Name> <JumpType> <JumpCue> <TriggerType> <TriggerTime> <Function Key Count> [<Function Key Register ID>...] <PlayItemType> [<PlayItem Register ID> <PlayItem Command>...]

Response	Codes and Format
<i>JumpType</i>	<i>None=0</i> <i>Direct=1</i>
<i>TriggerType</i>	<i>None=0</i> <i>Wait=1</i> <i>WaitX=2</i> <i>Time=3</i> <i>Time Of Day=4</i>
<i>TriggerTime</i>	<i>HH:MM:SS.FF (Hours:Minutes:Seconds.Frames)</i>
<i>PlayItem</i>	<i>Cue=0</i> <i>Play=1</i> <i>Stop=2</i>

Arguments

Argument	Description	Values
<ID>	Script or register ID to query.	Numeric script or register ID
<Cue Index>	Index of cue to query.	Numeric cue index
[ID Type]	Specifies that the ID provided is a script ID or a register ID.	S: Script ID (Default) R: Register ID

RSEC-Request Script Element Count

Returns the number of elements in a specified command key script.

Syntax

```
RSCC <Script ID> [Script or Register ID]
```

Response

<Number of elements in the requested script>

Arguments

Argument	Description	Values
<Script ID>	Script or Register ID.	Numeric script or register ID
[ID Type]	Specifies that the ID provided is a script ID or a register ID.	S: Script ID (Default) R: Register ID

RSF-Request System FrameRate

Allows the system to be interrogated for its current frame rate setting. The number returned from this command is the Frame Rate that is used in Frame Setup of the Configuration Manager. .

Syntax

RSF

Response

<System Frame Rate ID>

Frame Rate	ID
FR_60	0
NTSC	1
PAL	2
FR_48	3
FR_30	4
FR_29_97	5
FR_25	6
FR_24	7
FR_23_98	8

Arguments

None.

RSN–Request Source Name

Retrieves a list of sources defined within Spyder.

Syntax

RSN

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
RSN	Returns a list of sources names separated by spaces.	—

SAV–Force Server Save

Forces the remote Spyder frame to save all configuration and user data changes to non-volatile storage.

Syntax

SAV

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
SAV	Forces the remote frame to save all configuration and user changes to non-volatile storage.	—

SCL–Clear Still on Layer

Clears any loaded still images from one or more layers.

Syntax

```
SCL <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Layer1>...<LayerX>	Sets the IDs of the layers.	—

SCR–Script Cue Request

Requests the current execution cue of a specified script ID.

Returns a single integer value representing the current cue of the specified script or the command returns -1 if the specified script is not being executed on any layer.

Syntax

```
SCR <ID> <Type>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<ID>	Sets the script ID to request the status for.	—
<Type>	Defines the ID type for the specified script.	S = ScriptID R = RegisterID

SDN–Restart Spyder Server

Restarts or powers off the Spyder server application remotely.

Syntax

SDN <0 | 1>

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<0 1>	Remotely restarts or powers off the server application.	0 = Powers the server off 1 = Restarts the server

SLD–Load Still on Layer

Loads a file onto one or more layers.

Syntax

```
SLD <Filename> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Filename>	Sets the name of the file to load.	—
<Layer1>...<LayerX>	Sets the IDs of the layers.	—

SLR-Slide Layout Recall

Used initially by the SpyderPoint add-in for Microsoft PowerPoint, this command initiates a PowerPoint-like effect transition of layers onto a specified PixelSpace. Note that layer ID(s) used are dynamically selected, but ‘reserved’ layers can be specified which will not be used for layer slider operations (typically for background layers)

Syntax

```
SLR <PixelSpace ID> <Clear Layers> <Reserved Layer Count> [<Reserved Layer> ...]
[<Slide Layout Entry> ...]
```

Response

None.

Refer to *Command processor responses*, page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<PixelSpace ID>	ID of the PixelSpace to execute a slide layout recall to.	Numeric PixelSpace ID
<Clear Layers>	Determines whether all layers should be cleared from the PixelSpace.	0: Leave unused layers 1: Remove unused layers
<Reserved Layer Count>	Number of reserved layers being set in the next parameter(s)	Numeric count
<Reserved Layers>	One or more layer IDs which will not be used as part of the slide recall.	Numeric layer ID. This is documented as a single parameter. However, it is intended that the number of parameters for reserved layers is based on the provided reserved layer count argument.
<Layer Slide Entries>	One or more slide operations to be performed.	Slide entries have their own tilde (~) separated format, and each must contain exactly the following values: <ol style="list-style-type: none"> 1. Source Name 2. ZOrder 3. X Position 4. Y Position 5. Border Color R (0-255) 6. Border Color G (0-255) 7. Border Color B (0-255) 8. Border Thickness 9. Slide Transition Type (integer) 10. Transition Duration (frames)

SRS-Stop Running Scripts

Allows for either stopping running scripts (cue trigger countdowns) or disconnects scripts from layers.

Syntax

```
SRS <Operation Type>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Operation Type>	Specifies whether to cancel cue triggers or to disconnect layers from scripts.	D: Disconnect layers from scripts S: Cancel pending cue triggers

SRA–Source Apply

Applies an existing source to one or more specified layers.

A source is defined as an input configuration and a router input. To create sources and define connected routers, use Spyder Studio.

Syntax

```
SRA <Name> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Name>	Provides the name of the source.	—
<Layer1>...<LayerX>	Sets the IDs of the layers to apply the source to.	—

SWA- Swap Layers

Swaps all properties between two specified layers, including PixelSpace, content, and keyframe. Note: both layers must be visible on screen when the command is executed, or an ArgValue response will be returned.

Syntax

```
SWA <First Layer ID> <Second Layer ID>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<First Layer ID>	First layer of the two-layer swap.	Numeric layer ID
<Second Layer ID>	Second layer of the two-layer swap.	Numeric layer ID

TPC-Test Pattern Clear

Clears test patterns from a layer, PixelSpace, or output.

Syntax

```
TPC <Target> <ID>
```

Response

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Target>	Specifies whether to clear test patterns on outputs, PixelSpaces, or layers.	0: PixelSpace 1: Layer 2: Output
<ID>	Target ID to clear.	Numeric ID

TPL-Test Pattern Load

Loads the specified test pattern to a specified layer, pixelspace, or output.

Syntax

```
TPL <Target> <ID> <Pattern Type> [Outline] [CenterCircle] [CenterX] [Grid] [BG Red]
[BG Green] [BG Blue] [FG Red] [FG Green] [FG Blue]
```

Response

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Target>	Specifies whether to clear test patterns on outputs, PixelSpaces, or layers.	0: PixelSpace 1: Layer 2: Output
<ID>	Target ID to load.	Numeric ID
[Pattern Type]	Type of pattern to load	0: ColorBarsGray 1: ColorBarsHorizontal 2: ColorBarsPluge 3: SingleColor 4: ColorRamp8Bit 5: RGBColorGradient8Bit 6: Grid32 7: Grid64 8: Grid128 9: GrayGamma 10: Gray Top to Bottom 16 steps 11: Gray Bottom to top 16 steps 12: Gray Top to Bottom 32 steps 13: Gray Bottom to Top 32 steps 14: Gray Split 16 steps 15: Gray Split 32 steps 16: Grill H 17: Grill V 18: Sweep Left to Right 19: Sweep Right to Left 20: Sweep Top to Bottom 21: Sweep Bottom to Top
[Outline]	If enabled, a 1 pixel outline will be drawn around the test pattern using the foreground color.	0: disabled 1: enabled

[CenterCircle]	If enabled, an ellipse will be drawn to fill the test pattern using the foreground color.	0: disabled 1: enabled
[CenterX]	If enabled, an 'X' will be drawn to fill the test pattern image using the foreground color.	0: disabled 1: enabled
[Grid]	If enabled, a 32x32 pixel grid will be drawn over the test pattern using the foreground color.	0: disabled 1: enabled
[BG Red]	Background Red Color	0-255
[BG Green]	Background Green Color	0-255
[BG Blue]	Background Blue Color	0-255
[FG Red]	Foreground Red Color	0-255
[FG Green]	Foreground Green Color	0-255
[FG Blue]	Foreground Blue Color	0-255

TRN–Transition Layers

Transitions layers on and off of their currently assigned pixelspace.

Syntax

```
TRN <Mix> <Duration> <Layer1> [<Layer2>]...
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<Mix>	Transitions layers on and off of their currently assigned pixelspace.	0 = Mix off 1 = Mix on
<Duration>	Sets the transition duration in number of frames per second.	—
<Layer1>... <LayerX>	Sets the IDs of the layers to transition.	—

ZPA–Zoom/Pan Adjust

Changes the zoom and pan KeyFrame controls on a specified layer.

Syntax

```
ZPA <RecallMode> <Zoom> <HorizontalPan> <VerticalPan> <LayerID>
```

Response

<Result Code>

Refer to [Command processor responses](#), page 10, for additional information on Result Codes.

Arguments

Argument	Description	Values
<RecallMode>	Determines if adjustments are relative to the existing zoom/pan settings of the layer or can be set to specific values directly.	0 = Absolute values 1 = Relative adjustment
<Zoom>	Sets the zoom value.	0.0 to 20.0
<HorizontalPan>	Sets the horizontal pan value.	-2048 to 2048
<VerticalPan>	Sets the vertical pan value.	-2048 to 2048
<LayerID>	Sets the layer of the KeyFrame to adjust.	—

Corporate offices

USA – Cypress
ph: 714-236-8610
Canada – Kitchener
ph: 519-744-8005

Consultant offices

Italy
ph: +39 (0) 2 9902 1161

Worldwide offices

Australia
ph: +61 (0) 7 3624 4888
Brazil
ph: +55 (11) 2548 4753
China (Beijing)
ph: +86 10 6561 0240
China (Shanghai)
ph: +86 21 6278 7708

Eastern Europe and
Russian Federation
ph: +36 (0) 1 47 48 100
France
ph: +33 (0) 1 41 21 44 04
Germany
ph: +49 2161 664540

India
ph: +91 (080) 6708 9999
Japan (Tokyo)
ph: 81 3 3599 7481
Korea (Seoul)
ph: +82 2 702 1601
Republic of South Africa
ph: +27 (0)11 510 0094

Singapore
ph: +65 6877-8737
Spain
ph: + 34 91 633 9990
United Arab Emirates
ph: +971 4 3206688
United Kingdom
ph: +44 (0) 118 977 8000