



REDLINK COMMAND PROTOCOL: REFERENCE GUIDE

REVISION HISTORY

REV.	ECO NO.	DATE	DESCRIPTION
A	005267	07/19/12	Formal document release.
B	005471	09/21/12	Document reformatted.
C	005749	11/27/12	Document reformatted.
D	005790	12/04/12	Updated to for camera firmware 4.0.x, Ethernet references removed.
E	007389	10/07/13	Updated to extend parameter set, Ethernet configuration added.
F	007621	12/18/13	Updated to extend parameter set to match firmware 5.1.33 and later.
G	007697	01/20/14	Updated for parameter set 4.1 (firmware 5.1.35) and corrections to FANPCTT, FANPCTF, MPITCH, RCLIP, GCLIP, BCLIP
H	007827	02/21/14	Added discovery process, use of target values. Corrections or additional info to CAMNAME, SHANGLE, SHTIME, PAUSE, PLAY, PLAYPAUS, STOP, CAMMEID, CAMSCENE, CAMTAKE. Key release codes added to buttons.
H1	007832	02/24/14	Change to PDF settings.
J	007996	04/01/14	Name change, Many parameters added, T/H/D message types added, WiFi setup added.
K	008197	05/16/14	Updated for RCP Parameter Set 5.1. Added HCFOCUS, HCIRIS, HCZOOM and HCREINIT parameters and corrected type for MMMODE.
L	008645	09/05/14	Updated for RCP Parameter Set 6.0. Large number of parameters added. Change to discovery process.
M	009405	04/08/15	Updated for RCP Parameter Set 6.10. Large number of parameters added. Relative value update supported on some parameters. Five parameters marked for deprecation.
N	009840	07/23/15	Updated for RCP Parameter set 6.20. Corrected DATETIME description, added approximately 100 parameters.
P	010193	10/29/15	Updated for RCP Parameter set 6.30.

CONFIDENTIAL AND PROPRIETARY

This document and the information contained herein is confidential and proprietary, and is contractually protected from disclosure. Distribution other than by written agreement is prohibited.

DOCUMENT IS UNCONTROLLED IF PRINTED OR DOWNLOADED. VERIFY REVISION BEFORE USE.



TABLE OF CONTENTS

DISCLAIMER	4
COPYRIGHT NOTICE	4
TRADEMARK DISCLAIMER	4
OBJECTIVE	5
SCOPE	5
FEATURES	5
ACCESSING RCP IN EPIC/SCARLET	6
VIA SERIAL PORT	6
VIA GIGE	7
VIA WIFI	8
RCP VERSIONING	9
REDLINK COMMAND PROTOCOL SYNTAX	9
STRUCTURE	9
DATA TYPES	10
NUMERIC (MSG_INT AND MSG_UINT)	10
STRING (MSG_STR_PTR)	11
LIST (MSG_LIST_PTR)	11
DECORATED STRING TYPE	12
SPECIAL AGGREGATE TYPE	13
RELATIVE PARAMETER UPDATING	13
USAGE	13
VOLUME OF TRAFFIC	14
SOURCE CODE PACKAGE AVAILABILITY	14
DEFINED PARAMETER GROUPS	14
SYSINFO GROUP	14
EXAMPLES OF SYSINFO USAGE	15
USING CAMINFO TO DISCOVER CAMERAS	15
KEYS GROUP	16
EXAMPLES OF KEYCODE USAGE	18
EXAMPLES OF KEYACT USAGE	18
EXPOSURE GROUP	18
USING TARGET SETTINGS	19
EXAMPLE OF TURNING ON HDRX	19
IMAGE GROUP	19
EXAMPLE OF USING CURVE PARAMETER	22



LENS GROUP	23
OVERLAY GROUP	24
EXAMPLE OF CHANGING FRAME GUIDE OVERLAY	26
PROJECT GROUP	27
EXAMPLES OF SETTING METADATA	27
EXAMPLE OF CREATING FORMAT2 VALUE FIELD	30
EXAMPLE OF SETTING TIME CODE	30
STATE GROUP	31
EXAMPLES OF CONTROLLING RECORDING	31
SYSTEM GROUP	32
EXAMPLE OF DECODING TIME CODE FROM POSTTC	39
EXAMPLE OF MEDAGCII EXTENDED CLIP DATA	39
USE OF NOTIFY AND NOTIFYR	40
PLAYBACK GROUP	41
PLAYBACK OPERATION	42
EXAMPLE OF CHANGE TO PLAYBACK AND PLAY LAST CLIP	42
STATS GROUP	42
CLIP INDICATORS	43
DECODING DSHIST DATA	43
DECODING AUDIOVU DATA	44
AUDIO GROUP	45
OUTPUT GROUP	47
EXAMPLE OF SELECTING OVERLAY	50
ALGORITHM GROUP	50



DISCLAIMER

RED® has made every effort to provide clear and accurate information in this document, which is provided solely for the user's information. While thought to be accurate, the information in this document is provided strictly "as is" and RED will not be held responsible for issues arising from typographical errors or user's interpretation of the language used herein that is different from that intended by RED. All safety and general information is subject to change as a result of changes in local, federal or other applicable laws.

RED reserves the right to revise this document and make changes from time to time in the content hereof without obligation to notify any person of such revisions or changes. In no event shall RED, its employees or authorized agents be liable to you for any damages or losses, direct or indirect, arising from the use of any technical or operational information contained in this document.

For comments or questions about content in this document please send a detailed email to rcpsdk@red.com.

COPYRIGHT NOTICE

© 2015 RED.COM, INC.

All trademarks, trade names, logos, icons, images, written material, code, and product names used in association with the accompanying product are the copyrights, trademarks or other intellectual property owned and controlled exclusively by RED.COM, INC.

TRADEMARK DISCLAIMER

All other company, brand and product names are trademarks or registered trademarks of their respective holders. RED has no affiliation to, is not associated or sponsored with, and has no express rights in third-party trademarks. Android is a trademark of Google Inc. HDMI is a registered trademark of HDMI Licensing LLC in the United States and other countries. IOS is a registered trademark of Cisco in the U.S. and other countries and is used under license. LEMO is a registered trademark of LEMO USA. OS X is a registered trademark of Apple Inc. in the U.S. and other countries. Windows is a registered trademark of Microsoft Corporation.



OBJECTIVE

The purpose of this document is to describe the structure and usage of the REDLINK® Command Protocol (RCP). RCP is the unified command protocol used for the RED EPIC® and SCARLET cameras, regardless of sensor type.

SCOPE

Applies to WEAPON®, RAVEN™, EPIC-X, EPIC-M, and SCARLET-X® cameras that have either MYSTERIUM-X® or RED DRAGON® sensors, with RCP Parameter Set 5.0 and later.

FEATURES

- ▶ The current version is RCP2.
- ▶ RCP is supported via the serial port, GigE (Gigabit Ethernet), and REDLINK Bridge (WiFi).
- ▶ Designed as a lightweight protocol that can be used from a PC down to a microcontroller.
- ▶ RCP is text based and human readable.
- ▶ Includes optional error detecting checksum.
- ▶ Asynchronous.
- ▶ No master/slave concept.
- ▶ Support for 1:1 and 1:N communication.



ACCESSING RCP IN EPIC/SCARLET

VIA SERIAL PORT

Starting with camera firmware 3.3.x, the RCP is available through the CTRL connector (RS232 port) on the rear of the camera body. The RCP is disabled by default, and must be enabled in the DSMC[®] menu.

WEAPON and RAVEN cameras require a module that supports the user serial port. Currently this includes the DSMC^{2™} Base Expander, DSMC² Jetpack Expander, DSMC² Base I/O V-Lock Expander and the DSMC² REDVOLT[®] Expander.

1. Go to **Menu > Settings > Setup > Communication**.
2. Select the **Serial** tab.
3. Select **REDLINK Command Protocol** in the **Serial Protocol** drop-down menu.

Once selected, REDLINK Command Protocol is persistent across boots.

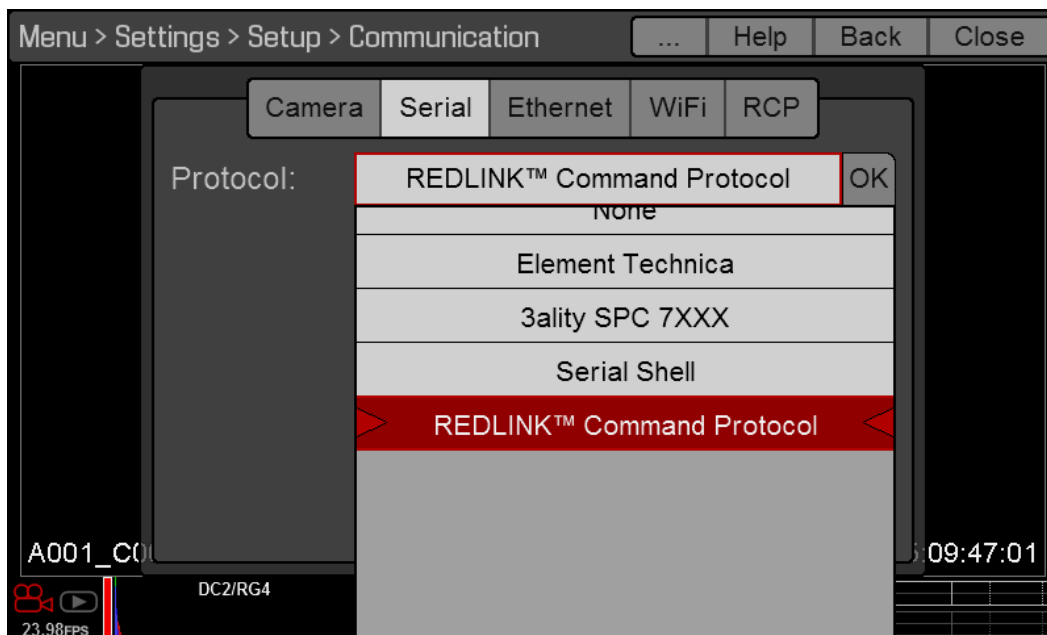


Figure 1. Serial Port Configuration Menu.

The port settings are 115200 baud, 8N1 (8 bits, no parity, 1 start bit, no flow control).

The 4-PIN 00 LEMO[®]-to-Flying Lead (P/N 790-0187) may be used for custom connections. See the [RED DSMC Operation Guide: Epic/Scarlet](#), section “CTRL (RS232 Control)” for details.



VIA GIGE

RCP is available through the GigE connector on the rear of the camera, and must be enabled in the camera menu. For WEAPON and RAVEN, the DSMC² REDVOLT Expander module is required.

1. Go to **Menu > Settings > Setup > Communication**.
2. Select the **Ethernet** tab.
3. Select the **Enable External Control** check box.

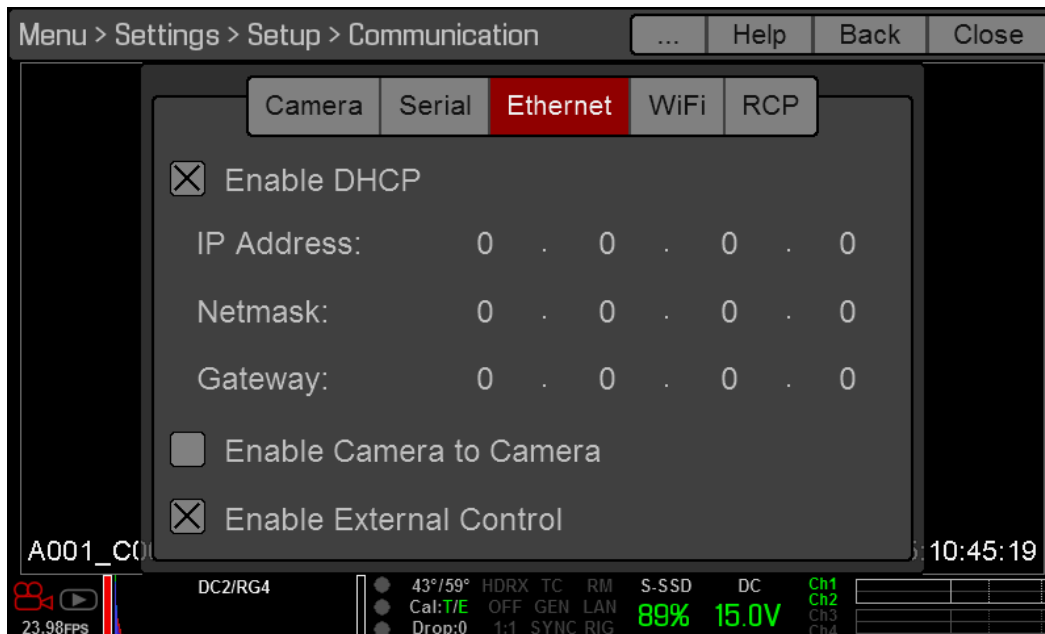


Figure 2. Ethernet Configuration Menu.

Communication is allowed over TCP (port 1111). It is your responsibility to configure the camera, network, and application properly. The camera can be configured for a fixed or dynamic IP address. A dynamic **discovery process** is also supported.

The LEMO-to-CAT5E Ethernet Cable (9') (P/N 790-0159) is required. See the **RED DSMC Operation Guide: EPIC/SCARLET**, section “GIG-E (Ethernet)” for details. The GIG-E 9-pin LEMO connector provides a 1000BASE-T (IEEE 802.3ab) gigabit Ethernet. Since the GIG-E connector does not support slower speeds (10BASE-T and 100BASE-T), ensure that any device you connect to supports 1000BASE-T.

Check that the LAN indicator turns green when External Control is enabled and the camera is connected to a network. If the LAN indicator turns yellow or red, contact a Bomb Squad representative to determine if the camera needs updating.

As of firmware version 5.1.33, the camera does not support more than eight (8) simultaneous connections. Your application should maintain just one. If the connection is lost, close down the connection before opening a new one. Once the maximum number of connections is reached, the camera ignores additional requests. Other than refusing the connection, the camera does not indicate why the connection is unsuccessful.



VIA WIFI

Starting with camera firmware 5.2.x, the RCP is available through the REDLINK Bridge module for EPIC and SCARLET cameras. The REDLINK Bridge is automatically enabled if connected, but you must enable WiFi and the connection method in the camera menu. The REDLINK Bridge can support one connection.

The WEAPON and RAVEN cameras have WiFi built in and also require enabling and specifying the connection method.

1. Go to **Menu > Settings > Setup > Communication**.
2. Select the **WiFi** tab.
3. Select the **Enable WiFi** check box.
4. Select **Ad-Hoc** or **Infrastructure** mode and set up the connection using the instructions from the [REDLINK Bridge Operation Guide](#).



Figure 3. WiFi Configuration Menu.



RCP VERSIONING

There is versioning at two levels in RCP. One is the version of RCP itself. This is currently at version 2. This will change with wholesale changes in the architecture of RCP. It can be queried with the RCPVER parameter. The other is versioning of the RCP parameter set. The parameter set changes over time with evolution of camera features.

The parameter set is versioned to allow you to determine compatibility with your application. This version can be retrieved from the RCPPSVER parameter. It is maintained as a major and minor version number packed as a 32 bit value with the major version in the high word and minor in the low. So Version = major <<16 + minor. A major version is incremented when a potentially non-backward compatible change is made (something removed or behavior changed). The minor version is incremented when the only change is an addition. In other words, major revision changes have the potential to break an application’s use of RCP. When writing applications with RCP, build in the check of this parameter and take appropriate action if a dependency on a particular version is found.

REDLINK COMMAND PROTOCOL SYNTAX

STRUCTURE

The general structure of an RCP Message looks like this:

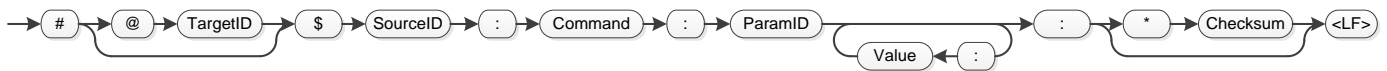


Figure 4. RCP Syntax.

RCP SYNTAX COMPONENTS

CHARACTER	COMMENTS	MANDATORY
#	Message start marker. Has to start each RCP Message	Yes
@	'@' followed by Target ID name. This is the intended target of the message. If it does not exist the message is to be interpreted by all receivers (Broadcast message) The Target ID can be up to eight (8) alphanumeric characters. For the purposes of external device, omit this field.	No
\$	'\$' followed by Source ID name. This identifies the sender of the message. The source ID can be up to eight (8) alphanumeric characters long. Use 'EXT' for creating messages to the camera. On responses it will be the camera name, i.e. EPIC or whatever has been defined in the Menu > Settings > Setup > Communications menu.	Yes
:	Separator. The separator separates all elements of the message between the source ID and the checksum. The last character of the message before the checksum shall be a separator. If a separator is to be used in a parameter or value it needs to be escaped with a backslash. For instance, to have colons in a time of day string use 11\:05\:39 for 11:05:39.	Yes



Command	<p>One of the possible commands a message can transport. Always one character long. At the moment the following commands are defined:</p> <ul style="list-style-type: none"> G – Get parameter value H – Get parameter list values S – Set parameter value U – Set relative parameter value V – Set list relative parameter values C – Current parameter value D – Current parameter list values <p><i>This document will use GET, SET, SET_RELATIVE, SET_LIST_RELATIVE, CURRENT, GET_LIST, and CURRENT_LIST to represent the command in text, but command string must use the single character.</i></p> <p>The current Value returned when changing a value via SET might not necessarily be the same as the set value. For example, setting a value might be answered with a reply of:</p> <ul style="list-style-type: none"> The same value (meaning it got set correctly) A different value (meaning some limit might have been enforced) No response (meaning it was not changed for whatever reason, possibly an error condition occurred) 	Yes
ParamID	Parameter identifier. Up to eight (8) Alphanumeric characters.	Yes
Value	<p>Value of the parameter. Represented as string. See 'Data Types' for a description of supported data types. For camera firmware versions prior to 4.0.x a value is required, even for parameters that do not require a value (such for a GET command). Use 0 in that case. In firmware versions 4.0.x and later, the inclusion of a value when not needed will be ignored. The application code should be tolerant of receiving more values than expected.</p> <p>Multiple parameter values are separated with a colon.</p>	No
*	Checksum Start Marker. If any characters follow the * it shall be the 8-bit checksum for all preceding characters (XOR of all character bytes) as a hex code in ASCII. The checksum shall be calculated over all the characters between the '#' and the '*'. Use of the checksum is optional. If not used, the '**' should be omitted.	No
<LF>	Terminator. The command shall be terminated with a <LF> (and an optional <CR>)	Yes

DATA TYPES

All data values are transmitted as strings, but they can represent different types of data. Below are descriptions of those different types.

NUMERIC (MSG_INT AND MSG_UINT)

Numeric data values can be of two broad types. The first is basic numbers in decimal format such as the ISO value being 800. The second type is based on predefined sets of enumerated values. In this case the number represents a selection that is inherently not a number such as the record state. When an enumerated value is expected for a parameter, the tables below specify which type is to be used. Key codes and format setting are a hybrid version of this type. They are numbers generated by combining several enumerated values. Details on how to create them are given in their respective sections. RED supplies a C header file (rcp_types_public.h), which lists the legal values for enumerated types. The header file is included in the REDLINK Code and Reference App Package. When creating an application to generate RCP commands, it is



highly recommended to use this header file, and use the enumerated symbol names to get the numeric values for the message parameters. Using the enumerated symbols will eliminate sensitivity to any future changes RED may make to specific values or re-ordering of the lists.

STRING (MSG_STR_PTR)

When the parameter value is a string, simply put the string contents without any quotation marks in the RCP command string as is. Many of the string type parameters that have only a GET and CURRENT command ability are simply used as data for the UI and are not meant to be parsed, but instead are simply for display. Therefore the list of possible values is not elaborated in many cases.

LIST (MSG_LIST_PTR)

String lists are used to transmit lists through the RCP protocol. To pass them as RCP data the list is put into one long value string. The cList class facilitates this. The separator for fields inside the lists is '|'. The first field in a list is a type indicator for the format of the list. The cList class in the RCP SDK should be used to parse the various cList types. Whereas the GET and CURRENT messages are used to access a parameter's current value, the GET_LIST and CURRENT_LIST are used to access a parameters possible values and the way in which they should be presented.

It is important that lists are used to get possible values, because they can change dynamically for various reasons, and can have different choices for different model cameras. In this way, the camera can present the appropriate list in any given condition.

There are two types of general lists supported and used by the D (CURRENT_LIST) message types. They are referred to as STRING_TYPE_NORMAL and STRING_TYPE_MIN_MAX_STEP. Normal is mainly for menu items such as ISO, FPS, aperture that are lists of selections in uneven steps, or just lists of enumerated values. The min_max_step type is used for controls like most of the look settings such as gains, Tint, and FLUT, which are for continuous ranges of values between two endpoints. Their formats are defined below.

STRING_TYPE_NORMAL

This list type contains the list of num/string pairs and an index indicating the currently active selection. The content of the list is a string with the fields separated by the '|' character. The syntax of the list is as below:

TYPE | CURRENT_INDEX | [NUM | STRING |]*

- ▶ **TYPE:** this is the type of list equal to 0 in this case.
- ▶ **CURRENT_INDEX:** 0 based index into the list of num/string pairs of the currently selected option.
- ▶ **NUM | STRING |:** one or more pairs of a numeric value and a decorated string. Note: the braces and asterisk are not part of the string. This is syntax notation for 'one or more occurrences'. Also note the trailing '|' in each pair so the overall string ends with a '|'.
 - **NUM:** a numeric value that is used within the camera to represent the selection. This is what should be sent in a SET command if changing the value.
 - **STRING:** a decorated string that is what should be displayed to the user. Do not convert the NUM portion to a string or attempt other conversion on it. Use the STRING. See Decorated String type for explanation of replacement characters.



STRING_TYPE_MIN_MAX_STEP

This list type contains the information to present a slider type of control where the possible values are spaced evenly over a range of values with some step size and precision. The content of the list is a string with the fields separated by the ‘|’ character. The data item represented in the control is often naturally a floating-point number. The data elements in the list are all integers and allow scaling for fixed-point operation. The syntax of the list is as below:

TYPE | CURRENT_INDEX | MIN | MAX | STEP | DIVIDER | PRECISION | PREFIX | POSTFIX |

- ▶ **TYPE:** the type of list equal to 1 in this case.
- ▶ **CURRENT_INDEX:** 0 based index into a virtual list represented by created by spacing values from MIN to MAX at STEP intervals. Its range of values would be (MAX – MIN)/STEP.
- ▶ **MIN:** the minimum natural value of the control multiplied by DIVIDER.
- ▶ **MAX:** the maximum natural value of the control multiplied by the DIVIDER.
- ▶ **STEP:** an increment between list entries. With a DIVIDER of 1000 and STEP of 100, the entries would represent steps of 0.1 in the natural data range.
- ▶ **DIVIDER:** a scale factor to convert from the list’s integer values to the natural data range.
- ▶ **PRECISION:** the smallest natural value increment. With a DIVIDER of 1000 and PRECISION of 1, the natural value is represented to 0.001.
- ▶ **PREFIX:** a decorated string to be shown in front of the value. In most cases it is empty.
- ▶ **POSTFIX:** a decorated string to be shown behind the value such as a small ‘k’ for color temperature. In most cases its empty.

An example using the FLUT parameter is below. Its natural range is -8.0 to +8.0, the selected index is the 160th being the last or 8.0 and can be specified in steps of 0.001. There are no prefix or postfix strings.

`#$SCARLET:D:FLUT:1|160|-8000|8000|100|1000|1|||:`

DECORATED STRING TYPE





Decorated strings are a way to indicate where special fonts or symbols should be used to mimic how the camera displays some information. These use HTML style special characters in place of custom symbols. For example the superscripted “1/” used in the exposure time display is represented by a “&red1over;”. So the decorated display string for 1/48 of a second would be “&red1over;48&redsec;”. Decorator characters can occur anywhere in the decorated string (not just be a prefix or suffix).

If the application platform can render the special symbols, the mapping is shown in the table below. The application will need to provide the parsing of the decorated string and perform the symbol replacement.

DECORATED STRING MAPPING

DECORATED STRING MAPPING		
DECORATOR MARKUP	DISPLAY SYMBOL	DESCRIPTION
&red1over;	1/	Small superscripted 1/
&redfover;	f/	Small superscripted f/
&redsec;	sec	Small lowercase sec
&rediso;	ISO	Small uppercase ISO



&redkelvin;	K	Subscripted uppercase K
°	°	Degree symbol
&redfps;	FPS	Small uppercase FPS
&redana2;		White ANA above black 2 in white box
&redana13;		White ANA above black 1.3 in white box
&redformatk;	K	Subscripted uppercase K
&redae;		AE icon
&redav		AV icon
&	&	Ampersand
&redcheck;	✓	Check mark
©	©	Copyright symbol
®	®	Registered symbol
™	™	Trademark symbol

SPECIAL AGGREGATE TYPE

A few parameters take and/or return a collection of data, possibly of different types. In these cases, C structure types define the collections. These are also provided in the C header file that RED supplies. The data is still represented as readable strings in the message, but are provided as a colon separated list. This applies to the value types MSG_AUDIOVU_DATA, MSG_CLIPINFO_PTR, MSG_EXTENDED_CLIPINFO_PTR, and MSG_CAMINFO_PTR.

RELATIVE PARAMETER UPDATING

Starting with RCP parameter set 6.1 there are now commands to do SET_RELATIVE and SET_LIST_RELATIVE on select parameters. These are coded as commands U and V respectively.

SET_RELATIVE takes the value field as an adjustment to be added to the parameter’s current value. This permits adjusting the value up or down without knowing its value before hand. This is intended for control interface models where the control might be a knob that has no absolute position, on relative adjustments from a current value. Making a relative update allows not having to do a GET on the parameter first.

SET_LIST_RELATIVE allows adjusting the current index within a list. The value passed for this command is an increment to the list index, either positive or negative. The parameter will be set to the corresponding list value that many entries away from the current value. For example, the ISO could be changed to the next higher setting by sending a SET_LIST_RELATIVE on the parameter ISO with a value of +1.

USAGE

The RCP command exchange works in an asynchronous manner since it allows for multiple agents to be sending SET or GET commands that may be asking for or changing the same parameter. When a parameter is changed, a CURRENT command is sent out to all agents to indicate the new value. If a SET command is sent that does not change the parameter value, no CURRENT will be generated. For this reason the CURRENT should not be used as an indication of acknowledgement of the SET command. The GET command can always be used to retrieve the current value of a parameter.



SET or GET commands that are invalid, either bad parameter or value, are simply ignored. The assumption is that commands are not being generated by a person typing, but rather are part of an already debugged application program. Therefore parameter names are not being created dynamically, only values might be. An out of range value ends up being ignored and therefore no CURRENT is generated because no value changes. If you are implementing some kind of GUI as part of your application, use the CURRENT value to set your display values rather than the value you SET. This will take into account any limiting or adjustments the camera may make to your requested value.

Not all parameters support all commands. Use the tables in each parameter group section that follows for individual parameter details.

VOLUME OF TRAFFIC

Normal operation of the camera can result in a lot of output traffic of CURRENT commands. They can be generated by actions within the camera that are unrelated to SET or GET commands. Since they are essentially broadcast messages, an application listening on the CTRL or GigE port would see everything and possibly flood the bandwidth of a 3rd party application. Be sure to select a processor in your application that can handle continuous UART traffic at the 115,200 baud speed.

SOURCE CODE PACKAGE AVAILABILITY

RED supplies a source code package that includes the header file for types and libraries for creating and parsing RCP message packets. Download REDLINK SDK Source and Reference Applications, to get this file and additional documentation. Included in that download are example applications for iOS, Android™, Windows®, and OS X® platforms. These applications demonstrate connecting to a camera via Ethernet and serial port and some basic functionality. Source code is provided as well as executable files for PC versions. This can be used as an example, or the foundation for a new application.

DEFINED PARAMETER GROUPS

SYSINFO GROUP

SYSINFO data is used to find the configuration of the camera. The S, G, and C columns indicate which commands are supported for a given parameter.

SYSINFO GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
CAMFWVER	MSG_STR_PTR		G	C	Camera firmware version as n.n.n
CAMVCREV	MSG_STR_PTR		G	C	Camera firmware SVN revision. Branch name and revision number.
CAMID	MSG_STR_PTR	S	G	C	RCP camera ID as a string. This is the same name as used in the source field in an RCP message. Up to 8 characters.
CAMINFO	MSG_CAMINFO_PTR		G	C	Camera serial number, type name, firmware version, IP address, and interface. As of RCP parameter set version 6, the interface is added and the IP address returns only an empty string. (see rcp_interface_t for interface types)



CAMNAME	MSG_STR_PTR		G	C	Camera type name as defined at factory: EPIC-M MONOCHROME, EPIC-M, EPIC-X MONOCHROME EPIC-X, SCARLET-X EPIC-M DRAGON, EPIC-X DRAGON, SCARLET-X DRAGON
CAMSER	MSG_STR_PTR		G	C	Camera serial number. 11 characters.
RCPVER	MSG_INT		G	C	RCP version.
RCPPSVER	MSG_INT		G	C	RCP parameter set version. 32 bits representing 16 bit major and minor versions as major.minor
TARGETID	MSG_STR_PTR	S	G	C	Custom target ID as a string. Used for filtering camera to camera synchronization messages along with the TARGET parameter.
RUNTIME	MSG_INT		G	C	Camera runtime in seconds
GROUPID	MSG_STR_PTR	S	G	C	RCP camera group ID as a string.

EXAMPLES OF SYSINFO USAGE

Send request for RCP parameter set version.

▶ #`$EXT:G:RCPPSVER:`

Which returns:

▶ #`$EPIC:C:RCPPSVER:196608:`

RCP parameter set version = 196608 = 0x00030000 = meaning version 3.00

Send request for camera info.

▶ #`$EXT:G:CAMINFO:`

Which returns:

▶ #`$EPIC:C:CAMINFO:102-7E3-8D5:EPIC-X:4.0.18::2:` (2= Brain gigabit Ethernet)

USING CAMINFO TO DISCOVER CAMERAS

There is a process for discovering cameras on the network if the IP address is unknown. To do this, broadcast a UDP datagram to port 1112 repeatedly with a small delay. For instance, 5 times with a delay between each of 500 milliseconds. This assumes that there are multiple cameras on the same network and they may collide when responding. This part of the process may be tailored based on your knowledge of your own configuration. The datagram should be the command to get camera information:

#`$EXT:G:CAMINFO:`. This will cause any camera on the network to respond with a CURRENT command with the camera PIN, camera name, firmware version, null string IP address, and the interface the response was generated from. A camera with a REDLINK Bridge in infrastructure mode will send a CAMINFO response for both the wireless on the Bridge and the GigE Ethernet port. A list can be built of all unique responses and then that list can be searched for the desired camera. The IP addresses of the source ports can be extracted from the UDP response header. This is a change to the discovery process beginning with RCP version 6. The interface name is used to distinguish the physical port. A connection can then be made to the desired camera and interface using TCP on port 1111 for further communication. Example source code for this process can be found in the Core Example of the REDLINK SDK.



KEYS GROUP

Generic key codes are the value entry for the KEYCODE parameter. They are 32 bit values. Values are shown by descriptive name and code in the KEY CODES table. Each name starts with the source device, then key, then event. The codes are grouped in complimentary pairs of press/release, clockwise/counter-clockwise groupings. For keys that have a press action, it is good practice to always send the complimentary release action afterwards. Not all press key events have a release.

Key sources are BRAIN, SH=Sidehandle, S-SSD= Side SSD, RM=REDmote, LCD, EVF, Pro I/O, UI=touchscreen soft key, TH=Top Handle. A full list of sources and keys is in keydefinitions.h along with a macro to build the keycode from the source, flags, and id components.

By using the key mapping and presets feature of the camera to strategically define keys for frequent actions, many camera operation functions can be controlled with just this one command.

KEYS GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
KEYCODE	MSG_INT	S			Send a generic key code
KEYACT	MSG_STR_PTR	S			Send a key action code, possibly with a argument as comma separated list (see key_action_t)
SELISO	MSG_INT	S			Deprecated as of RCP parameter set 6.1. Use SET_LIST_RELATIVE on ISO parameter instead.
SELSH	MSG_INT	S			Deprecated as of RCP parameter set 6.1. Use SET_LIST_RELATIVE on SHTIME parameter instead.

COMMON KEY CODES

KEY EVENT NAME	CODE	COMPLIMENT KEY EVENT NAME	CODE
BRAIN GPI In High	261	BRAIN GPI In Low	517
BRAIN Record Half Press	271	BRAIN Record Half Release	527
BRAIN Record Full Press	272	BRAIN Record Full Release	528
SH User A Press	65793	SH User A Release	66049
SH User B Press	65794	SH User B Release	66050
SH User C Press	65795	SH User C Release	66051
SH User D Press	65796	SH User D Release	66052
SH User 1 Press	65797	SH User 1 Release	66053
SH User 2 Press	65798	SH User 2 Release	66054
SH User 3 Press	65799	SH User 3 Release	66055
SH User 4 Press	65800	SH User 4 Release	66056
SH User 5 Press	65801	SH User 5 Release	66057
SH User 6 Press	65802	SH User 6 Release	66058
SH User 7 Press	65803	SH User 7 Release	66059
SH User 1+4 Press	65804	SH User 1+4 Release	66060
SH Still Select	65805	SH Movie Select	65806
SH Record Half Press	65807	SH Record Half Release	66063



SH Record Full Press	65808	SH Record Full Release	66064
SH Backlight Press	65821	SH Backlight Release	66077
SH Nav Menu Press	65822	SH Nav Menu Release	66078
SH Nav North Press	65823	SH Nav North Release	66079
SH Nav South Press	65824	SH Nav South Release	66080
SH Nav East Press	65825	SH Nav East Release	66081
SH Nav West Press	65826	SH Nav West Release	66082
SH Nav Enter Press	65827	SH Nav Enter Release	66083
SH Rocker + Press	65834	SH Rocker + Release	66090
SH Rocker - Press	65835	SH Rocker - Release	66091
SH Rotary Front CW	66586	SH Rotary Front CCW	67610
SH Rotary Jog CW	66587	SH Rotary Jog CCW	67611
S-SSD User 1 Press	196869	S-SSD User 1 Release	197125
S-SSD User 2 Press	196870	S-SSD User 2 Release	197126
S-SSD User 1+2 Press	196871	S-SSD User 1+2 Release	197127
S-SSD Record Half Press	196879	S-SSD Record Half Release	197140
S-SSD Record Full Press	196880	S-SSD Record Full Release	197141
RM User A Press	262401	RM User A Release	262657
RM User B Press	262402	RM User B Release	262658
RM User C Press	262403	RM User C Release	262659
RM User D Press	262404	RM User D Release	262660
RM Still Select	262413	RM Movie Select	262414
RM Record Half Press	262415	RM Record Half Release	262671
RM Record Full Press	262416	RM Record Full Release	262672
RM Nav Menu Press	262430	RM Nav Menu Release	262686
RM Nav North Press	262431	RM Nav North Release	262687
RM Nav South Press	262432	RM Nav South Release	262688
RM Nav East Press	262433	RM Nav East Release	262689
RM Nav West Press	262434	RM Nav West Release	262690
RM Nav Enter Press	262435	RM Nav Enter Release	262691
RM Rocker + Press	262442	RM Rocker + Release	262698
RM Rocker - Press	262443	RM Rocker - Release	262699
RM Rotary CW	263194	RM Rotary CCW	264218
LCD (BRAIN) User 1 Press	327941	LCD (BRAIN) User 1 Release	328197
LCD (BRAIN) User 2 Press	327942	LCD (BRAIN) User 2 Release	328198
LCD (BRAIN) Up Press	327967	LCD (BRAIN) Up Release	328223
LCD (BRAIN) Down Press	327968	LCD (BRAIN) Down Release	328224
LCD (BRAIN) User 1+2 Press	327943	LCD (BRAIN) User 1+2 Release	328199
LCD (Rear) User 1 Press	459013	LCD (Rear) User 1 Release	459269
LCD (Rear) User 2 Press	459014	LCD (Rear) User 2 Release	459270
LCD (Rear) Up Press	459039	LCD (Rear) Up Release	459295



LCD (Rear) Down Press	459040	LCD (Rear) Down Release	459296
LCD (Rear) User 1+2 Press	459015	LCD (Rear) User 1+2 Release	459271
Lens VTR (run/stop) Press	393488	Lens VTR (run/stop) Release	393744
EVF (BRAIN) User 1 Press	524549	EVF (BRAIN) User 1 Release	524805
EVF (BRAIN) User 2 Press	524550	EVF (BRAIN) User 2 Release	524806
EVF (Rear) User 1 Press	590085	EVF (Rear) User 1 Release	590341
EVF (Rear) User 2 Press	590086	EVF (Rear) User 2 Release	590342
Pro I/O GPI A High	655873	Pro I/O GPI A Low	655617
Pro I/O GPI B High	655874	Pro I/O GPI B Low	655618
Pro I/O SW 1 High	655875	Pro I/O SW 1 Low	655619
Pro I/O SW 2 High	655876	Pro I/O SW 2 Low	655620
UI Softkey A Press	721153	n/a	n/a
UI Softkey B Press	721154	n/a	n/a
UI Softkey C Press	721155	n/a	n/a
UI Softkey D Press	721156	n/a	n/a
UI Softkey 1 Press	721157	n/a	n/a
UI Softkey 2 Press	721158	n/a	n/a
UI Softkey 3 Press	721159	n/a	n/a
UI Softkey 4 Press	721160	n/a	n/a

EXAMPLES OF KEYCODE USAGE

Send key code for Side Handle, user key D, press, followed by release.

- ▶ # \$EXT:S:KEYCODE:65796:
- ▶ # \$EXT:S:KEYCODE:66052:

Move the menu cursor on UI one spot to the right, by mimicking one click of the Side Handle front rotary wheel in the clockwise direction.

- ▶ # \$EXT:S:KEYCODE:66586:

EXAMPLES OF KEYACT USAGE

The following example illustrates applying a preset using a key action. Assume there is a preset loaded on the camera called ISO10000. By looking in the key_action_t enum definition, the action KEY_ACTION_APPLY_PRESET can be found and its value of 69 discovered. Most key actions do not take a value beyond the action code, but a few do. When needed the second value is provided after the action code, separated by a comma. The proper process is not hard code the action codes, but rather use the enum and convert it to a string with sprintf() or something similar. The message below will apply the preset file ISO10000.

- ▶ # \$EXT:S:KEYACT:69,ISO10000:

EXPOSURE GROUP

The EXPOSURE group allows control of shutter angle, exposure time and HDRx mode.



EXPOSURE GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
HDRFACT	MSG_INT	S	G/H	C/D	HDR factors are = 2 raised to the power of factor, where factor is 1,2,3,4,5,6.
HDRMODE	MSG_INT	S	G/H	C/D	HDR mode off or HDRx (see hdr_mode_t)
MMSHCOLR	MSG_UINT		G	C	Indicates if motion mount shutter angle/time has been clipped and should be displayed in a different color. 1=clipped, 0=not
MMSHTIME	MSG_UINT	S	G	C	Motion mount exposure time denominator *100,000
MMSHANGL	MSG_UINT	S	G	C	Motion mount shutter angle in degrees *10
SHANGLE	MSG_INT		G/H	C/D	Shutter angle in degrees *1000, range of 1 to 360,000
SHANGLT	MSG_INT	S	G	C	Shutter angle target degrees * 1000
SHDISP	MSG_INT	S	G/H	C/D	Shutter display in time or angle (see shutter_display_t)
SHMODE	MSG_INT	S	G/H	C/D	Current shutter mode (see shutter_mode_t)
SHTIME	MSG_INT	V	G/H	C/D	Exposure time denominator * 1000, 1/48 second = 48000
SHTIMET	MSG_INT	S	G	C	Exposure time target denominator*1000. i.e. 48000 = 1/48

USING TARGET SETTINGS

There are four parameters that should be set using target values rather than direct settings. In the EXPOSURE group these are shutter angle (SHANGLT) and exposure time (SHTIMET). In the PROJECT group there is the REDCODE setting (RCTARGET). In the lens group there is aperture setting (APRTRT). The reason for using these rather than a direct value is that the system can automatically constrain them. For example, REDCODE setting is a function of REDCODE target, frame rate, image format and media speed. A low REDCODE value may be achievable at 24 fps. Setting the target to that value will cause a CURRENT for the REDCODE parameter to be emitted that matches the requested target. This indicates the target could be reached. If the target cannot be reached, the emitted CURRENT value will be different than the target. This is what the REDCODE is actually set to. If the frame rate is then moved to a higher value, the REDCODE may get limited. This will change the REDCODE setting (reflected in new CURRENT) but not the target. If the frame rate is then lowered back to 24, the REDCODE will automatically be returned to the previous target value, with a new CURRENT being emitted. Similarly, the exposure time could be limited when changing frame rates and use of the Motion Mount can impact shutter angles.

Use the target parameters (SHANGLT, SHTIMET, RCTARGET, and APRTRT) to do the SET operations, but use the CURRENT values from the corresponding absolute values (SHANGLE, SHTIME, REDCODE, APRTR) to display in a UI or process for other reasons, as these are the actual values.

EXAMPLE OF TURNING ON HDRX

Set the HDRx factor and then turn on HDRx.

- ▶ #EXT:S:HDRFACT:2:
- ▶ #EXT:S:HDRMODE:1: (1=HDR_MODE_HDRX in hdr_mode_t)

IMAGE GROUP

The image group provides control over the look qualities of the image. Items such as color space, color temperature, gamma space, ISO, Tint, and toggling raw mode on or off.



IMAGE GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
COLTMP	MSG_INT	S/U/ V	G/H	C/D	Color temperature in the range 1700 to 100000 Kelvin. Not available in Monochrome cameras.
COLTMPP	MSG_INT	S	H	D	List of pre-set color temperature values. Not available in Monochrome cameras.
CSPACE	MSG_INT	S	G/H	C/D	Current color space (see colorspace_t)
GSPACE	MSG_INT	S	G/H	C/D	Current gamma space (see gammaspace_t)
ISO	MSG_INT	S/V	G/H	C/D	ISO Values are = 250,320,400,500,640,800,1000,1280,1600,2000,2500,3200, 4000,5000,6400,12800
RAWMODE	MSG_INT	S	G	C	0 = Raw Mode off, 1 = Raw Mode On
RCURVE	MSG_INT	S	G	C	Red Curve, comma separated string of black x, y, toe x, y, center x, y, knee x, y, white x, y. X is the input to lookup and Y is the output of lookup. Both are percentages in the range 0 to 1.00, scaled by 1000. See example for more details.
RCURVESX	MSG_INT	S/U	G	C	Red curve shadow (black) x coordinate. Percentage in the range 0 to 1.00, scaled by 1000. Same units as in RCURVE.
RCURVESY	MSG_INT	S/U	G	C	Red curve shadow (black) y coordinate. Percentage in the range 0 to 1.00, scaled by 1000.
RCURVEDX	MSG_INT	S/U	G	C	Red curve dark (toe) x coordinate.
RCURVEDY	MSG_INT	S/U	G	C	Red curve dark (toe) y coordinate.
RCURVEMX	MSG_INT	S/U	G	C	Red curve midtone (center) x coordinate.
RCURVEMY	MSG_INT	S/U	G	C	Red curve midtone (center) y coordinate.
RCURVELX	MSG_INT	S/U	G	C	Red curve light (knee) x coordinate.
RCURVELY	MSG_INT	S/U	G	C	Red curve light (knee) y coordinate.
RCURVEHX	MSG_INT	S/U	G	C	Red curve highlight (white) x coordinate.
RCURVEHY	MSG_INT	S/U	G	C	Red curve highlight (white) y coordinate.
GCURVE	MSG_INT	S	G	C	Green Curve, comma separated string black x, y, toe x, y, center x, y, knee x, y, white x, y
GCURVESX	MSG_INT	S/U	G	C	Green curve shadow (black) x coordinate.
GCURVESY	MSG_INT	S/U	G	C	Green curve shadow (black) y coordinate.
GCURVEDX	MSG_INT	S/U	G	C	Green curve dark (toe) x coordinate.
GCURVEDY	MSG_INT	S/U	G	C	Green curve dark (toe) y coordinate.
GCURVEMX	MSG_INT	S/U	G	C	Green curve midtone (center) x coordinate.
GCURVEMY	MSG_INT	S/U	G	C	Green curve midtone (center) y coordinate.
GCURVELX	MSG_INT	S/U	G	C	Green curve light (knee) x coordinate.
GCURVELY	MSG_INT	S/U	G	C	Green curve light (knee) y coordinate.
GCURVEHX	MSG_INT	S/U	G	C	Green curve highlight (white) x coordinate.
GCURVEHY	MSG_INT	S/U	G	C	Green curve highlight (white) y coordinate.
BCURVE	MSG_INT	S	G	C	Blue Curve, comma separated string black x, y, toe x, y, center x, y, knee x, y, white x, y
BCURVESX	MSG_INT	S/U	G	C	Blue curve shadow (black) x coordinate.
BCURVESY	MSG_INT	S/U	G	C	Blue curve shadow (black) y coordinate.



BCURVEDX	MSG_INT	S/U	G	C	Blue curve dark (toe) x coordinate.
BCURVEDY	MSG_INT	S/U	G	C	Blue curve dark (toe) y coordinate.
BCURVEMX	MSG_INT	S/U	G	C	Blue curve midtone (center) x coordinate.
BCURVEMY	MSG_INT	S/U	G	C	Blue curve midtone (center) y coordinate.
BCURVELX	MSG_INT	S/U	G	C	Blue curve light (knee) x coordinate.
BCURVELY	MSG_INT	S/U	G	C	Blue curve light (knee) y coordinate.
BCURVEHX	MSG_INT	S/U	G	C	Blue curve highlight (white) x coordinate.
BCURVEHY	MSG_INT	S/U	G	C	Blue curve highlight (white) y coordinate.
LCURVE	MSG_INT	S	G	C	Luma Curve, comma separated string black x, y, toe x, y, center x, y, knee x, y, white x, y
LCURVESX	MSG_INT	S/U	G	C	Luma curve shadow (black) x coordinate.
LCURVESY	MSG_INT	S/U	G	C	Luma curve shadow (black) y coordinate.
LCURVEDX	MSG_INT	S/U	G	C	Luma curve dark (toe) x coordinate.
LCURVEDY	MSG_INT	S/U	G	C	Luma curve dark (toe) y coordinate.
LCURVEMX	MSG_INT	S/U	G	C	Luma curve midtone (center) x coordinate.
LCURVEMY	MSG_INT	S/U	G	C	Luma curve midtone (center) y coordinate.
LCURVELX	MSG_INT	S/U	G	C	Luma curve light (knee) x coordinate.
LCURVELY	MSG_INT	S/U	G	C	Luma curve light (knee) y coordinate.
LCURVEHX	MSG_INT	S/U	G	C	Luma curve highlight (white) x coordinate.
LCURVEHY	MSG_INT	S/U	G	C	Luma curve highlight (white) y coordinate.
REDG	MSG_INT	S/U	G/H	C/D	Red linear gain * 1000, where gain is in the range 0.000 to 10.000. 3 decimal places of precision allowed.
BLUEG	MSG_INT	S/U	G/H	C/D	Green linear gain * 1000, where gain is in the range 0.000 to 10.000. 3 decimal places of precision allowed.
GREENG	MSG_INT	S/U	G/H	C/D	Blue linear gain * 1000, where gain is in the range 0.000 to 10.000. 3 decimal places of precision allowed.
HDRXMONX	MSG_INT	S	G/H	C/D	Determines which HDR track to view (see <code>hdr_track_t</code>)
HDMIMUX	MSG_INT	S	G/H	C/D	Select between EVF and rear HDMI® on WEAPON (see <code>hdm_mux_t</code>)
SATURAT	MSG_INT	S/U	G/H	C/D	Saturation * 1000, where saturation is in the range 0.00 to 4.00 Not available in Monochrome cameras.
CONTRST	MSG_INT	S	G/H	C/D	Contrast * 1000, where contrast is in the range -1.00 to +1.00
BRIGHT	MSG_INT	S	G/H	C/D	Brightness * 1000, where brightness is in the range -10.00 to +10.00
EXPCOMP	MSG_INT	S	G/H	C/D	Exposure compensation * 1000, where exposure comp is in the range -7.00 to +7.00
FLUT	MSG_INT	S/U	G/H	C/D	FLUT * 1000, where FLUT is in the range -8.000 to +8.000, so 3 decimal places of precision allowed.
CMATBB	MSG_INT	S/U	G	C	Custom matrix Blue row, Blue component entry * 1000, where entry is in range of -2.000 to 2.000. 3 decimal places of precision allowed.
CMATBG	MSG_INT	S/U	G	C	Custom matrix Blue row, Green component entry * 1000.
CMATBR	MSG_INT	S/U	G	C	Custom matrix Blue row, Red component entry * 1000.
CMATGB	MSG_INT	S/U	G	C	Custom matrix Green row, Blue component entry * 1000.



CMATGG	MSG_INT	S/U	G	C	Custom matrix Green row, Green component entry * 1000.
CMATGR	MSG_INT	S/U	G	C	Custom matrix Green row, Red component entry * 1000.
CMATRB	MSG_INT	S/U	G	C	Custom matrix Red row, Blue component entry * 1000.
CMATRG	MSG_INT	S/U	G	C	Custom matrix Red row, Green component entry * 1000.
CMATRR	MSG_INT	S/U	G	C	Custom matrix Red row, Red component entry * 1000.
FRSHARP	MSG_INT	S	G/H	C/D	Sharpening value in the range of 0 to 100
SHADOW	MSG_INT	S/U	G/H	C/D	Shadow * 1000, where shadow is in the range -2.000 to +2.000, so 3 decimal places of precision allowed.
TINT	MSG_INT	S/U	G/H	C/D	Tint * 1000, where Tint is in the range -100.000 to +100.000, so 3 decimal places of precision allowed. Not available in Monochrome cameras.
ACES	MSG_INT	S	G	C	Enable/Disable lift/gamma/gain (LGG) processing. 0=disable, 1=enable
BGAIN	MSG_INT	S/U	G/H	C/D	LGG blue gain * 1000, where gain is in the range 0.000 to 2.000
BGAMMA	MSG_INT	S/U	G/H	C/D	LGG blue gamma * 1000, where gamma is in the range 0.000 to 4.000
BLIFT	MSG_INT	S/U	G/H	C/D	LGG blue lift * 1000, where gain is in the range -1.000 to +1.000
GGAIN	MSG_INT	S/U	G/H	C/D	LGG green gain * 1000, where gain is in the range 0.000 to 2.000
GGAMMA	MSG_INT	S/U	G/H	C/D	LGG green gamma * 1000, where gamma is in the range 0.000 to 4.000
GLIFT	MSG_INT	S/U	G/H	C/D	LGG green lift * 1000, where gain is in the range -1.000 to +1.000
RGAIN	MSG_INT	S/U	G/H	C/D	LGG red gain * 1000, where gain is in the range 0.000 to 2.000
RGAMMA	MSG_INT	S/U	G/H	C/D	LGG red gamma * 1000, where gamma is in the range 0.000 to 4.000
RLIFT	MSG_INT	S/U	G/H	C/D	LGG red lift * 1000, where gain is in the range -1.000 to +1.000

EXAMPLE OF USING CURVE PARAMETER

Curves are defined by five (5) pairs of numbers. Each pair is an x and y coordinate defining a point on a Bezier path that transforms an in value to a new out value applied to all pixels in the image. The five points are known as Shadow, Dark, Midtone, Light and Highlight. By default they are set to define a straight-line function that does not change the data. The values range from 0 to 1.000 scaled by 1000. 1.000 represents the full range of the video. The ten values are given as a comma separated list. The x values must conform to the following restriction: Shadow <= Dark <= Midtone <= Light <= Highlight.

The curves can be set using the all-in-one parameters RCURVE, GCURVE, BCURVE and LCURVE, or by accessing the individual coordinates using the parameters cCURVEpa, where 'c' is the color (R,G,B,L), 'p' is the point (S,D,M,L,H), and 'a' is the axis (X,Y). The individual point control parameters are convenient when the curve is being manipulated from a GUI where typically only one point is being pushed around at any time. The individual point access has the added benefit in that they support the relative update method. This is convenient when they are controlled by a knob or wheel type interface and absolute values are not needed. This saves retrieving the current value before setting a new one.



The command to set the blue curve to default would look like:

▶ #`$EXT:S:BCURVE:0,0,250,250,500,500,750,750,1000,1000:`

The command to lift the midtone of the green curve by 10 units from its current value would be:

▶ #`$EXT:U:GCVVMY:10:`

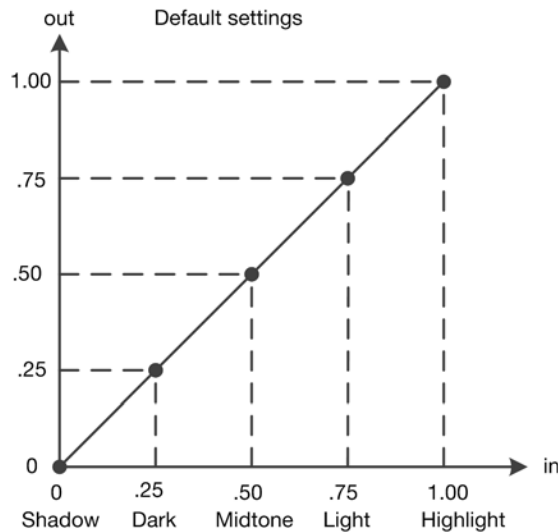


Figure 5. Default Curve Settings

LENS GROUP

The lens group allows control and query of the aperture, focal length and focal distance of electronic lenses that support such features. Notice that F#, Fl, and Fd are all scaled by 10 or 100 to allow representation of fractional part as a whole number. Divide the received number by the scale factor to get the value reported by the lens.

LENS GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
APERCTRL	MSG_INT		G	C	Is aperture controllable through UI? This also indicates whether aperture can be controlled by RCP. 0=no, 1=yes
APRTR	MSG_INT	S/V	G/H	C/D	Aperture F# *10. Use of APRTR for SET command is deprecated as of Parameter set 6.1.
APRTRT	MSG_INT	S	G	C	Aperture target value * 10.
LDISPM	MSG_INT	S	G/H	C/D	List aperture in 1/4 stops or 1/3 stops. (see aperture_list_t)
FLENGTH	MSG_INT		G/H	C/D	Focal length in mm *100
FOCSCTRL	MSG_INT		G	C	Is focus controllable through UI? This also indicates whether aperture can be controlled by RCP. 0=no, 1 = yes
FOCUS	MSG_INT	S/V	G/H	D	Focal distance in meters *100. A GET on this causes FOCUSF and FOCUSN to be sent out.
FOCUSF	MSG_INT		G	C	Far focal distance in meters*100
FOCUSN	MSG_INT		G	C	Far focal distance in meters*100



LENSFDMD	MSG_INT	S	G/H	C/D	Focus distance display mode, metric vs imperial for UI displayed lens data (see focus_distance_mode_t)
PLDETECT					Enable/disable auto detection of PL type lens. 0=disable, 255=enable.
PLPOWER					Enable/disable power to PL type lens. 0=disable, 1=enable.
SHRCKDST	MSG_INT	S	G/H	C/D	Enable/Disable showing showing distance on rack focus points. 0=disable, 1=enable.
MMMODE	MSG_UNIT	S	G/H	C/D	Motion shutter mode (see motion_mount_shutter_mode_t)
MMMISOND	MSG_INT	S	G/H	C/D	Flag indicating whether ISO is re-interpreted (ISO pull) for motion mount ND. 0=off, 1=on.
MMNDV	MSG_INT	S	G/H	C/D	Motion mount shutter ND value. Use GET_LIST for values, units are hundredths of a stop
MMNDMODE	MSG_INT	S	G/H	C/D	Motion mount ND display mode. (see motion_mount_display_mode_t)

OVERLAY GROUP

The overlay group allows selection of the false color and overlay modes of the camera.

OVERLAY GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
ZEBRA1	MSG_INT	S	G	C	Zebra 1 enable, set to 1 to enable, 0 to disable
ZEBRA1LO	MSG_INT	S	G/H	C/D	Zebra 1 Low IRE, in the range 1 to 100
ZEBRA1HI	MSG_INT	S	G/H	C/D	Zebra 1 High IRE, in the range 1 to 100
ZEBRA2	MSG_INT	S	G	C	Zebra 2 enable, set to 1 to enable, 0 to disable
ZEBRA2LO	MSG_INT	S	G/H	C/D	Zebra 2 Low IRE, in the range 1 to 100
ZEBRA2HI	MSG_INT	S	G/H	C/D	Zebra 2 High IRE, in the range 1 to 100
F0MODE	MSG_INT	S	G/H	C/D	Frame guide overlay mode selection. (see guide_aspect_ratio_t) Get the list at run time because it is dynamically populated.
F0SCL	MSG_INT	S	G/H	C/D	Frame guide overlay scale. Percent *100, range 0 - +10000
F0OFFSX	MSG_INT	S	G/H	C/D	Frame guide overlay horizontal offset. Percent * 100, range -10000 - +10000
F0OFFSY	MSG_INT	S	G/H	C/D	Frame guide overlay vertical offset. Percent * 100, range -10000 - +10000
F0LNST	MSG_INT	S	G/H	C/D	Frame guide overlay line style. Solid, dashed or bracket. Get list at run time for values. (see guide_line_style_t)
F0COLOR	MSG_INT	S	G/H	C/D	Frame guide overlay color. Get list at runtime for values. (see frameguide_color_t)
F0OPAC	MSG_INT	S	G/H	C/D	Frame guide overlay opacity percent. Range 0 - 100
F0NUM	MSG_INT	S	G	C	Frame guide user mode aspect ratio numerator *10000
F0ABSXOF	MSG_INT	S	G	C	Frame guide absolute horizontal offset pixels
F0ABSYOF	MSG_INT	S	G	C	Frame guide absolute vertical offset pixels
F0ABSW	MSG_INT	S	G	C	Frame guide absolute width pixels
F0ABSH	MSG_INT	S	G	C	Frame guide absolute height pixels
F1MODE	MSG_INT	S	G/H	C/D	Action guide mode selection. Off, user, full, or preset aspect ratio. (see guide_aspect_ratio_t)



F1SCL	MSG_INT	S	G/H	C/D	Action guide scale. Percent *100, range 0 - +10000
F1OFFSX	MSG_INT	S	G/H	C/D	Action guide horizontal offset. Percent * 100, range -10000 - +10000
F1OFFSY	MSG_INT	S	G/H	C/D	Action guide overlay vertical offset. Percent * 100, range - 10000 - +10000
F1LNST	MSG_INT	S	G/H	C/D	Action guide line style. Solid, dashed or bracket. Get list at run time for values.
F1COLOR	MSG_INT	S	G/H	C/D	Action guide color. (see frameguide_color_t)
F1OPAC	MSG_INT	S	G/H	C/D	Action guide opacity percent. Range 0 - 100
F1NUM	MSG_INT	S	G	C	Action guide user mode aspect ratio numerator *10000
F1ABSXOF	MSG_INT	S	G	C	Action guide absolute horizontal offset pixels
F1ABSYOF	MSG_INT	S	G	C	Action guide absolute vertical offset pixels
F1ABSW	MSG_INT	S	G	C	Action guide absolute width pixels
F1ABSH	MSG_INT	S	G	C	Action guide absolute height pixels
F1RELF0	MSG_INT	S	G	C	Enable/Disable relative to frame guide property. 0=disable, 1=enable
F2MODE	MSG_INT	S	G/H	C/D	Title guide mode selection. (see guide_aspect_ratio_t)
F2SCL	MSG_INT	S	G/H	C/D	Title guide scale. Percent *100, range 0 - +10000
F2OFFSX	MSG_INT	S	G/H	C/D	Title guide horizontal offset. Percent * 100, range -10000 - +10000
F2OFFSY	MSG_INT	S	G/H	C/D	Title guide overlay vertical offset. Percent * 100, range -10000 - +10000
F2LNST	MSG_INT	S	G/H	C/D	Title guide line style. Solid, dashed or bracket. Get list at run time for values.
F2COLOR	MSG_INT	S	G/H	C/D	Title guide color. (see frameguide_color_t)
F2OPAC	MSG_INT	S	G/H	C/D	Title guide opacity percent. Range 0 - 100
F2NUM	MSG_INT	S	G	C	Title guide user mode aspect ratio numerator *10000
F2ABSXOF	MSG_INT	S	G	C	Title guide absolute horizontal offset pixels
F2ABSYOF	MSG_INT	S	G	C	Title guide absolute vertical offset pixels
F2ABSW	MSG_INT	S	G	C	Title guide absolute width pixels
F2ABSH	MSG_INT	S	G	C	Title guide absolute height pixels
F2RELF0	MSG_INT	S	G	C	Enable/Disable relative to frame guide property. 0=disable, 1=enable
CGGUIDE	MSG_INT	S	G/H	C/D	Select which guide center cross is centered in. Frame, action, title or recording area. Get list at run time for values.
CGCOLOR	MSG_INT	S	G/H	C/D	Center guide color. (see frameguide_color_t)
CGOPAC	MSG_INT	S	G/H	C/D	Center guide opacity percent. Range 0 - 100
GGGUIDE	MSG_INT	S	G/H	C/D	Select which guide the grid is placed in . Frame, action, title or recording area. Get list at run time for values
GGCOLOR	MSG_INT	S	G/H	C/D	Grid guide color. (see frameguide_color_t)
GGOPAC	MSG_INT	S	G/H	C/D	Grid guide opacity percent. Range 0 - 100
SHGUIDE	MSG_INT	S	G/H	C/D	Select which guide the look around shading is placed outside of. Frame, action, title or recording area. Get list at run time for values.



SHCOLOR	MSG_INT	S	G/H	C/D	Look-around shading color. (see framgeguide_color_t)
SHOPAC	MSG_INT	S	G/H	C/D	Look-around shading opacity percent. Range 0 - 100
TCDMODE	MSG_INT	S	G/H	C/D	Time code display mode, ToD or Edge. (see tc_display_t)
FALSEC	MSG_INT	S	G/H	C/D	False color display mode (see false_color_mode_t)
HORIZON	MSG_INT	S	G	C	Enable, disable or toggle the horizon line in UI. (see ui_enable_horizon_t)
GIOZONE	List only	T	H	D	State and label pair for each exposure zone (gioscope). State is 0 or 1. Label is a string to display on control of that zone. Use colors in GIOCPRI and GIOCSEC to color the control.
GIOCPRI	List only		H	D	State 0 colors/labels for exposure zones (Gioscope) mode. Color is 24 bit RGB value for displaying control when in state 0. Use labels in GIOZONE list.
GIOCSEC	List only		H	D	State 1 colors/labels for exposure zones (Gioscope) mode. Color is 24 bit RGB value for displaying control when in state 1. Use labels in GIOZONE list.

EXAMPLE OF CHANGING FRAME GUIDE OVERLAY

To change the frame guide overlay, you must first know the allowable values. The full list of available values is provided in an enum, `guide_aspect_ratio_t`, but it is best to dynamically fetch the list. The current list can be fetched using the `F0MODE` parameter with an `H` command, which will return currently active setting and the list of values and strings of all valid choices.

First do a `GET_LIST` on `F0MODE`.

▶ `#$EXT:H:F0MODEL:`

The response will be something like below:

▶ `#$EPIC:D:F0MODE:0|3|-1|Off|0|Full|1333333|4:3|1777778|16:9|1850000|1.85:1|1896296|1.9:1|2400000|2.4:1|-2|User|-3|Absolute|:`

The list values start with a 0, which is the list type.

Next is the currently active setting = 3. This is a zero-based index to select the value/string pair. In this case the fourth pair is `1777778|16:9`. The `1777778` is the value that would be used in a `SET` or returned in a `GET` for the `16:9` aspect frame guide `F0MODE` parameter. Note the `\` to escape the `:` used within a string. There are routines in the `cList` class in the SDK for doing the parsing of the lists. They should be used for this task, as they will deal with the different list types and escaping of characters properly.

Breaking this list down completely gives the possible pairs of:

- ▶ -1 Off
- ▶ 0 Full
- ▶ 1333333 4:3
- ▶ 1777778 16:9
- ▶ 1850000 1.85:1
- ▶ 1896296 1.9:1
- ▶ 2400000 2.4:1
- ▶ -2 User



- ▶ -3 Absolute

So now to change the mode to Off, send the following SET command:

- ▶ # \$EXT:S:F0MODE:-1:

PROJECT GROUP

The project group includes two broad groups of parameters for the definition of project metadata and the setup of project settings such as sensor and project frame rates and REDCODE factor, among others. These two groups are listed below in separate tables to distinguish them.

PROJECT GROUP METADATA PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
CAMERAOP	MSG_STR_PTR	S	G	C	Camera Operator
CAMLOC	MSG_STR_PTR	S	G	C	Camera Location
CAMMEID	MSG_INT	S	G/H	C/D	Camera ID = 0 to 25 (equivalent A to Z)
CAMPOS	MSG_INT	S	G/H	C/D	Camera Position left, right, center (see cam_pos_t)
CAMSCENE	MSG_STR_PTR	S	G	C	Scene ID, restricted to 8 characters
CAMTAKE	MSG_INT	S	G	C	Take ID, limited to range of 1 to 999
CAMSHOT	MSG_STR_PTR	S	G	C	Shot ID. Limited to 1 character. Changing this resets the Take to 1.
CAMUNIT	MSG_STR_PTR	S	G	C	Camera Unit
DIRECTOR	MSG_STR_PTR	S	G	C	Director
DOPNAME	MSG_STR_PTR	S	G	C	Director Of Photography
PRODUCTN	MSG_STR_PTR	S	G	C	Production Company/Name
PROJINFO	MSG_STR_PTR	S	G	C	Information/Copyright
SLHEAD	MSG_INT	S	G	C	Number of frames of slate to be inserted at head of file in REDCINEX. Range of 0 to 255.
SLTAIL	MSG_INT	S	G	C	Number of frames of slate to be inserted at tail of file in REDCINEX. Range of 0 to 255.

MSG_STR_PTR type metadata fields are simple human readable strings limited to 254 characters, 8 characters for CAMSCENE, 1 character for CAMSHOT. Leading and trailing spaces will be removed when stored.

EXAMPLES OF SETTING METADATA

Set the camera operator.

- ▶ # \$EXT:S:CAMERAOP:Dinwiddie Fuhrmeister III:

Set the camera ID to 'C'.

- ▶ # \$EXT:S:CAMMEID:2:

PROJECT GROUP SETUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
FILEFMT	MSG_INT	S	G/H	C/D	File formats for SSD recording (see file_format_t)



FORMAT2	MSG_INT	S	G/H	C/D	Image resolution and format. The value of this command is created by combining 3 enumerated type codes (see resolution_t, aspect_ratio_t, anamorphic_t)
FORMATST	MSG_STR_PTR			C	Current format as a string
GENLOC	MSG_INT		G	C	Current Genlock (GEN) Icon State (see genlock_state_t)
GENSRC	MSG_INT	S	G/H	C/D	Select genlock source. (see genlock_source_t)
BHDSGLO	MSG_INT	S/U	G	C	BRAIN HD-SDI genlock phase shift in pixel clocks Range of 0 to 65535. Positive value will delay HDSDI with respect to genlock reference.
TURRGLO	MSG_INT	S/U	G	C	REDCAST module genlock phase shift in pixel clocks. 0 to 65535. Delay output relative to genlock reference.
MAXRC	MSG_INT		G	C	Current allowable maximum REDcode compression ratio *100.
MAXSHA	MSG_INT		G	C	Current allowable maximum shutter angle in degrees * 1000
MAXSHT	MSG_INT		G	C	Current allowable maximum exposure denominator *1000. i.e. 48000 = 1/48 second
MINRC	MSG_INT		G	C	Current achievable minimum REDcode compression ratio *100. The media in use, frame rate and format selections may limit this to a higher number than the target REDcode.
MINSHA	MSG_INT		G	C	Current allowable minimum shutter angle in degrees * 1000
MINSHT	MSG_INT		G	C	Current allowable minimum exposure denominator *1000.
FRPRMODE	MSG_INT	S	G/H	C/D	Frame processing mode (see frame_processing_t)
FRPRNUM	MSG_INT	S	G/H	C/D	Number of frames to process, can be 2, 4, 8, or 16
PROJFPS	MSG_INT	S	G/H	C/D	Project frame rate * 1001 where possible project FPS are: 23.98,24,25,29.97,47.96,48,50,59.94. which yields 24000, 24024, 25025, 30000, 48000, 48048, 50050, and 60000. Use list handling to get current list do not hardcode this.
RAMPADUR	MSG_INT	S	G	C	Recording mode Ramp A duration in seconds. Range of 0 to 120.
RAMPAFPS	MSG_INT	S	G	C	Ramp A target FPS in frame rate * 1001
RAMPBDUR	MSG_INT	S	G	C	Ramp B duration in seconds. Range of 0 to 120.
RAMPBFPS	MSG_INT	S	G	C	Ramp B target FPS in frame rate * 1001
RAMPCDUR	MSG_INT	S	G	C	Ramp C duration in seconds. Range of 0 to 120.
RAMPCFPS	MSG_INT	S	G	C	Ramp C target FPS in frame rate * 1001
RAMPDDUR	MSG_INT	S	G	C	Ramp D duration in seconds. Range of 0 to 120.
RAMPDFPS	MSG_INT	S	G	C	Ramp D target FPS in frame rate * 1001
RAMPDRUR	MSG_INT	S	G	C	Ramp reset duration in seconds. Range of 0 to 120.
RCTARGET	MSG_INT	S	G	C	Target REDcode value * 100. The REDcode setting that user wants to achieve if media, frame rate, and format allow it.
REDCODE	MSG_INT	V	G/H	C/D	REDcode quality value * 100, where REDcode value is in the range 3 to 18. This is the value actually in use. May be different than target value. Use of REDCODE for SET command is deprecated as of Parameter set 6.1.
PRISTOR	MSG_INT	S	G/H	C/D	Primary recording storage location (see primary_storage_t)
RECMODE	MSG_INT	S	G/H	C/D	Record mode (see record_mode_t)
PRERECDR	MSG_UINT	S	G/H	C/D	Pre-record duration in seconds.



PRERECON	MSG_INT	S	G	C	Enable/disable pre-recording. 0=disable, 1=enable
RECLIMEN	MSG_INT	S	G	C	Enable/disable record duration limit. 0=disable, 1=enable
RECLIMFR	MSG_UINT	S	G	C	Total record duration limit in frames.
RECTLFR	MSG_UINT	S	G	C	Time-lapse number of frames in burst.
RECTLINT	MSG_UINT	S	G	C	Time-lapse interval in seconds * 100.
RECPRE	MSG_UINT	S	G	C	Number of pre-record frames before mark frame.
RECPOST	MSG_UINT	S	G	C	Number of post-record frames after mark frame.
SENSFPS	MSG_INT	S	G/H	C/D	Sensor frame rate * 1001. Scaled same as project frame rate but without restriction of list.
MINFPS	MSG_INT		G	C	Minimum sensor FPS * 1001
MAXFPS	MSG_INT		G	C	Maximum sensor FPS * 1001
SYNC	MSG_INT		G	C	Current Sensor Sync (SYNC) Icon State (see sync_state_t)
SHSYNC	MSG_INT	S	G/H	C/D	Select shutter sync mode. (see sync_mode_t)
TCSET	MSG_INT	S			Desired Time-of-Day time code in seconds from midnight. Frames portion of time code cannot be set. It will be forced to 0.
TCLTCOUT	MSG_INT	S	G	C	Enable/disable linear time code output. 0 = disable, 1 = enable.
CUSTOMFPS	List only	T	H	D	Custom sensor FPS list
MASTRFPS	List only		H	D	Master sensor FPS list
CUSTOMFMT	List only	T	H	D	Custom format list
MASTRFMT	List only		H	D	Master format list
CUSTOMPHS	List only	T	H	D	Custom shutter angle list
MASTRPHS	List only		H	D	Master shutter angle list
CUSTOMINT	List only	T	H	D	Custom exposure time list
MASTRINT	List only		H	D	Master exposure time list
CUSTOMISO	List only	T	H	D	Custom ISO list
MASTRISO	List only		H	D	Master ISO list
CUSTOMRC	List only	T	H	D	Custom REDCODE list
MASTRRC	List only		H	D	Master REDCODE list
CUSTOMAWB	List only	T	H	D	Custom color temperature list
MASTRAWB	List only		H	D	Master color temperature list
USECFLL	MSG_INT	S	G	C	Use custom sensor FPS list, 0=use master, 1=use custom
USECFRL	MSG_INT	S	G	C	Use custom list, 0=use master, 1=use custom
USECPHSL	MSG_INT	S	G	C	Use custom shutter angle list, 0=use master, 1=use custom
USECINTL	MSG_INT	S	G	C	Use custom exposure time list, 0=use master, 1=use custom
USECISOL	MSG_INT	S	G	C	Use custom ISO list, 0=use master, 1=use custom
USECRCL	MSG_INT	S	G	C	Use custom REDCODE list, 0=use master, 1=use custom
USECWBL	MSG_INT	S	G	C	Use custom color temperature list, 0=use master, 1=use custom
PORECFPS	MSG_UINT	S	G	C	Recording frame rate in Preview Overdrive Mode (Scarlet only)
TSRVAD	MSG_STR_PTR	S	G	C	Tethered server host IP address, string
VCODEC	MSG_INT	S	G/H	C/D	Video recording codec selection (see video_codec_t)



VCODECR	MSG_INT	S	G/H	C/D	Video codec resolution (see codec_resolution_t)
VRISPDMD	MSG_INT		G	C	Indicates if Varispeed is enabled. 0=disabled, 1 = enabled
PROXYFPS	MSG_INT		G	C	Video codec record frame rate * 1001. i.e 24000 = 23.976
PRORESH	MSG_INT		G	C	Video codec image height in lines (WEAPON and RAVEN only)
PRORESW	MSG_INT		G	C	Video codec image width in pixels (WEAPON and RAVEN only)

EXAMPLE OF CREATING FORMAT2 VALUE FIELD

The value field of the FORMAT2 parameter is a 32 bit number generated by combining four (4) separate values, much the way key codes are created. Three come from enumerated types and one that is simply 0 or 1. This allows the specification of the image format resolution, aspect ratio, anamorphic aspect, and look-around all at once. It is possible to create combinations that are not legal formats for the camera. If this happens, it will be rejected and no change will occur.

The bits of the value field are allocated as:

- ▶ Reserved (bit 17-31), set to 0
- ▶ Look-around (bit 16), 0 = off, 1 = on
- ▶ Anamorphic (bits 12-15), see anamorphic_t type definition
- ▶ Aspect ratio (bits 4-11), see aspect_ratio_t type definition
- ▶ Resolution (bits 0-3), see resolution_t type definition

To create the code for 6k WS (widescreen 2.4:1) with 2x anamorphic, refer to rcp_types_public.h and select the values as below

- ▶ Look-around = 0
- ▶ Anamorphic = ANA_2 = 1
- ▶ Aspect ratio = AR_2_4_1 = 5
- ▶ Resolution = RESOLUTION_6K = 9
- ▶ Putting them all together in hexadecimal format gives 0x01059 = 4185
 - Easy way to assemble value is:
 - Value = (look_around << 16) + (anamorphic << 12) + (aspect << 4) + resolution

The correct command to set this in the camera would be

- ▶ #EXT:S:FORMAT2:4185:

EXAMPLE OF SETTING TIME CODE

The camera's time-of-day time code is set by specifying the time in seconds since midnight. Note that this is not the same as setting the camera's real time clock. To get the values to use for this parameter, multiply and sum the hours, minutes and seconds as below. The number of frames cannot be set, it will be forced to 0 at the time of setting the time code.

- ▶ Value = hours*3600 (seconds/hour) + minutes*60 (seconds/minute) + seconds
- ▶ 10:16:34:00 = 10*3600 + 16*60 + 34 = 36994
- ▶ #EXT:S:TCSET:36994:



STATE GROUP

The state group allows switching the camera between preview and playback modes, starting and stopping recording and turning off the camera. The camera cannot be turned on remotely by RCP. The camera must be in preview mode to record.

The PLAYBACK and RECORD parameters are asymmetric in terms of the value types used for the SET versus the GET and CURRENT commands. This permits representing the distinction of being in playback mode, but not actually play anything. Starting of playing content is controlled by parameters in the PLAYBACK group. There is an example of that in a later section. Also the recording operation has several phases that are reported.

STATE GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
CAMMODE	MSG_INT	S	G	C	Capture camera mode, i.e motion versus still (see camera_capture_mode_t)
PLAYBACK	MSG_INT	S	G	C	Switch between playback/preview. (See set_playback_state_t/playback_state_t)
PREREC	None	S			Start Pre-Record.
PWRSAVE	MSG_INT	S	G	C	Power save state (see power_save_state_t)
TSRVSTAT	MSG_INT	S	G	C	Tethered server state (see tethered_server_state_t)
RECORD	MSG_INT	S	G	C	Start/Stop Recording. (See set_record_state_t/record_state_t for parameters)
SHUTDOWN	none	S		C	Shutdown camera. CURRENT returns empty value field just before shutting down.

With the PLAYBACK and RECORD parameters it is important to process the CURRENT commands that get returned for positive indication of the state change. Changing modes and stopping and starting record can take some amount of time so protection against sending changes before the previous command takes affect should be included.

EXAMPLES OF CONTROLLING RECORDING

Start recording and then stop recording with discrete commands.

- ▶ #EXT:S:RECORD:1: (1 = SET_RECORD_STATE_START in set_record_state_t)
- ▶ make sure #EPIC:C:RECORD:1: is received before next step. (1 = RECORD_STATE_RECORDING in record_state_t)
- ▶ #EXT:S:RECORD:0: (0 = SET_RECORD_STATE_STOP)
- ▶ make sure #EPIC:C:RECORD:0: is received before starting another record. (0 = RECORD_STATE_NOT_RECORDING)

Start recording and then stop recording with toggle.

- ▶ #EXT:S:RECORD:2: (2 = SET_RECORD_STATE_TOGGLE)
- ▶ Make sure #EPIC:C:RECORD:1: is received before next step.
- ▶ #EXT:S:RECORD:2:
- ▶ Make sure #EPIC:C:RECORD:0: is received before starting another record



SYSTEM GROUP

The SYSTEM group is a collection of parameters of various functionalities. They cover temperature reporting, media content operations, time code posting, and power supply information.

SYSTEM GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
AFRCKNPT	MSG_INT	S	G/H	C/D	Number of autofocus rack points. Range of 2 to 4.
AFRCKSPD	MSG_UINT	S	G	C	Multiplier for lens speed during rack point changes. Range of 1 to 50.
AFRCKTO	MSG_INT	S			Rack to specific rack point. 0=A, 1=B, 2=C, 3=D
APAUMOTN	MSG_INT	S	G	C	Enable/disable automatic updating of motion auto preset. 0=disable. 1= enable.
APAUPLAY	MSG_INT	S	G	C	Enable/disable automatic updating of playback auto preset. 0=disable. 1= enable.
APAUSTIL	MSG_INT	S	G	C	Enable/disable automatic updating of stills auto preset. 0=disable. 1= enable.
APMOTION	MSG_STR_PTR	S	G/H	C/D	Specifies which preset is automatically loaded when entering motion capture mode.
APPLAY	MSG_STR_PTR	S	G/H	C/D	Specifies which preset is automatically loaded when entering playback mode.
APSTART	MSG_STR_PTR	S	G/H	C/D	Specifies which preset is automatically loaded when starting up.
APSTILLS	MSG_STR_PTR	S	G/H	C/D	Specifies which preset is automatically loaded when entering stills capture mode.
CALINT	MSG_INT		G	C	Current calibration integration time. Denominator*1000. 48000 = 1/48 seconds.
CALTEMP	MSG_INT		G	C	Current calibration temperature. Degrees C.
CAMUPG	none	S			Checks for camera firmware upgrade package on media and generates a notification accordingly.
CLIPNAME	MSG_STR_PTR		G	C	Returns the current clip name. The one for next recording.
CLIPNMST	MSG_INT		G	C	Indicates if camera's values of reel or position match what is on media. 1=mismatch, 0=match.
CLOK	List only		H	D	Camera look list.
CLOKAPLY	MSG_STR_PTR	S			Apply specified camera look.
CLOKDEL	MSG_STR_PTR	S			Delete specified look from camera.
CLOKE2M	MSG_STR_PTR	S			Export specified camera look to media.
CLOKEA2M	none	S			Export all camera looks to media.
CLUT	none		H	D	Returns a list of 3D LUTs in the camera.
CLUTE2M	MSG_STR_PTR	S			Export the specified 3D LUT to media
CLUTEA2M	none	S			Export all 3D LUTs to media
CLUTDEL	MSG_STR_PTR	S			Delete the specified 3D LUT from the camera.
CLUTCH0	MSG_STR_PTR	S	G/H	C/D	Assign specified 3D LUT (by name) to channel 0
CLUTCH1	MSG_STR_PTR	S	G/H	C/D	Assign specified 3D LUT to channel 1
CLUTCH2	MSG_STR_PTR	S	G/H	C/D	Assign specified 3D LUT to channel 2



CLUTCH3	MSG_STR_PTR	S	G/H	C/D	Assign specified 3D LUT to channel 3
CLUTF1	MSG_STR_PTR	S	G/H	C/D	Assign specified 3D LUT to Look B
COVR	List only		H	D	Camera overlay list.
COVRDEL	MSG_STR_PTR	S			Delete specified overlay from camera.
COVRE2M	MSG_STR_PTR	S			Export specified camera overlay to media.
COVREA2M	None	S			Export all camera overlays to media.
CPRE	List only		H	D	Camera preset list.
CPREAPLY	MSG_STR_PTR	S			Apply specified camera preset.
CPREDEL	MSG_STR_PTR	S			Delete specified preset from camera.
CPREE2M	MSG_STR_PTR	S			Export specified camera preset to media.
CPREEA2M	none	S			Export all camera presets to media.
CPREUP	MSG_STR_PTR	S			Update specified preset in camera.
GESTDTL	MSG_INT	S	G	C	Enable/disable double tap left touch screen gesture. 1= Enable, 0=Disable.
GESTDTR	MSG_INT	S	G	C	Enable/disable double tap right touch screen gesture. 1= Enable, 0=Disable.
GESTPNCH	MSG_INT	S	G	C	Enable/disable pinch touch screen gesture. 1= Enable, 0=Disable.
CURRTEMP	MSG_INT			C	Current ASIC temperature in degrees Celsius. This is output every 1 second automatically
CURSENST	MSG_INT			C	Current sensor temperature in degrees Celsius. This is output every 1 second automatically
MEDACLI	MSG_CLIPINFO_PTR		G	C	Deprecated as of RCP parameter set 6.1. Use MEDAGCII. (See clipinfo_t)
MEDACLPC	MSG_INT		G	C	Total number of clips from the media. Clip index will be 0 to number of clips -1
MEDAGCII	MSG_EXTENDED_CLIPINFO_PTR		G	C	Takes a clip index number and returns clip information of name, sensor FPS, start and end time codes plus index and clip date and time (See extended_clipinfo_t for details)
MEDASRNU	MSG_STR_PTR		G	C	Media serial number.
MEDATCHD	MSG_INT		G	C	Indicates if media is attached. 1=yes, 0=no.
MEDIA	MSG_INT	S	G/H	C/D	Current media. Get list at runtime.
MEDIANUM	MSG_STR_PTR		G	C	Media model number.
MEDIAPCT	MSG_INT		G	C	Current media free space percentage 0-100, -1 indicates no media present.
MEDIAMIN	MSG_INT		G	C	Current media free space in minutes remaining.
MEDIAMOD	MSG_INT	S	G/H	C/D	Select how media remaining space is displayed, percent or time (see media_display_t)
MEDIASZ	MSG_UINT		G	C	Media size in MiB (Mebibytes).
MEDIASZA	MSG_UINT		G	C	Available media size in MiB.
MEDIAUPG	None.	S			Look for media firmware on media and download to camera.
MEDIAVER	MSG_STR_PTR		G	C	Media firmware version.



MLABEL	MSG_STR_PTR		G	C	Current media port name 'S-SSD' for side SSD. 'R-SSD' for rear SSD.
MLOK	List only		H	D	Media look list.
MLOK12C	MSG_STR_PTR	S			Import specified look from media.
MLOK1A2C	None	S			Import all looks from media.
MLUT	None		H	D	Returns a list of 3D LUTs on the media.
MLUT12C	MSG_STR_PTR	S			Import the specified 3D LUT from media
MLUT1A2C	None	S			Import all 3D LUTs from media
MOVR	List only		H	D	Media overlay list.
MOVR12C	MSG_STR_PTR	S			Import specified overlay from media.
MOVR1A2C	None	S			Import all overlays from media.
MPRE	List only		H	D	Media preset list.
MPRE12C	MSG_STR_PTR	S			Import specified preset from media.
MPRE1A2C	None	S			Import all presets from media.
MTMST0	MSG_INT		G	C	Focus motor state. (see mc_motor_status_t)
MTMST1	MSG_INT		G	C	IRIS motor state. (see mc_motor_status_t)
MTMST2	MSG_INT		G	C	Zoom motor state. (see mc_motor_status_t)
OLPF	MSG_INT	S	G/H	C/D	OLPF type. Use GET list (H) to get valid entries.
RESETDEF	None	S			Initiates a Reset Factory Defaults operation. A notification will be generated.
RESETHW	None	S			Initiates a hardware Rediscovery. A notification will be generated.
RMCAMMAC	MSG_STR_PTR		G	C	REDMOTE communication camera MAC address
RMCH	MSG_INT	S	G/H	C/D	REDMOTE communication channel. Get list at run time.
RMDMAC	List only		H	D	List of visible REDMOTE mac addresses
RMEN	MSG_INT	S	G	C	Enable/disable REDMOTE wireless communication. 0=disable, 1=enable
RMMAC	MSG_STR_PTR		G	C	MAC address of attached REDMOTE.
RMPAIR	MSG_STR_PTR	S			Pair to the REDMOTE with the specified address. A D:RMPMAC will be sent with updated list afterwards.
RMPMAC	List only		H	D	List of paired REDMOTE MAC addresses.
RMSCAN	None	S			Initiate a scan for REDMOTE devices. Progress notifications (NOTIFY) will be send while this is happening and a D:RMDMAC will be sent when complete.
RMSTATE	MSG_INT		G	C	REDMOTE connected state (see redmote_connected_state_t)
RMUNPAIR	MSG_STR_PTR	S			Unpair from the REDMOTE with the specified address. A D:RMPMAC will be sent with updated list afterwards.
RMUPG	None	S			Initiate a firmware upgrade of an attached REDMOTE. Progress notifications (NOTIFY) will be sent while it is upgrading.



SMDLG9IN	MSG_INT	S	G	C	Enable/disable small dialogs on 9inch monitor. 0=disable, 1=enable same size as 5 inch.
PLABEL	MSG_STR_PTR		G	C	Current power source name, such as 'DC'
POSTTC	MSG_INT			C	Current time code of hh:mm:ss:ff packed one field per byte in a 32 bit number. Note there is no GET command, this is sent automatically on each second boundary, so frame count will be 00.
BATTMODE	MSG_INT	S	G/H	C/D	Show level percentage versus runtime (see battery_display_mode_t)
BATTRTM	MSG_INT		G	C	Current battery run time in minutes
PWRBATL	MSG_INT		G	C	Power level = (0...100 = Percent, Above 100 = Voltage * 1000, <0 = Seconds remaining * -1). This is output automatically every 4 seconds.
PWRSRC	MSG_INT		G/H	C/D	Current Power Source (see power_source_t)
PSLEVEL1	MSG_UINT	S	G/H	C/D	Number of seconds before power save level 1 is entered. 0=disable
PSLEVEL2	MSG_UINT	S	G/H	C/D	Number of seconds before power save level 2 is entered. 0=disable
PSLEVEL3	MSG_UINT	S	G/H	C/D	Number of seconds before power save level 3 is entered. 0=disable
TCJAM	MSG_INT		G	C	Timecode Jam State 'TC' icon (see tc_sync_state_t)
TCSOURCE	MSG_INT	S	G/H	C/D	Select timecode source, BRAIN or rear (see tc_source_t)
RIGSTATE	MSG_INT		G	C	Rig metadata state 'RIG' icon (see metadata_status_t)
NWSTAT	MSG_INT		G	C	Network state 'LAN' icon 0=disabled or no connection, 2=connected, 3=enabled but port is unreliable.
WLANSTAT	MSG_INT		G	C	WLAN status (see wl_status_t)
FSMODE	MSG_INT	S	G/H	C/D	Fan mode (see fan_mode_t)
RECFS	MSG_INT	S	G/H	C/D	Maximum record fan speed percentage
PREVFS	MSG_INT	S	G/H	C/D	Maxium preview fan speed percentage
FPRDELAY	MSG_INT	S	G/H	C/D	Amount of time (ms) to delay transition fan from record to preview speed at record end
DRPFRMS	MSG_INT			C	Number of dropped frames during record since boot up.
NOTIFY	MSG_NOTIFICATION_PTR		G	C	System notification dialog, see description below
NOTIFYR	MSG_NOTIFICATION_RESPONSE_PTR	S			Reply to NOTIFY, see description below.
FANTRGTT	MSG_INT	S	G/H	C/D	Adaptive mode target temperature in degrees C
FANPCTT	MSG_INT		G	C	Top Fan speed in Percent: range of 0 to 100. This is output periodically once per second. On WEAPON this is the left fan when viewed from behind camera.



FANPCTF	MSG_INT		G	C	Front Fan speed in Percent: range of 0 to 100. This is output periodically once per second. . On WEAPON this is the right fan when viewed from behind camera.
TWARN1	MSG_INT		G	C	First temperature warning threshold in degrees C
TWARN2	MSG_INT		G	C	Second temperature warning threshold in degrees C
TWARN3	MSG_INT		G	C	Third temperature warning threshold in degrees C
TPATTERN	MSG_INT	S	G	C	Display test pattern (see monitor_test_pattern_t)
TPATTONE	MSG_INT	S	G	C	Enable/Disable the audio test tone. 0=disable, 1=enable
CALSTAT	MSG_UINT		G	C	Calibration offset warning: high word is Integration time delta, low word is temperature delta. Deltas are in the range -3 to +3, shifted to 0 to 6 for this message. Each step in temperature delta is 5 degree C (4 for DRAGON)
EVFTALLY	MSG_INT	S	G	C	Enable/Disable display of Tally light on EVF during record.
GEIPADDR	MSG_UINT	S	G	C	Network IP address of camera. 32 bit number where each byte represents one octet of the address.
GENETMSK	MSG_UINT	S	G	C	Netmask. Each byte is one octet.
GEGWADDR	MSG_UINT	S	G	C	Gateway address. Each byte is one octet
GEDHCP	MSG_INT	S	G	C	Enable/Disable DHCP. 0=disable, 1=enable
GEC2C	MSG_INT	S	G	C	Enable/Disable camera to camera control on UDP. 0=disable, 1=enable
GEEXT	MSG_INT	S	G	C	Enable/Disable external camera control on TCP. 0=disable, 1=enable
SERPROTO	MSG_INT	S	G/H	C/D	Select serial protocol (see uart_protocol_t)
GPIN	MSG_INT	S	G/H	C/D	Select function of the general purpose input signal (trigger). (see gpi_function_t)
GPOUT	MSG_INT	S	G/H	C/D	Select function of the general purpose output signal (tally). (see gpo_function_t)
SAVELOG	none	S			Send with no value to initiate saving a log to media.
TAGGED	none			C	Sent when a frame is tagged.
EJECT	MSG_INT	S			Eject the specified media. (see media_loc_t)
FMTREELN	MSG_INT	S	G	C	Reel number for format dialog
FMTCAMID	MSG_INT	S	G/H	C/D	Camera ID for format dialog. Letter A through Z. Get list at runtime for values
FMTCAMPO	MSG_INT	S	G/H	C/D	Camera position. Get list at runtime for values.
FMTETCS	MSG_INT	S	G	C	Edge time code offset from 00:00:00:00 in seconds
FMTMEDIA	MSG_INT	S			Format the specified media (see media_loc_t)
SFMTMEDI	MSG_INT	S			Secure format the specified media (see media_loc_t)
DROPREG	MSG_UINT			C	Burst mode drop region indicator (see drop_region_t)



TARGET	MSG_INT	S	G/H	C/D	RCP Target for outgoing messages. All, none or custom (see rcp_target_t)
DATETIME	MSG_INT	S	G	C	Real time clock time as seconds since 1/1/1970
TIMEZONE	MSG_INT	S	G/H	C/D	Time Zone offset from GMT. Range of -11 to +12
UILOCK	MSG_INT	S	G	C	Lock/unlock the UI. 0=unlock, 1=lock
R3DST	MSG_INT		G	C	R3D recording state (see redcode_raw_record_state_t)
QTST	MSG_INT		G	C	Quicktime recording state (see quicktime_record_state_t)
SSDTALLY	MSG_INT	S	G	C	Enable/disable tally light on Side SSD during recording (WEAPON only)
CALENTRY	MSG_STR_PTR	S	G/H	C/D	Active sensor calibration map entry, by name. Use SET to load the map with given name.
POPIOGE	MSG_INT	S	G	C	PRO-IO GPIO port aux power enable, 1=enable, 0=disable
POPIOGS	MSG_INT		G	C	PRO-IO GPIO port aux power status CURRENT returns 0=off, 1=on, -1 if not attached
POPIOGR	none	S			PRO-IO GPIO port aux power reset send SET without data to cause reset of port
POPIOPE	MSG_INT	S	G	C	PRO-IO PWR port aux power enable, 1=enable, 0=disable
POPIOPS	MSG_INT		G	C	PRO-IO PWR port aux power status CURRENT returns 0=off, 1=on, -1 if not attached
POPIOPR	none	S			PRO-IO PWR port aux power reset send SET without data to cause reset of port
POPIOAE	MSG_INT	S	G	C	PRO-IO AUX port power enable, 1=enable, 0=disable. This port does not support reset.
POPOPE	MSG_INT	S	G	C	PLUS ONE PWR port aux power enable, 1=enable, 0=disable
POPOPS	MSG_INT		G	C	PLUS ONE PWR port aux power status CURRENT returns 0=off, 1=on, -1 if not attached
POPOPR	none	S			PLUS ONE PWR port aux power reset send SET without data to cause reset of port
POBAE	MSG_INT	S	G	C	Battery module AUX port power enable, 1=enable, 0=disable
POBAS	MSG_INT		G	C	Battery module AUX port power status CURRENT returns 0=off, 1=on, -1 if not attached
POBAC	MSG_INT		G	C	Battery module AUX port CURRENT returns current draw in milliamps.
POBAR	None	S			Battery module AUX port power reset send SET without data to cause reset of port
POJPUE	MSG_INT	S	G	C	JETPACK USB port power enable, 1=enable, 0=disable
POJPUS	MSG_INT		G	C	JETPACK USB port aux power status CURRENT returns 0=off, 1=on, -1 if not attached



POJPUR	none	S			JETPACK USB port aux power reset send SET without data to cause reset of port
TETHERS	MSG_INT		G	C	Indicates if tethering is supported, 1=yes, 0=no
TETHERSI	MSG_INT		G	C	Indicates if tethering is supported on the interface that sends the GET command. 1=yes, 0=no
TSRVSP	MSG_UINT	S	G	C	Total storage space on server in Mebibytes
TSRVSPR	MSG_UINT	S	G	C	Remaining storage space on server in Mebibytes
TSRVRN	MSG_INT	S	G	C	Tethered server reel number
TSRVCN	MSG_INT	S	G	C	Tethered server clip number
WLANMODE	MSG_INT	S	G/H	C/D	WiFi mode (see wl_mode_t)
WLANAHEN	MSG_INT	S	G/H	C/D	WiFi ad hoc encryption type (see wl_encryption_t)
WLANAHPW	MSG_STR_PTR	S	G	C	WiFi ad hoc passphrase when using encryption
WLANAHID	MSG_STR_PTR		G	C	WiFi ad hoc SSID
WLANCHAN	MSG_INT	S	G/H	C/D	WiFi channel
WLANAPEN	MSG_INT	S	G/H	C/D	WiFi infrastructure mode encryption type (see wl_encryption_t)
WLANAPPW	MSG_STR_PTR	S	G	C	WiFi infrastructure mode passphrase.
WLANAPID	MSG_STR_PTR	S	G	C	WiFi infrastructure mode SSID
WLANSIG	MSG_INT		G	C	WiFi signal strength in dBm
WLANCONN	None	S			Trigger WiFi connection based on parameters in WLANAPEN, WLANAPPW, and WLANAPID
WLANSKAN	None	S			Start wireless scan. Send SET without data.
WLANSKRL	MSG_STR_PTR		H	D	WiFi scan results returned as list of strings of SSID name for each detected device.
WLANSKCS	MSG_STR_PTR	S			WiFi scan select. Perform SET with SSID of selected WiFi source returned from WLANSKRL.
WLANIP	MSG_UINT		G	C	WiFi IP address assigned by DHCP in infrastructure mode. 0 if none assigned. Leftmost octet in MS byte.
BEEPEN	MSG_INT	S	G	C	Enable/Disable all beeps. 0=disable, 1=enable
BEEPREC	MSG_INT	S	G/H	C/D	Select sound to play when record is started. (see audio_signal_t).
BEEPSTOP	MSG_INT	S	G/H	C/D	Select sound to play when record is stopped. (see audio_signal_t).
BEEPTAGS	MSG_INT	S	G/H	C/D	Select sound to play when frame is tagged. (see audio_signal_t).
RESETUSR	None	S			Performing a SET with no data causes a restoration of default settings, followed by deleting all user files. These include calibration files, looks, LUTs, presets, and overlays. A confirmation notification will be generated before proceeding.
CCAL	List only		H	D	In camera sensor calibration file list
CCALAPLY	MSG_STR_PTR	S			Apply the specified in camera calibration file
CCALDEL	MSG_STR_PTR	S			Delete the specified in camera calibration file
CCALE2M	MSG_STR_PTR	S			Export the specified in camera calibration to media



CCALEA2M	None	S			Export all in camera calibration files to media
MCAL	MSG_INT		H	D	On media sensor calibration file list
MCALAPLY	MSG_STR_PTR	S			Apply the specified on media calibration file
MCALI2C	MSG_STR_PTR	S			Import the specified on media calibration file to the camera
MCALIA2C	None	S			Import all on media calibration files to the camera
OGTSTATE	MSG_INT		G	C	Open gate state, off or active (see open_gate_state_t)
OGTMODE	MSG_INT	S	G	C	Open gate mode, allowed or not allowed (see open_gate_mode_t)

EXAMPLE OF DECODING TIME CODE FROM POSTTC

The POSTTC parameter is sent out with a CURRENT command by the camera on every :00 frame boundary (once per second) when TOD time code is selected. This cannot be turned off. When edge time code is selected POSTTC is only transmitted when the time code is changing (upon a media format or while recording). A typical POSTTC message will look as below:

- ▶ # \$DRAGON:C:186459904:
 - To decode, convert the decimal value to hexadecimal giving 0x0B1D2700
 - Each byte then maps to hours, minutes, seconds, and frames
 - Hours = 0x0B = 11
 - Minutes = 0x1D = 29
 - Seconds = 0x27 = 39
 - Frames = 0x00 = 00
 - Giving a time-of-day time code of 11:29:39:00

EXAMPLE OF MEDAGCII EXTENDED CLIP DATA

The MEDAGCII parameter returns the extended clip info for the index given. The index is simply the number, 0 to count-1 of all the clips in the media. The count of clips is retrieved with the MEDACLPC parameter. The index requested is returned in the response so it can be checked that the clip did exist. Giving an index outside the range of 0 to count-1 will return the data for the closest clip to the requested index. The full media content details can be gathered by first using MEDACLPC to get the total count, and then iterating, sending a MEDAGCII for each valid index from 0 to count-1. Below is an example of getting one clip's data and interpreting the value fields. Note that if a clip is corrupted, some of these fields may be missing or corrupted as well.

Request:

- ▶ # \$EXT:G:MEDAGCII:0:

Response:

- ▶ # \$DRAGON:C:MEDAGCII:0:A002_C001_0906NM:09/06/2013:15\52\54:24000:E\01\00\00\00:E\01\00\02\07:T\15\52\55\03:T\15\52\57\10:

Data interpretation:

- ▶ 0 clip index 0



- ▶ A002_C001_0906NM clip name
- ▶ 09/06/2013 clip date
- ▶ 15\52\54 clip time, note the ‘\’ escape characters needed to allow ‘:’ in the data
- ▶ 24000 clip sensor FPS, in this case 24000/1001 = 23.976
- ▶ E\01\00\00\00 clip start edge time code, note the ‘\’
- ▶ E\01\00\02\07 clip end edge time code
- ▶ T\15\52\55\03 clip start time of day time code
- ▶ T\15\52\57\10 clip end time of day time code

USE OF NOTIFY AND NOTIFYR

The NOTIFY command is used to cause a pop-up notification dialog. The notification can be dismissed by the client or the camera depending on the situation. For example the media formatting progress is automatically dismissed, whereas a notification that requires a user response is dismissed by sending that response. The NOTIFYR command is used to send a response when needed.

- ▶ `#$CAMERA_NAME:C:NOTIFY:[BASE_64_ENCODED_UUID]:[TITLE]:[MESSAGE]:[PROGRESS_BAR_TYPE]:[PROGRESS_PERCENT]:[RESPONSES_CLIST]:[TIMEOUT]:[TYPE]:`

Where:

- ▶ `BASE_64_ENCODED_UUID`: unique ID for this notification instance
- ▶ `TITLE`: title of notification (single line)
- ▶ `MESSAGE`: message of notification. Hard line breaks are specified with “|”. Line wrapping should be done on the client side
- ▶ `PROGRESS_BAR_TYPE`: 0 = none, 1 = normal, 2 = infinite. Note that it is up to the client to auto-update infinite progress bar.
- ▶ `PROGRESS_PERCENT`: 0 – 100
- ▶ `RESPONSES_CLIST`: cList of all responses available. The text for each entry should be the label used on buttons, the int value should be sent back as the user response. The most number of buttons used to date is 3. There is a practical limit to what can fit on the UI, but there is no coded limit.
- ▶ `TIMEOUT`: if 0 notification is shown until it has been removed. Otherwise the number of seconds notification should be shown before dismissing itself
- ▶ `TYPE`: type of notification

Note, the same message is sent when an update to the notification occurs. Note that `MESSAGE` and `PROGRESS_PERCENT` are the only fields that can be updated

Example:

- ▶ `#$EPIC:C:NOTIFY:q4wBALMRARCAASy2nQAE1w==:Media:Side SSD media removed without first ejecting. Data integrity risk.:0:0:0|0|1|OK|:0:21:`

When a notification is to be removed a NOTIFY message with only the `BASE_64_ENCODED_UUID` field is sent, example:

- ▶ `#$DEFAULT:S:NOTIFY: q4wBALMRARCAASy2nQAE1w==:`

Note that this message will be sent if the notification is responded to on any device (or the system decides to close it on its own). If you are generating your own notification pop-up on some other device, this is when



you would close it.

When an external device (such as REDmote, iPad, etc) connects to a camera it should request all the currently active notifications so that they can be displayed on the external device. Sending a GET NOTIFY message does this:

▶ #`$EXT_DEV:G:NOTIFY:0:`

The camera will respond with a CURRENT NOTIFY message for every active notification (or 0 if there are no active notifications)

The response to a NOTIFY is NOTIFYR. The target notification is identified by using the corresponding UUID.

▶ #`$DEVICE_NAME:C:NOTIFYR:[BASE_64_ENCODED_UUID]:[RESPONSE]:`

Where:

- ▶ `BASE_64_ENCODED_UUID`: UUID of notification the user is responding to
- ▶ `RESPONSE`: INT value from user options `RESPONSES_CLIST`

PLAYBACK GROUP

The PLAYBACK group consists of parameters to initiate and control in-camera playback of recorded clips. To use these parameters, the camera must first be put into playback state and the change to that state confirmed (see STATE group for more information).

PLAYBACK GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
CLIPLEN	MSG_INT		G	C	Clip length of currently loaded clip in frames
CLIPLOAD	MSG_INT	S	G	C	Select playback clip based on clip index in file list (See SYSTEM group for info on getting clip list). Clip index is 0 based.
CLPDIR	MSG_INT	S	G	C	Set clip playback direction (See <code>player_direction_t</code>)
CLPINPT	MSG_INT	S	G	C	Play clip in point frame number, 0 based.
CLPNUM	MSG_INT	S			Playback Clip number +/- relative to current loaded clip
CLPOUTPT	MSG_INT	S	G	C	Play clip out point frame number, 0 based.
LOOP	MSG_INT	S	G	C	Play clip in loop, looped list or no loop (See <code>player_loop_t</code>)
PAUSE	MSG_INT	S		C	Pause current clip. A value is required but it is ignored. Always returns current of 0 as an acknowledge.
PLAY	MSG_INT	S		C	Play current clip. A value is required but it is ignored. Always returns current of 0 as an acknowledge.
PLAYPAUS	MSG_INT	S			Toggle Play/Pause current clip. A value is required but it is ignored. Causes a C:PAUSE or C:PLAY as appropriate.
PLAYPOS	MSG_INT	S	G	C	Current playback frame number, 0 based
PLAYRPOS	MSG_INT	S			Change playback frame to new frame +/- x frames relative to current frame
PLAYSPD	MSG_INT	S	G	C	Set Playback speed in multiples of two, 1x, 2x, 4x, 8x, or 16x
PBCLSTDT	MSG_STR_PTR		G	C	Clip metadata start date. Format is YYYYMMDD
PBCLSTTM	MSG_STR_PTR		G	C	Clip metadata start time. Format is HHMMSS



PBCOLTMP	MSG_INT		G	C	Clip metadata color temp in degrees kelvin.
PBHDRTRC	MSG_INT		G	C	Playback HDR track count
PBISO	MSG_INT		G	C	Clip metadata ISO value
PBTINT	MSG_INT		G	C	Clip metadata Tint value * 1000. Where Tint is in the range -100.0 to +100.0
PBZOOM	MSG_INT		G	C	Clip metadata lens zoom in millimeters * 100.

PLAYBACK OPERATION

When playback is invoked by changing the state of the camera, the last recorded clip is loaded by default. Other clips can be chosen either by index and loaded with CLIPLOAD or by stepping up or down with CLIPNUM. The selected clip is then played back and controlled using PLAY, PLAYPAUS, LOOP and PLAYSPD. Clip in and out points can be set with CLPINPT and CLPOUTPT, but they only pertain to the current clip and are not remembered if the clip is changed.

EXAMPLE OF CHANGE TO PLAYBACK AND PLAY LAST CLIP

On entering playback, the last clip is automatically loaded, with in point set to first frame and direction and looping set to previous selections. Defaults for looping and direction are no looping and forward. Playback is simply started with PLAY. The clip is then paused and the camera put back in Preview mode.

- ▶ #EXT:S:PLAYBACK:1: (1 = SET_PLAYBACK_STATE_START in set_playback_state_t)
- ▶ make sure #EPIC:C:PLAYBACK:1: is received before next step. (1 = PLAYBACK_STATE_IN_PLAYBACK in playback_state_t)
- ▶ #EXT:S:PLAY:0:
- ▶ #EXT:S:PAUSE:0:
- ▶ #EXT:S:PLAYBACK:0: (0 = SET_PLAYBACK_STATE_STOP in set_playback_state_t)
- ▶ make sure #EPIC:C:PLAYBACK:0: is received before next step. (0 = PLAYBACK_STATE_NOT_IN_PLAYBACK in playback_state_t)

STATS GROUP

The STATS group comprises a group of parameters that are automatically emitted on a regular basis or as needed.

STATS GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
DSHIST	MSG_STR_PTR			C	Down sampled Histogram Data, [bottom clip % (hex encoded byte)][top clip % (hex encoded byte)](for R, G, B channels: [starting index (hex encoded byte)][number of cols (hex encoded byte)][packed data – each col is 4 bits (base64 encoded string)]). Updated 10 times per second.
HISTTYPE	MSG_INT	S	G/H	C/D	Type of histogram being displayed. (see histogram_type_t)
MROLL	MSG_INT			C	Camera Roll in degrees*10000 Positive value is a roll to the right about the axis of the lens as viewed from behind camera. Updated at 3.33 times per second, or 10 times in 3 seconds.



MPITCH	MSG_INT			C	Camera Pitch in degrees*10000 Positive value is pitch up of the front of the camera. Updated at 3.33 times per second, or 10 times in 3 seconds.
RCLIP	MSG_INT		G	C	Red Clipping Indicator, based on the threshold value send message to turn on or off the green indicator. 0 = off, 1 = on.
GCLIP	MSG_INT		G	C	Green clipping indicator
BCLIP	MSG_INT		G	C	Blue clipping indicator
AUDIOVU	MSG_AUDIO_VU_DATA			C	VU meter peak readings in DB. 10 values in comma separated list. 4 input channels and 6 output channels. Updated at 10 Hz. See example below for processing requirements.

CLIP INDICATORS

Clip indicators are only sent when they change. If they become set before the connection to a camera is made the application will have no way to know the correct state unless it changes and a new CURRENT command is emitted. Starting with RCP parameter set 4.1 (firmware version 5.1.35), the GET option was added so that the initial state can be queried upon camera connection. Any application that needs to monitor these indicators should issue a GET upon starting up.

DECODING DSHIST DATA

The data returned by the DSHIST in a CURRENT command is formatted as below.

`#$EPIC:C:DSHIST:[BOTTOM_CLIP][TOP_CLIP][RED_DATA][GREEN_DATA][BLUE_DATA][LUMA_DATA]:`

Where:

BOTTOM_CLIP: percent of bottom clipping 0-100 (hex encoded byte, that is: 0->x00, 100->x64). This indicates how tall to draw the clip goal post.

TOP_CLIP: percent of top clipping 0-100 (hex encoded byte, that is: 0->x00, 100->x64)

RED_DATA, GREEN_DATA, BLUE_DATA, LUMA_DATA: histogram data for each channel. Unused channels will have a single column of 0. When histogram display mode is RGB, Luma will be unused. When histogram display mode is Luma, red, green and blue will be unused.

Each histogram data section has the following form:

`[START_COL][NUM_COLS][PACKED_DATA]`

START_COL: starting column of packed data (hex encoded byte). Columns are numbered 0 to 127.

NUM_COLS: number of columns in packed data (hex encoded byte)

PACKED_DATA: each column is four (4) bits of data (**base 64 encoded**)

The packed data format (before base64 encoding):

byte 0								byte 1								...	byte n							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	...	7	6	5	4	3	2	1	0
col 0				col 1				col 2				col 3				...	col n*2				col n*2 + 1			

An odd number of columns will have an extra nibble in byte n that is to be disregarded. This will be the low nibble (bits 0 to 3).



Base64 encoding allows representation of binary data as ASCII characters. It generates four (4) bytes of characters for every three (3) bytes of data and pads on the end with '=' character(s). So the length of the encoded packed data can be calculated using integer math and truncating divides as:

- ▶ Number of base64 characters = $\left(\left(\left(\left(\text{columns} + 1\right) / 2\right) + 2\right) / 3\right) * 4$

Note: Each data channel contains a start column and spans the number of columns specified. This is used as a very simple compression method since empty columns on the far left and right do not need to be transferred. Each column not in the given range should be interpreted as 0. One or more of the channels may contain no data (that is, NUM_COLS is 0).

Note: One or more of the channels may contain no data (that is, NUM_COLS is 0). In this case the channel should not be drawn at all.

Using the following captured DSHIST output, we can decode it as follows. This was with lens cap just cracked, so it is a very dark scene which makes the color data very small. Normal well-exposed scenes will make this message larger.

```
#$EPIC:C:DSHIST:0600021bEjRWeaqYiaqa79qYdjA=0219E0aKuniZqqveyYh2IA==001eESNGmqmlqqvMy6mHdIQh0001AA==:
```

- ▶ 06 = 6% bottom clipping
- ▶ 00 = 0% top clipping

Red_Data:

- ▶ 02 = first column is 2. Columns are numbered 0 to 127
- ▶ 1b hex = 27 columns. Yielding $\left(\left(\left(\left(27+1\right) / 2\right) + 2\right) / 3\right) * 4 = 20$ characters of data.
- ▶ EjRWeaqYiaqa79qYdjA= is the packed data

Green_Data:

- ▶ 02 = first column is 2. Columns are numbered 0 to 127
- ▶ 19 hex = 25 columns. Yielding $\left(\left(\left(\left(25+1\right) / 2\right) + 2\right) / 3\right) * 4 = 20$ characters of data.
- ▶ E0aKuniZqqveyYh2IA== is the packed data

Blue_Data:

- ▶ 00 = first column is 0. Columns are numbered 0 to 127
- ▶ 1e hex = 30 columns. Yielding $\left(\left(\left(\left(30+1\right) / 2\right) + 2\right) / 3\right) * 4 = 20$ characters of data.
- ▶ ESNGmqmlqqvMy6mHdIQh is the packed data

Luma_Data:

- ▶ 00 = first column is 0. Columns are numbered 0 to 127
- ▶ 01 hex = 1 columns. Yielding $\left(\left(\left(\left(1+1\right) / 2\right) + 2\right) / 3\right) * 4 = 4$ characters of data.
- ▶ AA== is the packed data which means 0 in this case.

DECODING AUDIOVU DATA

Audio VU meter data is sent as a list of 10 integer numbers, one for each of the 4 input channels, followed by the 6 output channels. All numbers are peak readings in dB. Order is input channels 1, 2, 3, 4, then



outputs CAM MIX, REAR MIX, MONITOR MIX. Raw values must be processed as follows:

$$\text{dB} = (\text{raw value} - 20) \text{ clipped to the range } -52 \text{ to } 0.$$

Disabled channels report raw values of 0, which is a legal value for an enabled channel. The parameters CH12ST and CH34ST must be used to determine whether to process those channel pairs.

An example message is below. In this case only camera analog input channel 1 and 2 are enabled but with no microphone on channel 2 and no attenuations have been applied in the output mixing stage.

```
▶ # $DRAGON:C:AUDIOVU:5:-58:0:0:5:-58: 5:-58: 5:-58: 5:-58:
```

This results in channel 1 dB of -15, channel 2 dB of -52 and the same values on the outputs.

AUDIO GROUP

The AUDIO group provides for control of on audio input gain and output volume.

AUDIO GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
CH12SRC	MSG_INT	S	G/H	C/D	Select channel 1 / 2 source. (see audio_channel_source_t)
CH34SRC	MSG_INT	S	G/H	C/D	Select channel 3 / 4 source. (see audio_channel_source_t)
CH12ILNK	MSG_INT	S	G	C	Enable/Disable gain slider linking for channel 1 / 2. 0=disable, 1=Enable
CH34ILNK	MSG_INT	S	G	C	Enable/Disable gain slider linking for channel 3 / 4. 0=disable, 1=Enable
CH12OLNK	MSG_INT	S	G	C	Enable/Disable volume slider linking for channel 1 / 2. 0=disable, 1=Enable
CH34OLNK	MSG_INT	S	G	C	Enable/Disable volume slider linking for channel 3 / 4. 0=disable, 1=Enable
CH12ONME	MSG_STR_PTR		G	C	Channel 1 / 2 output name (string)
CH34ONME	MSG_STR_PTR		G	C	Channel 3 / 4 output name (string)
CH56ONME	MSG_STR_PTR		G	C	Headphone (MON) channel output name (string)
CH12ST	MSG_INT		G	C	Channel 1 / 2 audio status, CH1 and CH2 icon, 0=off, 1=on
CH34ST	MSG_INT		G	C	Channel 3 / 4 audio status, CH3 and CH4 icon, 0=off, 1=on
VUSRC	MSG_INT	S	G/H	C/D	Selects source of signals for the VU meter display, either input channels or output channels. 0=input, 1=output
CH1MODE	MSG_INT	S	G/H	C/D	Select channel 1 mode, balanced/unbalanced/phantom/etc. Options change with source (see audio_channel_mode_t)
CH2MODE	MSG_INT	S	G/H	C/D	Select channel 2 mode, balanced/unbalanced/phantom/etc.
CH3MODE	MSG_INT	S	G/H	C/D	Select channel 2 mode, balanced/unbalanced/phantom/etc.
CH4MODE	MSG_INT	S	G/H	C/D	Select channel 2 mode, balanced/unbalanced/phantom/etc.
CH1NAME	MSG_STR_PTR	S	G	C	String for channel 1 name, max 6 characters
CH2NAME	MSG_STR_PTR	S	G	C	String for channel 2 name, max 6 characters
CH3NAME	MSG_STR_PTR	S	G	C	String for channel 3 name, max 6 characters
CH4NAME	MSG_STR_PTR	S	G	C	String for channel 4 name, max 6 characters
CH1GAIN	MSG_INT	S	G/H	C/D	Channel 1 audio gain, range 8 to 38, representing +30 to +60 dB, 1 dB per lsb



CH2GAIN	MSG_INT	S	G/H	C/D	Channel 2 audio gain, see CH1GAIN
CH3GAIN	MSG_INT	S	G/H	C/D	Channel 3 audio gain, see CH1GAIN
CH4GAIN	MSG_INT	S	G/H	C/D	Channel 4 audio gain, see CH1GAIN
CH1VOL	MSG_INT	S	G/H	C/D	Channel 1 audio volume, range 71 to 127, representing -28 to 0 dB, 0.5dB per lsb
CH2VOL	MSG_INT	S	G/H	C/D	Channel 2 audio volume, see CH1VOL
CH3VOL	MSG_INT	S	G/H	C/D	Channel 3 audio volume, see CH1VOL
CH4VOL	MSG_INT	S	G/H	C/D	Channel 4 audio volume, see CH1VOL
CH1LIMIT	MSG_INT	S	G	C	Enable/Disable channel 1 limiter. 0=disable, 1=enable
CH2LIMIT	MSG_INT	S	G	C	Enable/Disable channel 2 limiter. 0=disable, 1=enable
CH3LIMIT	MSG_INT	S	G	C	Enable/Disable channel 3 limiter. 0=disable, 1=enable
CH4LIMIT	MSG_INT	S	G	C	Enable/Disable channel 4 limiter. 0=disable, 1=enable
CH1LIMPR	MSG_INT		G	C	Indicates if channel 1 limiter is present. 0=no, 1=yes
CH2LIMPR	MSG_INT		G	C	Indicates if channel 2 limiter is present. 0=no, 1=yes
CH3LIMPR	MSG_INT		G	C	Indicates if channel 3 limiter is present. 0=no, 1=yes
CH4LIMPR	MSG_INT		G	C	Indicates if channel 4 limiter is present. 0=no, 1=yes
CH1MUTE	MSG_INT	S	G	C	0=not muted, 1 = muted
CH2MUTE	MSG_INT	S	G	C	0=not muted, 1 = muted
MIX12VL1	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX12VL2	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX12VL3	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX12VL4	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX34VL1	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX34VL2	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX34VL3	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX34VL4	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIX12PN1	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX12PN2	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX12PN3	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX12PN4	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX34PN1	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX34PN2	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX34PN3	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIX34PN4	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIXHDVL1	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIXHDVL2	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIXHDVL3	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIXHDVL4	MSG_INT	S	G/H	C/D	Mix volume 0 to 100%, range 0 to 255
MIXHDPN1	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIXHDPN2	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIXHDPN3	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127
MIXHDPN4	MSG_INT	S	G/H	C/D	Mix pan 100% left to 100% right, range -127 to 127



CH1_48V	MSG_INT	S	G	C	Enable/Disable channel 1 48v power. 0=disable, 1=enable
CH2_48V	MSG_INT	S	G	C	Enable/Disable channel 2 48v power. 0=disable, 1=enable
CH3_48V	MSG_INT	S	G	C	Enable/Disable channel 3 48v power. 0=disable, 1=enable
CH4_48V	MSG_INT	S	G	C	Enable/Disable channel 4 48v power. 0=disable, 1=enable
CH1_48VP	MSG_INT		G	C	Indicates if channel 1 can source 48v. 0=no, 1=yes
CH2_48VP	MSG_INT		G	C	Indicates if channel 2 can source 48v. 0=no, 1=yes
CH3_48VP	MSG_INT		G	C	Indicates if channel 3 can source 48v. 0=no, 1=yes
CH4_48VP	MSG_INT		G	C	Indicates if channel 4 can source 48v. 0=no, 1=yes

OUTPUT GROUP

The OUTPUT group provides control of the monitor mode of each monitor individually for displayed content. Content choices can be overlays, clean, or mirroring another monitor. Use the GET_LIST message to get valid choices for each, as they are not uniform.

OUTPUT GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
MAGNIFY	MSG_INT	S	G/H	C/D	Magnify mode setting (see magnify_mode_t)
MONPRTY	List only	T	H	D	Monitor priority list. Get list at run time for values.
BHDSDIEN	MSG_INT	S	G	C	Enable/disable BRAIN HD-SDI monitor. 0=disable, 1=enable
PHDS1EN	MSG_INT	S	G	C	Enable/disable Pro-IO HD-SDI monitor 1. 0=disable, 1=enable
PHDS2EN	MSG_INT	S	G	C	Enable/disable PRO-IO HD-SDI monitor 2. 0=disable, 1=enable
LCDG	MSG_INT	S	G/H	C/D	Look (A or B) going to BRAIN LCD
HDMIG	MSG_INT	S	G/H	C/D	Look (A or B) going to BRAIN HDMI
HSDDIG	MSG_INT	S	G/H	C/D	Look (A or B) going to BRAIN HDSDI
EVFG	MSG_INT	S	G/H	C/D	Look (A or B) going to BRAIN EVF
PLCDG	MSG_INT	S	G/H	C/D	Look (A or B) going to Pro-IO LCD
PHDSDI1G	MSG_INT	S	G/H	C/D	Look (A or B) going to Pro-IO HD-SDI 1
PHDSDI2G	MSG_INT	S	G/H	C/D	Look (A or B) going to Pro-IO HD-SDI 2
REVFG	MSG_INT	S	G/H	C/D	Look (A or B) going to rear EVF
LCDM	MSG_INT	S	G/H	C/D	BRAIN LCD monitor mode (see monitor_mode_t)
HDMIM	MSG_INT	S	G/H	C/D	BRAIN HDMI monitor mode
HSDDIM	MSG_INT	S	G/H	C/D	BRAIN HD-SDI monitor mode
PLCDM	MSG_INT	S	G/H	C/D	Pro-IO LCD monitor mode
PHDSDI1M	MSG_INT	S	G/H	C/D	Pro-IO HD-SDI 1 (Preview) monitor mode
PHDSDI2M	MSG_INT	S	G/H	C/D	Pro-IO HD-SDI 2 (PGM) monitor mode
EVFM	MSG_INT	S	G/H	C/D	BRAIN EVF monitor mode
REVFM	MSG_INT	S	G/H	C/D	Rear EVF monitor mode
BLCDOV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for BRAIN LCD. SET command takes a string, not integer like other lists. See example.
BHDMIOV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for BRAIN HDMI.
BHDSDIOV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for BRAIN HD-SDI.



PHDS1OV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for Pro-IO HD-SDI 1.
PHDS2OV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for Pro-IO HD-SDI 2.
PLCDOV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for rear LCD.
BEVFOV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for BRAIN EVF.
REVFOV	MSG_STR_PTR	S	G/H	C/D	Select overlay type for rear EVF.
HDMIR	MSG_INT	S	G/H	C/D	BRAIN HDMI resolution. (see monitor_resolution_t)
HDSDIR	MSG_INT	S	G/H	C/D	BRAIN HD-SDI resolution. (see monitor_resolution_t)
PHDSDI1R	MSG_INT	S	G/H	C/D	Pro-IO HD-SDI 1 resolution. (see monitor_resolution_t)
PHDSDI2R	MSG_INT	S	G/H	C/D	Pro-IO HD-SDI 2 resolution. (see monitor_resolution_t)
LCDF	MSG_INT	S	G/H	C/D	BRAIN LCD frame rate. Get the list for valid settings.
HDMIF	MSG_INT	S	G/H	C/D	BRAIN HDMI frame rate. Get the list for valid settings.
PHDSDI1F	MSG_INT	S	G/H	C/D	Pro-IO HD-SDI 1 frame rate. Get the list for valid settings.
PHDSDI2F	MSG_INT	S	G/H	C/D	Pro-IO HD-SDI 2 frame rate. Get the list for valid settings.
HDSDIF	MSG_INT	S	G/H	C/D	BRAIN HDSDI frame rate. Get the list for valid settings.
EVFF	MSG_INT	S	G/H	C/D	BRAIN EVF frame rate. Get the list for valid settings.
PLCDF	MSG_INT	S	G/H	C/D	Rear LCD frame rate. Get the list for valid settings.
REVFF	MSG_INT	S	G/H	C/D	Rear EVF frame rate. Get the list for valid settings.
BLCDTL	MSG_INT	S	G	C	Enable/Disable use of video tools on BRAIN LCD. 0=disable, 1=enable.
BHDMITL	MSG_INT	S	G	C	Enable/Disable use of video tools on BRAIN HDMI.
BHDSITL	MSG_INT	S	G	C	Enable/Disable use of video tools on BRAIN HDSDI
PLCDTL	MSG_INT	S	G	C	Enable/Disable use of video tools on rear LCD.
PSDI1TL	MSG_INT	S	G	C	Enable/Disable use of video tools on Pro-IO HD-SDI 1
PSDI2TL	MSG_INT	S	G	C	Enable/Disable use of video tools on Pro-IO HD-SDI 2
BEVFTL	MSG_INT	S	G	C	Enable/Disable use of video tools on BRAIN EVF
REVFTL	MSG_INT	S	G	C	Enable/Disable use of video tools on rear EVF
BLCDFMSP	MSG_INT		G	C	Indicates if BRAIN LCD connected monitor supports flip/mirror (rotation). 1= supported, 0=not
RLCDFMSP	MSG_INT		G	C	Indicates if rear LCD connected monitor supports flip/mirror (rotation). 1= supported, 0=not
BLCDFM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on BRAIN LCD. 0=disable, 1=enable.
BLCDFM2	MSG_INT	S	G	C	Enable/Disable UI and video flip/mirror on BRAIN LCD. 0=disable, 1=enable.
BHDMIFM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on BRAIN HDMI
BHDSIFM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on BRAIN HD-SDI
PLCDFM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on rear LCD
RLCDFM2	MSG_INT	S	G	C	Enable/Disable UI and video flip/mirror on rear LCD
PSDI1FM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on Pro-IO HD-SDI 1
PSDI2FM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on Pro-IO HD-SDI 2
BEVFFM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on BRAIN EVF
BEVFFMSP	MSG_INT		G	C	Indicates if connected BRAIN EVF supports rotation (flip/mirror). 0=not supported, 1=supported.



BEVFFM2	MSG_INT	S	G	C	Enable/Disable UI and video flip/mirror on BRAIN EVF. 0=disable, 1=enable.
REVFFM	MSG_INT	S	G	C	Enable/Disable UI flip/mirror on rear EVF
REVFFMSP	MSG_INT		G	C	Indicates if connected rear EVF supports rotation (flip/mirror). 0=not supported, 1=supported.
REVFFM2	MSG_INT	S	G	C	Enable/Disable UI and video flip/mirror on rear EVF. 0=disable, 1=enable.
BLCDVF	MSG_INT	S	G	C	Enable/Disable video flip on BRAIN LCD. 0=disable, 1=enable.
BHDMIVF	MSG_INT	S	G	C	Enable/Disable video flip on BRAIN HDMI
BHDSDIVF	MSG_INT	S	G	C	Enable/Disable video flip on BRAIN HD-SDI
PLCDVF	MSG_INT	S	G	C	Enable/Disable video flip on rear LCD
PSDI1VF	MSG_INT	S	G	C	Enable/Disable video flip on Pro-IO HD-SDI 1
PSDI2VF	MSG_INT	S	G	C	Enable/Disable video flip on Pro-IO HD-SDI 2
BEVVFV	MSG_INT	S	G	C	Enable/Disable video flip on BRAIN EVF
REVFVF	MSG_INT	S	G	C	Enable/Disable video flip on rear EVF
BLCDFO	MSG_INT	S	G	C	Enable/Disable framed overlay on BRAIN LCD. 0=disable, 1=enable.
BHDMIFO	MSG_INT	S	G	C	Enable/Disable framed overlay on BRAIN HDMI
BHDSIFIFO	MSG_INT	S	G	C	Enable/Disable framed overlay on BRAIN HD-SDI
PLCDFO	MSG_INT	S	G	C	Enable/Disable framed overlay on rear LCD
PSDI1FO	MSG_INT	S	G	C	Enable/Disable framed overlay on Pro-IO HD-SDI 1
PSDI2FO	MSG_INT	S	G	C	Enable/Disable framed overlay on Pro-IO HD-SDI 2
BEVFFO	MSG_INT	S	G	C	Enable/Disable framed overlay on BRAIN EVF
REVFFO	MSG_INT	S	G	C	Enable/Disable framed overlay on rear EVF
BLCDDM	MSG_INT	S	G	C	Enable/Disable docked menus (not auto-hide) on BRAIN LCD. 0=disable, 1=enable.
BHDMIDM	MSG_INT	S	G	C	Enable/Disable docked menus on BRAIN HDMI
BHDSIDM	MSG_INT	S	G	C	Enable/Disable docked menus on BRAIN HD-SDI
PLCDDM	MSG_INT	S	G	C	Enable/Disable docked menus on rear LCD
PSDI1DM	MSG_INT	S	G	C	Enable/Disable docked menus on Pro-IO HD-SDI 1
PSDI2DM	MSG_INT	S	G	C	Enable/Disable docked menus on Pro-IO HD-SDI 2
BEVFDM	MSG_INT	S	G	C	Enable/Disable docked menus on BRAIN EVF
REVFDM	MSG_INT	S	G	C	Enable/Disable docked menus on rear EVF
BLCDBR	MSG_INT	S	G/H	C/D	BRAIN LCD brightness setting. Range of 1 – 8
BEVFBR	MSG_INT	S	G/H	C/D	BRAIN EVF brightness setting. Range of 1 – 8
PLCDBR	MSG_INT	S	G/H	C/D	Rear LCD brightness setting. Range of 1 – 8
REVFBR	MSG_INT	S	G/H	C/D	Rear EVF brightness setting. Range of 1 – 8
FUIBR	MSG_INT	S	G/H	C/D	Switchblade brightness setting. Range of 1 – 8
IMGSPCRV	MSG_INT	S	G	C	Enable/Disable split monitor curves. 0=disable, 1=enable.
LCDCV	MSG_INT	S	G/H	C/D	Split curve output range for BRAIN LCD. (see monitor_curve_t)
HDMICV	MSG_INT	S	G/H	C/D	Split curve output range for HDMI.



HSDICV	MSG_INT	S	G/H	C/D	Split curve output range for BRAIN HSDI.
PLCDCV	MSG_INT	S	G/H	C/D	Split curve output range for rear LCD.
PHDS1CV	MSG_INT	S	G/H	C/D	Split curve output range for Pro-IO LCD.
PHDS2CV	MSG_INT	S	G/H	C/D	Split curve output range for Pro-IO LCD.
EVFCV	MSG_INT	S	G/H	C/D	Split curve output range for BRAIN EVF.
REVFCV	MSG_INT	S	G/H	C/D	Split curve output range for rear EVF.
ROTF0	MSG_INT	S	G	C	Enable/Disable rotation of video Look A. 1=enable, 0=disable
ROTF1	MSG_INT	S	G	C	Enable/Disable rotation of video Look B. 1=enable, 0=disable
VCODECG	MSG_INT	S	G/H	C/D	ProRes Codec video source (see monitor_feed_t)
VIDEOSRC	MSG_INT	S	G/H	C/D	Video source to Look A and Look B (see video_source_t)

EXAMPLE OF SELECTING OVERLAY

The SET message for the parameters BLCDOV, BHDIOV, BHDSIOV, PHDS1OV, PHDS2OV, PLCDOV, BEVFOV, and REVFOV takes a string to define the overlay selection. The string is the name of the overlay. The camera comes with several overlays already defined, but users can create new ones of their own naming convention. Therefore, always do a GET_LIST first to get the available choices. This list will have the form of a Clist with integer values and string pairs. However, rather than return the integer values in the SET message, the corresponding string is used instead.

Get the overlay list:

- ▶ #EXT:H:BLCDOV:
- ▶ #DRAGON:D:BLCDOV:0|4|2|3D Left (RED)|3|3D Right (RED)|0|None (RED)|1|Overview (RED)|4|Standard (RED)|:
- ▶ There are 5 choices and the current selection is Standard (RED)

Supposing the desire is to select None (RED). Where other parameters would normally require sending the value 0, this parameter will use the string "None (RED)"

- ▶ #EXT:S:BLCDOV:None (RED):

ALGORITHM GROUP

The ALGORITHM group provides control of the autofocus mode and zone type.

ALGORITHM GROUP PARAMETERS

NAME	VALUE TYPE	S	G	C	DESCRIPTION
AFMODE	MSG_INT	S	G/H	C/D	Auto-focus mode (see af_mode_t)
AFZONE	MSG_INT	S	G/H	C/D	Auto-focus zone type (see af_focuszone_t)
AFSIZE	MSG_INT	S	G/H	C/D	Auto-focus zone size. Small, medium, large. Get list at runtime. Changing size may change position if frame boundary is touched.
AFENHNCD	MSG_INT	S	G	C	Enable/disable enhanced AF. 0=disable, 1=enable
AFENHMON	MSG_INT	S	G/H	C/D	Auto-focus enhanced monitor selection. Get list at runtime



AFENCONS	MSG_INT	S	G/H	C/D	Auto-focus confirm style (see focus_assist_type_t)
AFRCKSEL	MSG_INT	S/U			Select the current rack point auto-focus window to be adjusted by the AFWPOS_ parameter (in rack mode, 0=A, 1=B, 2=C, 3=D)
AFWPOSX	MSG_INT	S/U	G	C	Position of current selected AF window center, X coordinate percent * 100 (range -10000 to +10000), 0% = center of frame. +/- 100% will put edge of window at edge of frame, range will scale from that center point to center of frame. Changing window size with AFSIZE will change scale.
AFWPOSY	MSG_INT	S/U	G	C	Position of AF window center, Y coordinate percent*100 (range -10000 to +10000), 0% = center of frame.
AFWPOSC	none	S			Re-focus at new position. Send this after changing position coordinates to final location.
HCFOCUS	MSG_UINT	S/U	G	C	Linear hand control signal for focus, range of 0 to 65535
HCIRIS	MSG_UINT	S/U	G	C	Linear hand control signal for iris, range of 0 to 65535
HCZOOM	MSG_UINT	S/U	G	C	Linear hand control signal for zoom, range of 0 to 65535
HCREINIT	MSG_UINT	S			Reinitialize the lens. Requires a value of 0. This should be called before using the HCFOCUS, HCIRIS, or HCZOOM to calibrate the end points.
AEMODE	MSG_INT	S	G/H	C/D	Auto-exposure mode setting (see ae_mode_t)
AEEVSHFT	MSG_INT	S/V	G/H	C/D	EV shift in EV * 10
AEMETERM	MSG_INT	S	G/H	C/D	Auto-exposure metering mode (see ae_metering_t)
AESPEED	MSG_INT	S	G/H	C/D	Auto-exposure algorithm speed (see ae_speed_t)
AESELECT	MSG_INT	S	G/H	C/D	Auto-exposure value selection. Constrain to use from list or allow any value (see ae_selection_t)
AEEXPRI	MSG_INT	S	G/H	C/D	Auto-exposure priority. Use this single parameter now instead of the separate aperture and exposure locks (see ae_exposure_priority_t)
AELOCKAP	MSG_INT	S	G	C	Deprecated as of RCP parameter set 6.10. Use AEEXPRI parameter instead.
AELOCKEX	MSG_INT	S	G	C	Deprecated as of RCP parameter set 6.10. Use AEEXPRI parameter instead.
AWBMODE	MSG_INT	S	G	C	Auto-White Balance mode (see awb_mode_t)