RAPIX AV Interfacing

Issue	2
Date	20 Oct 2022
Document	ICD-AV

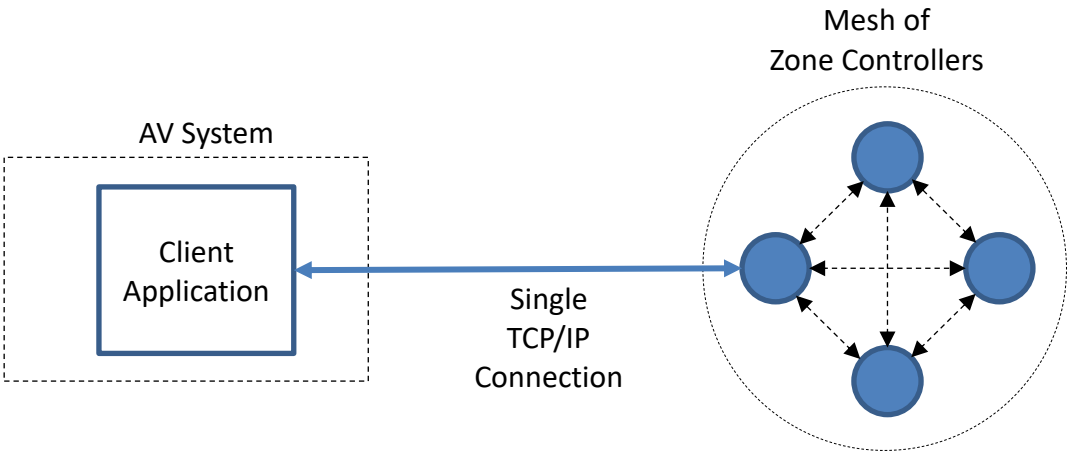
TABLE OF CONTENTS

1 INTRODUCTION

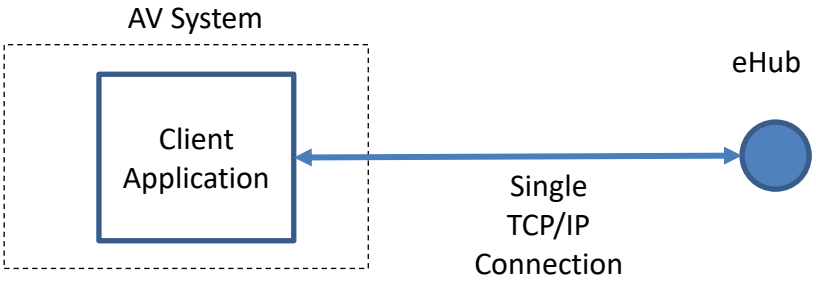
This document describes the process for controlling and monitoring a RAPIX system from a third-party system via the unencrypted interface to a Rapix Zone Controller or Rapix eHub. It focuses on the commands typically required for interfacing with an Audio/Visual (AV) system.

Zone Controllers

A single TCP/IP connection is required between the Client Application and the RAPIX lighting system as shown below. The connection can be made to any one of the Zone Controllers; they all contain the complete system status and exchange control and monitor messages as required.



eHub



Form of Messages

All messages within the RAPIX system use the DGCM (Distributed General Communication Message) format.

Third-party control uses the same DGCM message format, with either an encrypted or unencrypted communication channel.

This document describes only the subset of commands that can be used for unencrypted control and monitoring.

For details of the commands, refer to the Appendix.

Additional information about the use of other messages, encrypted connections and the DGCM Driver can be obtained on request from Ozuno.

2 USING THE RAPIX API

This section shows how to use the RAPIX API connection to a Zone Controller or eHub. This is useful to confirm your understanding of the RAPIX API prior to commencing the design of a DGCM Client.

2.1 RAPIX API Test Tool

To generate the text of a RAPIX API command using the RAPIX API Test Tool:

1. Open RAPIX Integrator.
2. Open your Project file (or create a new project).
3. Select **RAPIX API** from the **Tools** menu.
4. The **RAPIX API** form will be displayed.
5. Select the **Category** of command, the **Command** and the command parameters.
6. The DGCM command string will be displayed in the **RAPIX API Command** box.
7. To copy the command to the clipboard, click on the **Copy** button.

To test a RAPIX API command by sending it to a Zone Controller or eHub:

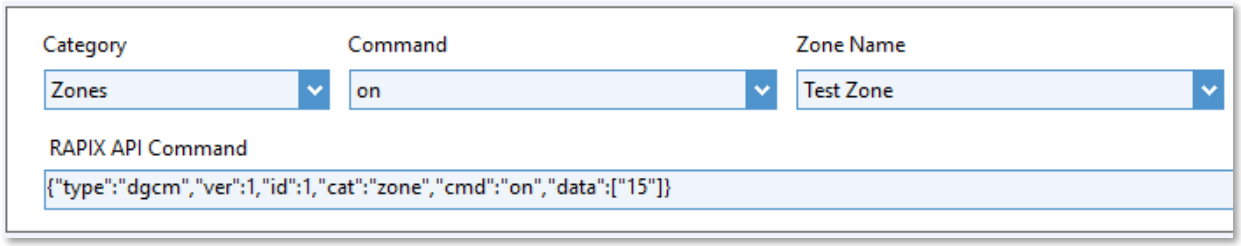
1. The Zone Controller or eHub needs to be configured to allow a connection from the computer you are using. Refer to section 2.3 for details.
2. Select the **IP Address** of the device (Zone Controller or eHub)
3. Click the **Open** button
4. Select the **Category** of command, the **Command** and the command parameters.
5. Click on the **Send** button
6. The text of the command and the reply from the device will be shown in the message log at the bottom of the form.

2.2 Examples

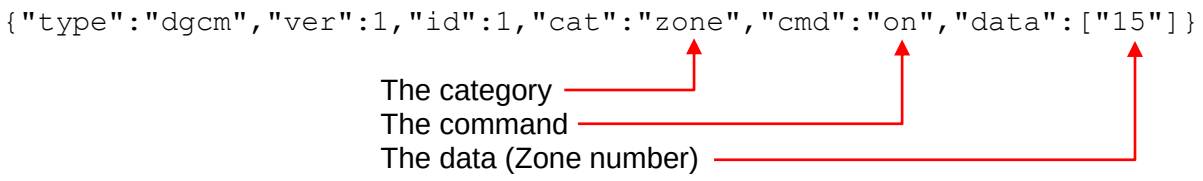
2.2.1 Turning a Zone On

For this example, we want to turn a Zone called "Test Zone" on. The steps are:

1. Open the RAPIX API tool (as above)
2. Select the **Zones** category
3. Select the **on** command
4. Select the Zone
5. The RAPIX API Command will be shown



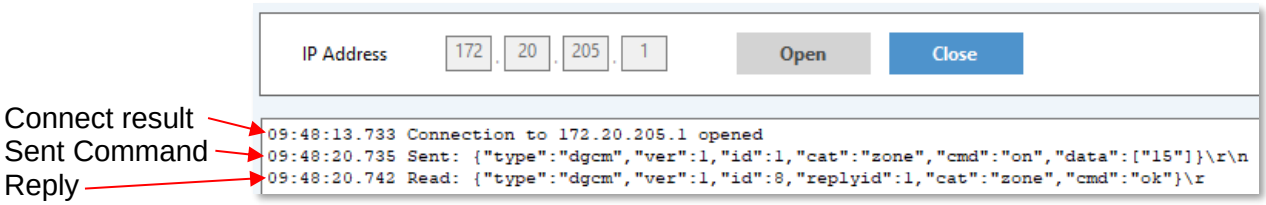
The generated command is shown below:



Additional details about the commands can be found in the Appendix section 3.5 if required.

To test this command, we will send it to a Zone Controller at IP Address 172.20.205.1:

1. Enter the **IP Address**
2. Click on **Open**
3. A message will be displayed in the log showing success or failure
4. Click on **Send**
5. The Zone is turned on and the log shows the sent command and the reply from the Zone Controller (in this case "ok").

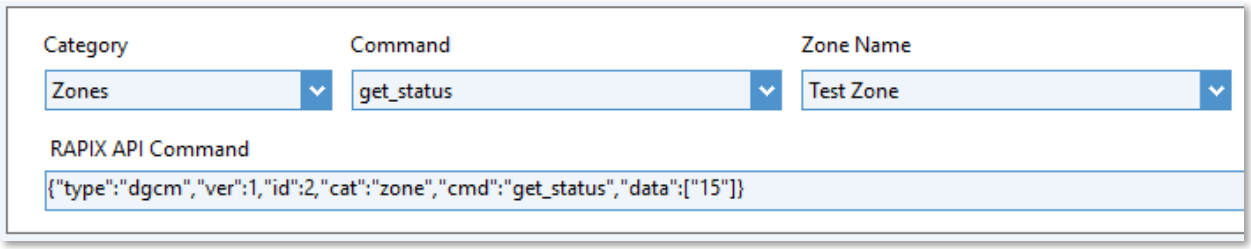


Note that the sent command has "\r\n" on the end of it. This is a carriage return character followed by a line feed (a.k.a. new line) character. **These are needed at the end of all commands sent.** Refer to the Appendix section 2.3 for details.

2.2.2 Reading a Zone Level

To read the level of a Zone:

1. Select the **Zones** category
2. Select the **get_status** command
3. Select the Zone
4. The RAPIX API Command will be shown



This time when we send the command, there will be a reply from the Zone Controller with data in it:


```
{"type":"dgcm","ver":1,"id":9,"replyid":2,"cat":"zone","cmd":"status","data":
                          ["15","238","0","254","254","0"]}\r
```


Zone number
Average Level
Min. Level
Max. Level
Target Level
Error status

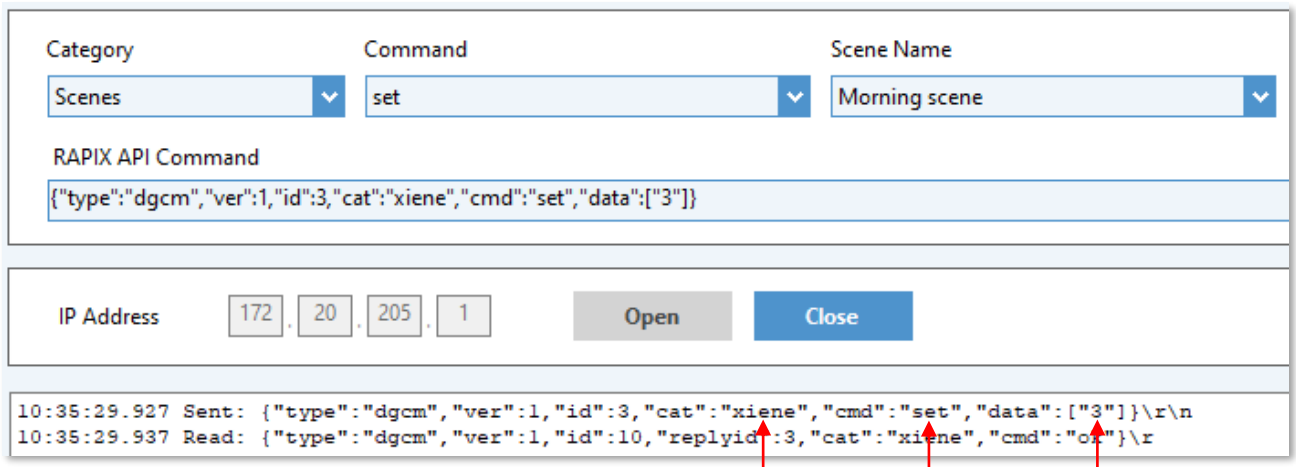
Refer to the Appendix section 3.4 for details about levels.

Note that an AV system should not regularly use the "zone get_status" command to read the level of Zones. It is much more efficient to send the "zone events_on" command. Once this has been done, a "zone status" command will be sent by the Zone Controller each time a Zone changes. Refer to the Appendix section 3.5 for details.

2.2.3 Setting a Scene

For this example, we want to set an RAPIX Scene called "Morning scene". The steps are:

1. Open the RAPIX API tool (as above)
2. Select the **Scenes** category
3. Select the **set** command
4. Select the Scene
5. The RAPIX API Command will be shown
6. If the Zone Controller is not connected, follow the steps above
7. Click on **Send**
8. The Scene will be set
9. The log will show the command sent and the reply

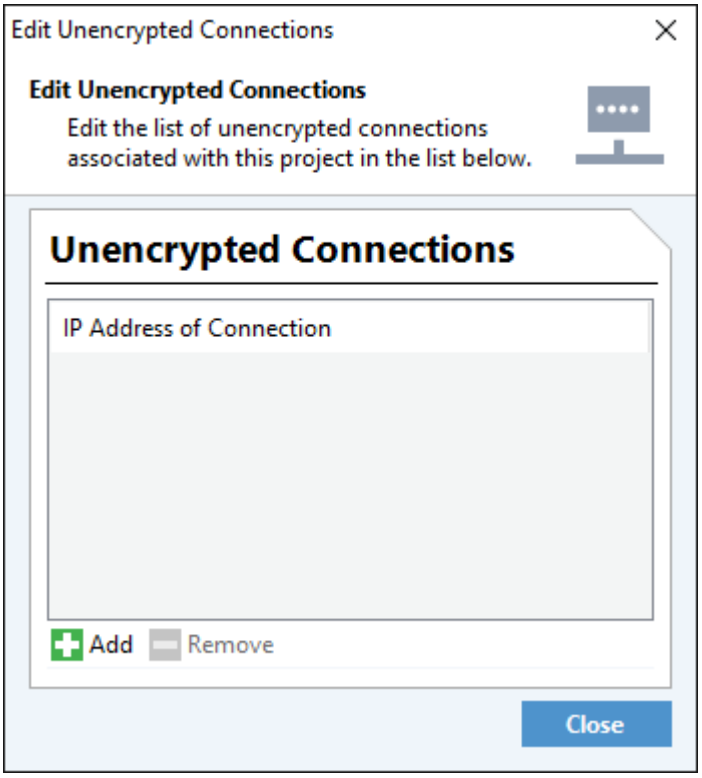


2.3 Connection to Zone Controller or eHub

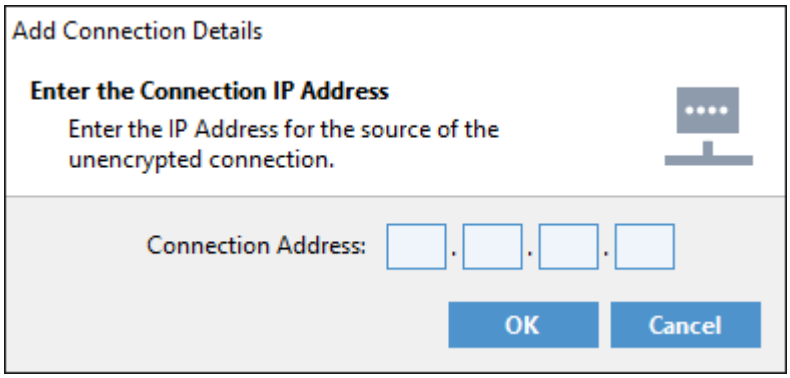
2.3.1 Enabling Third-Party Control: Zone Controller

To enable a third-party connection to the Zone Controllers:

1. Run RAPIX Integrator
2. Open the RAPIX Project for the site.
3. In the **Unencrypted Connections** box, click the **Edit Connection List** button
4. The **Unencrypted Connections** form will be shown:



5. Click the **Add** button. The **Add Connection Details** form will be shown:



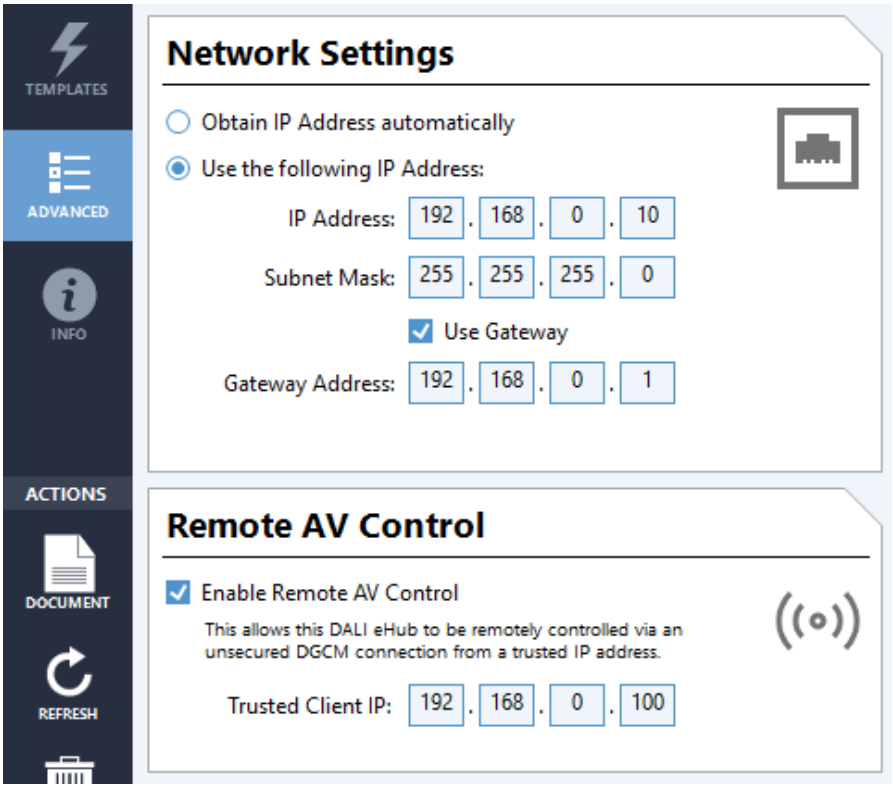
6. Enter the IP Address of the third-party device. Click on **OK**. Click on **Close**.

Transfer the project to the Zone Controllers.

2.3.2 Enabling Third-Party Control: eHub

To enable a third-party connection to the eHub:

1. Run RAPIX Integrator.
2. Open the RAPIX Project for the site.
3. Open the eHub configuration and select the Advanced tab.
4. Enter a static IP Address for the eHub.
5. Enable Remote AV Control.
6. Enter the IP Address of the device controlling the eHub.
7. Save the changes to the eHub.



2.3.3 TCP/IP Connection – Socket Details

A TCP/IP socket needs to be opened from the client application to any one of the Zone Controllers, or an eHub, on port 36689.

2.3.4 Sending Messages

All messages sent to the Zone Controller or eHub must be terminated with a Carriage Return (0x0D) and Line Feed (0x0A) in that order.

2.3.5 Receiving Messages

All messages sent by the Zone Controller will be terminated with a Carriage Return (0x0D).

More than one DGCM message may be sent in a single TCP/IP packet for efficiency. In this case, each will be separated by a carriage return.

2.3.6 Zone Controller – Initialisation of Connection

To enable Zone events (if required), send the **zone events** command:

```
{"type":"dgcm","ver":1,"id":1,"cat":"zone","cmd":"events","data":["on"]}
```

(followed by Carriage Return / Line Feed)

The reply will be a **zone ok** message:

```
{"type":"dgcm","ver":1,"id":123,"replyid":1,"cat":"zone","cmd":"ok"}
```

(followed by a Carriage Return).

To set the Zone scaling to be from 0 to 255 (instead of the default 0 to 254), send a **zone set_zone_range** command:

```
{"type":"dgcm","ver":1,"id":2,"cat":"zone","cmd":"set_zone_range","data":["
0","255","0"]}
```

The reply will be a **zone ok** message.

To enable RAPIX Scene events (if required), send the **xiene events** command:

```
{"type":"dgcm","ver":1,"id":1,"cat":"xiene","cmd":"events","data":["on"]}
```

(followed by Carriage Return / Line Feed)

The reply will be a **xiene ok** message:

```
{"type":"dgcm","ver":1,"id":123,"replyid":1,"cat":"xiene","cmd":"ok"}
```

(followed by a Carriage Return).

2.3.7 eHub – Initialisation of Connection

The eHub does not support scaling or events. There is no special initialisation needed after making a connection.

3 APPENDIX

3.1 Abbreviations

API	Application Programming Interface
AV	Audio/Visual
DALI	Digital Addressable Lighting Interface
DGCM	Distributed General Communication Messages
ICD	Interface Control Document
JSON	JavaScript Object Notation
LSB	Least Significant Byte
MSB	Most Significant Byte
TCP/IP	Transmission Control Protocol/Internet Protocol

3.2 Document Conventions

The names of protocol fields, category names and command names are in **bold** to distinguish them from other uses of the words.

Hexadecimal numbers are expressed in C notation (e.g. 0xFF).

3.3 Encrypted vs Trusted Connections

The Zone Controller and eHub support a substantial command set using an encrypted connection.

Both products also support a command subset using an unencrypted trusted channel. This channel is trusted in the sense that the Zone Controller or eHub require that the IP address of the device issuing the commands be set in the product configuration.

This means that the Zone Controller or eHub “trusts” the IP address making the incoming connection, and allow the subset of commands defined in this document.

Until the Zone Controller or eHub has that trusted IP address set, no trusted connections are accepted.

3.4 Message Structure

All DGCM messages use the JSON format, and consist of:

- Message Header
  - Type
  - Version
  - Source (for messages from a Zone Controller only)
- Message Identifier
- Reply identifier (for reply messages only)
- Message Category
- Message Command
- Message Data

3.4.1 Example

A typical message would be one to switch on Zone with UID 5, as follows:

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "zone",
  "cmd": "on",
  "data": ["5"]
}
```

3.4.2 Notes

White Space

White space in JSON is ignored. The above message example is equivalent to:

```
{"type":"dgcm","ver":1,"id":123,"cat":"zone","cmd":"on","data":[
"5"]}
```

The JSON parser ignores white space and hence accepts a message with redundant white space.

In the interest of minimising message size, it is recommended that messages be sent without additional white space.

Capitalisation

Capitalisation of message details should be as per this document.

In general, the parameter names (**type**, **ver**, **id**, **replyid**, **cat**, **cmd**, **data**), categories (e.g. **zone**) and commands (e.g. **on**) are lower case. Text data can be any case.

Case is significant in the parameter names – case mismatches will cause commands to be rejected.

Order

The order of the parameters shall be as listed in this document. A good JSON deserializer will handle parameters in any order, but it is helpful for simple devices to have consistency.

Note that the eHub requires the parameters to be in the correct order.

Quotation Marks

Note that the quotation characters are the ASCII 0x22 character. In this document, they may be inadvertently shown using quotation marks with different Unicode values.

The text within a data string needs to be “escaped”. If a double-quote character " is used within a string, it needs to be replaced with \". Similarly, a back-slash character \ within a string needs to be replaced by \\.

Standard JSON serializers and deserializers will handle this automatically.

Text Encoding

All text is encoded as UTF-8. This allows for support of any languages, but is efficient for English (ASCII) characters.

3.4.3 Message Parameters

Type

This is fixed at “**dgcm**” to minimise the possibility of confusion between different types of JSON messages.

Version

The message version (**ver**) refers to the format of a generic message.

If the structure of the headers (not the contents of the **data** section) is changed, the version number will change. A device should reject messages with a different version number.

Only version 1 is currently defined and is the subject of this document.

Source

The message **source** is the Controller id, in the case of a message being sent by a Zone Controller. Other devices should not include this parameter, but need to be able to process a message with a **source**.

The intent of the **source** parameter is so that a controller can broadcast a message and then ignore the received echo of the message. It is also useful for logging purposes to know where messages came from.

The eHub will not accept a command containing **source**.

Id

The **id** of a message is primarily used to match replies to messages. A message **id** cannot be reused too often because replies could get mismatched. The value shall be between 0x00000000 and 0x7FFFFFFF to allow the use of signed 32 bit numbers (with roll-over).

Reply Id

The reply id (**replyid**) must match the **id** of the message that it is replying to. The **replyid** field should only be included for a message which is a reply.

Category

The message category (**cat**) identifies the class of commands. It allows a device to quickly recognise whether it is interested in the message.

Command

The message command (**cmd**) identifies the particular command. It does not have to be unique across all categories. So, for example, there could be a **find** command within both the **discovery** and **database** categories.

Data

If there is no **data** defined for the message, then the **data** section should not be included.

Parameters within the **data** section are an array of strings. The order is important. If a parameter is unknown, null or blank, then it should be included as a blank string “”.

The **data** is an array of strings and must include the square brackets, even if there is only a single string.

Data fields that are numbers shall be encoded as strings, for example:

“data”:[“1”,”2”,”3”]

3.4.4 Levels

The default Zone Levels are an 8-bit number based on the DALI levels as listed below:

Level	Meaning
0	Off
1	On, Minimum
2 - 253	On
254	On, Maximum
255	Unused

Note that the Zone level scaling can be changed through the use of the Set Range command.

3.4.5 Fade Times

Available fade times are listed in the table below.

The bold values (0 – 90.5 sec) are standard DALI Fade Times. The others are extended fade times, generated by the RAPIX Devices. These can require many DALI messages, so standard DALI fade times should be used if suitable.

The Zone Controller messages use fade times in seconds. Any fade time may be requested, but the closest available fade time will actually be used.

The eHub uses an index to select the fade time. The eHub Fade Time coding is shown below.

Fade Time Code	Fade Time
0	**Instant**
1	**700 ms**
2	**1 s**
3	**1.4 s**
4	**2 s**
5	**2.8 s**
6	**4 s**
7	**5.7 s**
8	**8 s**
9	**11.3 s**
10	**16 s**
11	**22.6 s**
12	**32 s**
13	**45.3 s**
14	**64 s**
15	**90.5 s**
16	2 min
17	3 min
18	4 min
19	5 min
20	6 min
21	7 min

Fade Time Code	Fade Time
22	8 min
23	9 min
24	10 min
25	15 min
26	20 min
27	25 min
28	30 min
29	35 min
30	40 min
31	45 min
32	50 min
33	55 min
34	60 min
35	90 min
36	2 hr
37	3 hr
38	4 hr
39	6 hr
40	8 hr
41	12 hr
42	18 hr
43	24 hr

3.4.6 Colours

DALI Type 8 devices support four colour types:

- Colour Temperature
- RGB(WAF)
- XY
- Primary N

RAPIX currently supports colour temperature, RGB / RGBWAF and XY.

DALI-2 will remove the Primary N colour type, so RAPIX has no plans to support this.

Colour values in DGCM messages are hexadecimal strings, where the bytes are as follows:

Colour Type	MS Byte 7	6	5	4	3	2	1	LS Byte 0
Colour Temperature	Unused	01	Unused	Unused	Unused	Unused	MSB	LSB
RGB(WAF)	Unused	03	Red	Green	Blue	White	Amber	Free
XY	Unused	04	Unused	Unused	X MSB	X LSB	Y MSB	Y LSB
MASK	00	FF	FF	FF	FF	FF	FF	FF

Note: for the RGB(WAF) colour components, the values are 0 – 254 (0x00 – 0xFE), with 255 (0xFF) being "MASK" (no change).

Note: for the XY colour components, the values are 0 – 65534 (0x0000 – 0xFFFE), with 65535 (0xFFFF) being "MASK" (no change).

Examples:

Colour temperature 3000K (0x0BB8): `"0x01000000000BB8"`

RGB value #FE0000 (i.e. red): `"0x03FE0000FFFFFF"`

RGBWAF value #00FE00 (i.e. green): `"0x0300FE00000000"`

RGBWAF amber value #7F: `"0x03000000007F00"`

XY value X = 0x8000, Y = 0x4000 `"0x04000080004000"`

MASK (no colour change): `"0xFFFFFFFFFFFFFF"` or `""`

Note: the two leading zeros (i.e. the MSB) are optional. For example, "0x**00**01000000000BB8" and "0x01000000000BB8" are treated as identical.

3.5 Message Definition

3.5.1 Acknowledgement

Acknowledgement messages have the same category as the message they are acknowledging.

Acknowledgement messages have a reply id matching the message they are acknowledging.

3.5.1.1 Success

If a message succeeds, there may be a reply with a command of **ok**.

3.5.1.2 Failure

If a message fails, there will be a reply with a command of **error**. Additional details of the error may be in the data section.

If a message is not recognised by the Zone Controller, it will reply with a failure message.

If a message is not recognised by the eHub, it will usually reply with a failure message. In some cases though, it will not reply at all and may close the connection. This is a security measure, and typically happens when trying to execute a command that requires an encrypted connection.

3.5.2 Zone Messages

3.5.2.1 Zone On

Behaviour	Switch a Zone on (to the maximum level).	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **zone** Command: **on** Data: Zone Id	
Notes This is equivalent to a **zone fade_to_level** command with level = 254 and fade time = 0.		
Example Turn on Zone with ID 5. `{ "type": "dgcm", "ver": 1, "id": 123, "cat": "zone", "cmd": "on", "data": ["5"] }`		

3.5.2.2 Zone Off

Behaviour	Switch a Zone off.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **zone** Command: **off** Data: Zone Id	
Notes This is equivalent to a **zone fade_to_level** command with level = 0 and fade time = 0.		
Example Turn off Zone with ID 5. `{ "type": "dgcm", "ver": 1, "id": 123, "cat": "zone", "cmd": "off", "data": ["5"] }`		

3.5.2.3 Zone Fade to Level

Behaviour	The zone level will fade to the new level.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **zone** Command: **fade_to_level** Data: Zone Id Level Fade Time	

Notes

Zone Controller:

If the Fade Time is blank, the fade will use the current fade time. Fade times are in seconds.

eHub:

Fade time must be present, and is represented as a code number.

Code numbers are shown in section 3.4.

Example

Fade Zone with ID 5 to level 182 at speed 6. On a Zone Controller, this would be over 6 seconds. On an eHub, code 6 corresponds to fading over 4 seconds.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "zone",
  "cmd": "fade_to_level",
  "data": ["5","182","6"]
}
```

3.5.2.4 Zone Fade to Colour

Behaviour	The zone level will fade to the new level and colour.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **fade_to_color** Data: Zone Id Level Colour Fade Time	

Notes

Zone Controller:

Fade times are in seconds.

If Level is 255 ("MASK") or blank, then there will be no level change, just a colour change.

If Colour is an empty string "", then there will be no colour change, just a level change. In this case, it is more suitable to use the **fade_to_level** command.

Example

Fade Zone with ID 5 to level 182, colour temperature 3000K over 0.7 seconds.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "zone",
  "cmd": "fade_to_color",
  "data": ["5","182","0x01000000000BB8","0.7"]
}
```

Set Zone with ID 5 to colour temperature 3000K immediately, with no level change.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "zone",
  "cmd": "fade_to_color",
  "data": ["5","","0x01000000000BB8","0"]
}
```

3.5.2.5 Zone Nudge / Offset

Behaviour	The zone level will changed up or down by an amount.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **zone** Command: **nudge** Data: Zone Id Offset Fade Time	

Notes

A Zone must be ON for a nudge up to do anything.

A nudge down will not nudge below the minimum ON level (that is, a nudge down will never result in a zone being turned off).

Offset is a signed number – use negative numbers to nudge downwards.

Zone Controller:

Fade times are in seconds.

eHub:

The eHub will always do a nudge quickly (not instantly).

Fade time can be omitted.

If the fade time is present, it will be ignored.

Example

Nudge Zone with ID 5 up by 20 over 2 seconds. On an eHub the nudge would be done quickly (< 1 second) and the fade time is ignored.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "zone",
  "cmd": "nudge",
  "data": ["5","20","2"]
}
```


3.5.2.6 Stop Fade

Behaviour	Stop a running fade.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **zone** Command: **stop_fade** Data: Zone Id	
Notes	This has no affect if the zone is not currently performing a fade. The zone level will stop wherever it is at the time the command is received.	

Example	Stop fading on Zone with ID 5.
	
	`{`
	`  "type": "dgcm",`
	`  "ver": 1,`
	`  "id": 123,`
	`  "cat": "zone",`
	`  "cmd": "stop_fade",`
	`  "data": ["5"]`
	`}`	

3.5.2.7 Get Status

Behaviour	Get the status of one or more Zones.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **get_status** Data: Zone Id	
Notes Not supported by eHub. A status message will be generated as the reply. If the Zone Id is “all”, then a series of **status** messages will be returned. Refer to section **Error! Reference source not found.** for the Zone Status message that is returned. ***The status of the Zone should not be read repeatedly (i.e. polled). To get updates when a Zone changes, the Zone events should be enabled.***		

Example
Get the status of Zone with ID 5.
 	
`{`
`  "type": "dgcm",`
`  "ver": 1,`
`  "id": 123,`
`  "cat": "zone",`
`  "cmd": "get_status",`
`  "data": ["5"]`
`}`		

3.5.2.7.1 Status

Behaviour	Sent by Zone Controller to report the state of a Zone.	
Originator	Zone Controller	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **status** Data: Zone Id Average Level Minimum Level Maximum Level Target Level Error Condition	

Notes

Not supported by eHub.

Additional information about Zone Level inference follows.

Error definition follows.

Zone Controller:

Sent

- When the levels of a Zone change (if zone events are on), or
- When the error condition of a Zone changes (if zone events are on), or
- in reply to a get_status message.

In the case where this message is unsolicited, because of being emitted with zone events enabled, the “reply_id” portion of the message will not be present (as the message is not a reply).

Example

Zone with ID 5 is reported with Average, Min, Max levels 190 (i.e. they are all consistent) and a Target Level 254. There are no errors.

```
{
  "type": "dgcm",
  "ver": 1,
  "source": 92,
  "id": 456,
  "reply_id": 123,
  "cat": "zone",
  "cmd": "status",
  "data": ["5","190","190","190","254","0"]
}
```

If zone extended status format has been enabled (see section 3.5.2.9), then the zone extended_status message will be sent by the zone controller instead of the zone status message:

Behaviour	Sent by Zone Controller to report the state of a Zone.	
Originator	Zone Controller	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **extended_status** Data: Zone Id Average Level Minimum Level Maximum Level Target Level Average Colour Target Colour Reserved (blank) Reserved (blank) Error Condition	
Notes As for zone status message		

Example
Zone with ID 5 is reported with
• Average, Min, Max Level = 127
• Target Level = 254.
• Average colour temperature = 3000K
• Target colour temperature = 5000K
• No errors.
 	
`{`
`"type": "dgcm",`
`"ver": 1,`
`"source": 92,`
`"id": 456,`
`"reply_id": 123,`
`"cat": "zone",`
`"cmd": "extended_status",`
`"data": ["5","127","127","127","254",`
`"0x01000000000BB8","0x01000000001388",`
`"","","0"]`
`}`		

Note that additional information can be inferred about the Zone from the minimum and maximum levels:

Minimum Level	Maximum Level	Inferred information about Zone
0	0	All devices are off
> 0	> 0	All devices are on
0	Any	Some (maybe all) devices are off
Any	> 0	Some (maybe all) devices are on
0	> 0	Some devices are on and some are off
Same level		All devices are at the same level

A bit-field is used to allow multiple conditions to be reported in the Error Condition:

Bit-field Value	Name	Meaning
0x00	OK	Everything in the Zone is OK
0x01	LEVEL UNKNOWN	The level of some or all devices in the zone is unknown. This is not necessarily an error.
0x02	LAMP FAILURE	One or more devices has a lamp failure
0x04	DEVICE_FAILURE	One or more devices has an internal failure
0x08	DEVICE MISSING	One or more devices in the Zone are not responding (but the DALI Line is OK)
0x10	DALI LINE FAILURE	One or more DALI Lines (which are part of the Zone) have a communication failure
0x20	ZONE CONTROLLER COMMS FAILURE	One or more Zone Controllers (which are part of the Zone) is not communicating
0x40	EM FAILURE	One or more Emergency Devices has a failure
0x80	DEVICE_CONFLICT	One or more addresses has multiple devices
Others	undefined	Do not use

The simplest way to use the Error Condition value is:

1. If the value is 0, all is OK
2. If the least significant bit is set, then the Zone Level is unknown / uncertain
3. If any of the other bits are set, there is a failure

3.5.2.8 Zone Events

Behaviour	Enable or Disable the reporting of Zone Status whenever the level of a Zone changes, or the error condition of the Zone changes.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **events** Data: state, as either “on” or “off”	

Notes

Not supported by eHub.

Any change of level ***of any Zone*** will cause a Zone Status message to be sent by the Zone Controller.

Any change of error status ***of any Zone*** will cause a Zone Status messages to be sent by the Zone Controller.

If extended zone format is enabled, then Zone Extended Status messages will be sent by the Zone Controller.

Refer to section 3.5.2.9 for Extended Zone Status Format.

Refer to section **Error! Reference source not found.** for the Zone Status message.

Example

Enable Zone Events.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "zone",
  "cmd": "events",
  "data": ["on"]
}
```


3.5.2.9 Extended Zone Status Format

Behaviour	Enable or Disable extended zone status format.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **extended** Data: state, as either “on” or “off”	
Notes Not supported by eHub. When enabled, the **zone extended_status** messages are sent by the Zone Controller instead of the **zone status** messages. ***The Zone Events also need to be enabled to receive extended events.*** Refer to section **Error! Reference source not found.** for the Zone Extended Status message.		

Example
Enable Zone Events, then enable Extended Zone Status format.
`{ "type": "dgcm", "ver": 1, "id": 123, "cat": "zone", "cmd": "events", "data": ["on"] }`
`{ "type": "dgcm", "ver": 1, "id": 124, "cat": "zone", "cmd": "extended", "data": ["on"] }`		

3.5.2.10 Change or Set Zone Range

Behaviour	Set the control and reporting range of levels used by Zones.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **zone** Command: **set_zone_range** Data: Minimum value Maximum value Number of decimal places	
Notes Not supported by eHub. The default range of Zone levels is 0 – 254. It is possible to override this range to suit the range of a client. For example, if the range is set to 0 – 100 with 1 decimal place, then if the zone is at level 127, the level will be reported as “50.0”. The Zone Range that is set applies to all zones.		

Example
Set the Zone Range to have a minimum of 0, maximum 100, and 1 decimal place.
 	
`{`
`  "type": "dgcm",`
`  "ver": 1,`
`  "id": 123,`
`  "cat": "zone",`
`  "cmd": "set_zone_range",`
`  "data": ["0","100","1"]`
`}`		

3.5.3 RAPIX Scene Messages

3.5.3.1 Set an RAPIX Scene

Behaviour	Set the RAPIX Scene, making it active and setting its constituent levels in its Zones.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **xiene** Command: **set** Data: RAPIX Scene ID	
Notes		
Example		

Set the RAPIX Scene with ID 4321.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xiene",
  "cmd": "set",
  "data": ["4321"]
}
```

3.5.3.2 Off

Behaviour	Turn off RAPIX Scene. It is made inactive, and all constituent Zones are turned off.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **xiene** Command: **off** Data: RAPIX Scene ID	
Notes		
Example		

Turn off RAPIX Scene with ID 4321.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xiene",
  "cmd": "off",
  "data": ["4321"]
}
```


3.5.3.3 Nudge

Behaviour	Adjust the level of all constituents of an RAPIX Scene.	
Originator	Client	
Products	☒ Zone Controller	☒ eHub
DGCM	Category: **xiene** Command: **nudge** Data: RAPIX Scene ID Amount Fade Time (seconds)	

Notes

A positive adjustment will increase the level of the loads in the RAPIX Scene relative to their current level. A negative adjustment will decrease the levels.

Repeatedly nudging a RAPIX Scene will change the levels until they reach their maximum (for a positive nudge) or minimum (for a negative nudge) levels.

Zone Controller:

The adjustment amount is scaled for the Client range.

eHub:

The fade time is not used. If present, it will be ignored. The eHub always performs a fast nudge (< 1 second).

Example

Nudge RAPIX Scene with ID 4321 up by 20, over 1 second.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xiene",
  "cmd": "nudge",
  "data": ["4321","20","1"]
}
```

3.5.3.4 Offset

Behaviour	Apply a fixed offset to the level of all constituents of an RAPIX Scene.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xiene** Command: **offset** Data: RAPIX Scene ID Amount Fade Time (seconds)	

Notes

A positive adjustment will increase the level of the loads in the RAPIX Scene relative to their current level. A negative adjustment will decrease the levels.

Applying an offset to a RAPIX Scene will change the levels of all members of the RAPIX Scene. Applying the same offset again will have no effect, since the levels will already have had the offset applied.

Zone Controller:

The adjustment amount is scaled for the Client range.

eHub:

This command is not supported.

Example

Set an offset to RAPIX Scene with ID 4321. The offset is to be 30, applied over 1 second.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xiene",
  "cmd": "offset",
  "data": ["4321","30","1"]
}
```

3.5.3.5 Get Status

Behaviour	Get the status of an RAPIX Scene.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xiene** Command: **get_status** Data: RAPIX Scene ID	

Notes

A reply will be generated that is a **xiene status**. See section 3.5.3.5.1 for details.

Zone Controller:

If the RAPIX Scene ID is “all” then a series of **xiene status** messages will be sent, one per RAPIX Scene.

eHub:

This command is not supported.

The status of the RAPIX Scene should not be read repeatedly (i.e. polled). To get updates when an RAPIX Scene changes, the RAPIX Scene events should be enabled.

Example

Get the status of RAPIX Scene with ID 4321.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xiene",
  "cmd": "get_status",
  "data": ["4321"]
}
```

3.5.3.5.1 Status

Behaviour	The status of an RAPIX Scene is reported.	
Originator	Zone Controller	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xiene** Command: **status** Data: Xiene Id Status Offset	

Notes

The status value is true (1) if the RAPIX Scene is set, otherwise it is false (0).

If the RAPIX Scene is set and has been adjusted, then the offset is the amount that is has been changed from the default.

If the RAPIX Scene is not set, the offset will be 0 and should be ignored.

Zone Controller:

Sent in response to a **xiene get_status** message, or for RAPIX Scene events.

The offset is in the range set by the client.

eHub:

Not supported.

Example

RAPIX Scene with ID 4321 is set, and has an offset of 20 applied.

```
{
  "type": "dgcm",
  "ver": 1,
  "source": 92,
  "id": 456,
  "reply_id": 123,
  "cat": "xiene",
  "cmd": "status",
  "data": ["4321","1","20"]
}
```

3.5.3.6 RAPIX Scene Events

Behaviour	Get an RAPIX Scene status whenever the level in a Zone constituent of an RAPIX Scene changes.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xiene** Command: **events** Data: "on" or "off"	
Notes A **xiene status** will be sent whenever any zone that is a member of an RAPIX Scene has a change of level. See section 3.5.3.5.1 for details. eHub: This command is not supported.		

Example
Turn on RAPIX Scene events.
 	
`{`
`  "type": "dgcm",`
`  "ver": 1,`
`  "id": 123,`
`  "cat": "xiene",`
`  "cmd": "events",`
`  "data": ["on"]`
`}`		

3.5.4 Operating Property Messages

An Operating Property is essentially an enumeration. Each Operating Property has an id (16 bit), a name and a value (8 bit). The Operating Properties and values can be named.

3.5.4.1 Set an Operating Property

Behaviour	Set the Operating Property value.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_op_prop** Command: **set** Data: Operating Property Number (0 – 65534) Operating Property Value (0 – 255)	
Notes The Operating Property value will be broadcast to all DALI Lines where there is an Xi device that uses the Operating Property.		
Example Set the Operating Property with ID 4321 to a value of 2: `{ "type": "dgcm", "ver": 1, "id": 123, "cat": "xi_op_prop", "cmd": "set", "data": ["4321","2"] }`		

3.5.4.2 Get Status

Behaviour	Get the Operating Property value.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_op_prop** Command: **get_status** Data: Operating Property Number (0 – 65534) or "all"	

Notes

The reply will be an **xi_op_prop status** message (see section 3.5.4.2.1).

If the data is "all", then the status of all Operating Properties will be returned.

The status of the Operating Property should not be read repeatedly (i.e. polled). To get updates when an Operating Property changes, the Operating Property Events should be enabled.

Example

Get the status of Operating Property with ID 4321:

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xi_op_prop",
  "cmd": "get_status",
  "data": ["4321"]
}
```

3.5.4.2.1 Status

Behaviour	The status of an Operating Property.	
Originator	Zone Controller	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_op_prop** Command: **status** Data: Operating Property Id Value	

Notes

This is a reply to the **xi_op_prop get_status** message.

If the value is blank, then its value is unknown.

If "all" Operating Properties were requested, then the data will be:

Operating Property 1 Id

Operating Property 1 Value

Operating Property 2 Id

Operating Property 2 Value

…

Example

Operating Property with ID 4321 has a value of 20.

```
{
  "type": "dgcm",
  "ver": 1,
  "source": 92,
  "id": 456,
  "reply_id": 123,
  "cat": "xi_op_prop",
  "cmd": "status",
  "data": ["4321","20"]
}
```

3.5.4.3 Operating Property Events

Behaviour	Enable or Disable the reporting of Operating Property Status whenever the value of an Operating Property changes.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_op_prop** Command: **events** Data: state, as either “on” or “off”	

Notes

Not supported by eHub.

Any change of state ***of any*** Operating Property will cause an Operating Property Status message to be sent by the Zone Controller.

Refer to section 3.5.4.2.1 for the Operating Property Status message.

Example

Enable Operating Property Events.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xi_op_prop",
  "cmd": "events",
  "data": ["on"]
}
```

3.5.5 Flag Messages

A Flag is used to enable or disable some system functionality. Flags are organised in a hierarchy of Flag Groups and Flags. A Flag can be in one of two states; set or not set (cleared).

3.5.5.1 Set a Flag State

Behaviour	Set a Flag state.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_flag** Command: **set** Data: Flag Group (0 – 254) Flag (0 – 255) State (0 = not set, 1 = set)	

Notes

The Flag state will be broadcast to all DALI Lines where there is an Xi device that uses the Flag.

Example

Set the Flag Group 3, Flag 5 to a value of 2:

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xi_flag",
  "cmd": "set",
  "data": ["3","5","2"]
}
```

3.5.5.2 Get Status

Behaviour	Get a Flag state.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_flag** Command: **get_status** Data: Flag Group (0 – 254) or "all" Flag (0 – 255) or "all"	

Notes

The reply will be an **xi_flag status** message (see section 3.5.5.2.1).

If the Flag Group data is "all", then the status of all Flag Groups and their Flags will be returned.

If the Flag data is "all", then the status of all Flags in the specified Flag Group will be returned.

The status of the Flag should not be read repeatedly (i.e. polled). To get updates when a Flag changes, the Flag events should be enabled.

Example

Get the status of Flag Group 3, Flag 5:

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xi_flag",
  "cmd": "get_status",
  "data": ["3","5"]
}
```

Get the status of all Flags:

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "xi_flag",
  "cmd": "get_status",
  "data": ["all","all"]
}
```

3.5.5.2.1 Status

Behaviour	The status of one or more Flags.	
Originator	Zone Controller	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_flag** Command: **status** Data: Flag Group Id Flag Id State (0 = not set, 1 = set, empty = unknown)	

Notes

This is a reply to the **xi_flag get_status** message.

If the state is blank, then it is unknown.

If multiple Flags have been requested, they will be listed in the data as follows:

Flag Group 1 Id
Flag 1 Id
State
Flag Group 1 Id
Flag 2 Id
State
…
Flag Group 2 Id
Flag 1 Id
State
Flag Group 2 Id
Flag 2 Id
State
…

Example

Flag Group 2, Flag 3 is set:

```
{
  "type": "dgcm",
  "ver": 1,
  "source": 92,
  "id": 456,
  "reply_id": 123,
  "cat": "xi_flag",
  "cmd": "status",
  "data": ["2","3","1"]
}
```

3.5.5.3 Flag Events

Behaviour	Enable or Disable the reporting of Flag Status whenever the state of a Flag changes.	
Originator	Client	
Products	☒ Zone Controller	☐ eHub
DGCM	Category: **xi_flag** Command: **events** Data: state, as either “on” or “off”	
Notes Not supported by eHub. Any change of state ***of any Flag*** will cause a Flag Status message to be sent by the Zone Controller. Refer to section 3.5.5.2.1 for the Flag Status message.		

Example
Enable Flag Events.
 	
`{`
`  "type": "dgcm",`
`  "ver": 1,`
`  "id": 123,`
`  "cat": "xi_flag",`
`  "cmd": "events",`
`  "data": ["on"]`
`}`		

3.5.6 Remote Operations

Remote operations are supported only by the eHub. These provide an alternative, low level means of providing control by simulating an operation that would be performed by a human operator.

These commands apply only to the gangs and channels of input devices (buttons, knobs, touch switches) that have already been configured on the eHub; the commands simulate that operation of those input devices.

3.5.6.1 Remote Operate

Behaviour	Remotely activate a gang on an eHub Channel.	
Originator	Client	
Products	☐ Zone Controller	☒ eHub
DGCM	Category: **remote** Command: **operate** Data: Channel Gang Operation Code	

Notes

Zone Controller:

This command is not supported.

eHub:

Use of an invalid (non-existent) channel or gang will result in an error.

Channel and Gang are 1-based.

Permitted operation codes are:

00 = Null, no operation
01 = Gang was tapped (short press / release)
02 = Gang has been pressed and is being held (press & hold)
03 = Gang was released from a preceding press & hold
04 = Gang was double tapped

Example

Double tap gang 7 on channel 1.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "remote",
  "cmd": "operate",
  "data": ["1","7","4"]
}
```

3.5.6.2 Remote Level

Behaviour	Remotely apply a level or level change to a gang on a channel.	
Originator	Client	
Products	☐ Zone Controller	☒ eHub
DGCM	Category: **remote** Command: **level** Data: Channel Gang Type Amount	

Notes

Zone Controller:

This command is not supported.

eHub:

Use of an invalid (non-existent) channel or gang will result in an error.

Channel and Gang are 1-based.

Type = 0 means the amount is an absolute amount to be applied to a Zone.

Type = 1 means the amount is a relative change (up or down) similar to that applied by Rotary Peripherals.

Applying an absolute amount to a Xiene is not defined and will be ignored.

Amount is either the signed change, or absolute amount, as determined by Type.

Example

Apply a level of 182 to the Zone controlled by gang 7 on channel 1.

```
{
  "type": "dgcm",
  "ver": 1,
  "id": 123,
  "cat": "remote",
  "cmd": "level",
  "data": ["1","7","1","182"]
}
```