External Control Protocol (JSON) Reference Manual

020-102837-16

Terra

CHRISTIE

Revision History

The following table summarizes the changes in each revision. Refer to the command details for more details.

<table>
<tr><th>Date</th><th>Version</th><th>Revision</th><th>Description</th></tr>
<tr><td>04/30/18</td><td>1</td><td>1</td><td>First release.</td></tr>
<tr><td>9/12/18</td><td>2</td><td>1</td><td>R1.0.4. Changed the RS-232 baud rate to 9600.</td></tr>
<tr><td>10/25/18</td><td>3</td><td>1</td><td>Added the following commands:
GetLayoutByName, SwitchVideo, SwitchAudio</td></tr>
<tr><td>11/1/18</td><td>4</td><td>1</td><td>Added a section on how to enable access to Terra Manager using JSON APIs (refer to Enable Access and Device Configuration page 13).</td></tr>
<tr><td>11/8/18</td><td>5</td><td>1</td><td>Added the following command:
ApplyLayoutByName
Updated the following commands:
<table>
<tr><td>GetLayoutsByRange
GetDeviceByID
GetDeviceByRange</td><td>Changed the response parameters from ID and Name to id and name.</td></tr>
<tr><td>GenericVideoAudioSwitch
SwitchKVM
SwitchVideo
SwitchAudio
SwitchUART</td><td>Changed the Request Example parameters from fromdevice to fromdeviceid and from todevice to todeviceid.</td></tr>
<tr><td>ClearDisplayArray</td><td>Changed the Response Parameter from Success to success.</td></tr>
<tr><td>GetWindowsForDisplayArray</td><td>Changed the Request Parameter from None to id.</td></tr>
<tr><td>SendUART</td><td>Added the following request parameters: deviceid, targetgroup, port, and data.</td></tr>
<tr><td>GetDeviceByID
GetDeviceByRange</td><td>Added the following response parameters: height, width, and fps.</td></tr>
</table></td></tr>
<tr><td>1/30/19</td><td>6</td><td>1</td><td>Added two listenertypes: irdata and rs232data.
Added a new parameter (device) for listenertypes irdata and rs232data.
Added the following new commands:
SaveLayoutByName
DeleteLayoutByName
DeleteLayoutByID.</td></tr>
<tr><td>5/17/19</td><td>7</td><td>1</td><td>Added the following commands:
GetWindowsForDisplayArrayCount
GetWindowsForDisplayArrayByRange
GetDevicesSubscriptions
Added the following response parameters for GetDeviceByID and GetDeviceByRange
isreceiver
istransmitter
inputconnector
Corrected the response parameter in the documentation for GetLayoutCount.
Corrected the response parameters in the documentation for GetWindowsForDisplayArray.</td></tr>
</table>

6/11/19	8	1	Added a new response parameter (displayarrayid) to the following commands: • GetLayoutByID • GetLayoutByName • GetLayoutsByRange
9/12/19	9	1	Updated sendUART command with escape sequences. Changed the recommended baud rate for the serial port.
10/1/19	10	1	• Added documentation for JSON API Tester. • Documentation correction: Changed name of DeleteLayoutByID command to DeleteLayout. • Updated the escape character rules for SendUART. • Added examples for SendUART. • Added documentation for SendInfrared and ShutdownSystem commands.
10/22/19	11	1	• Updated the GetWindowsForDisplayArray command by adding sourcename and viewindex to the response parameters.
12/17/19	12	1	• Updated how to enable access using Terra Manager 1.3.0. • GetWindowsForDisplayArray - added SourceName request parameter and updated examples. • GetWindowsForDisplayArrayCount - corrected response parameter. • GetWindowsForDisplayArrayByRange – added condensed request parameter.
5/26/20	13	1	• Added Master Layout commands. • Added new request parameters to StopDevice • Updated Summary of Events section. • Updated the Event Response Examples section. • Added BlinkDevice command. • Added number to request parameters for SwitchAudio. • Added additional request parameters for StopDevice. • Updated the listenertypes in the Listener Commands section. • Ability to subscribe to an API message for threat with too many failed logins. • TX USB paired and unpaired. • Added new return values (layout_id, rows, columns, aspect_ratio) to the GetDisplayArraysByRange and GetDisplayArrayByID APIs.
8/12/20	14	1	Updated the special character list and HEX examples for the SendUART command.
11/2/20	15	1	Updated the documentation for the port parameters for SendUART.
05/17/21	16	1	Corrected the recommended baud rate.

NOTICES

COPYRIGHT AND TRADEMARKS

GENERAL

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

Content

Terra API

This document describes the JSON interface control protocol commands for remote access to the Terra system. This protocol is based on the JSON-RPC 2.0 specification.

Reference and Related Documents

Access the latest documentation from the Christie website at http://bit.ly/TerraDownloads

Additional information is available in the following documents:

- Terra Installation and Setup Guide for Controlled Systems (020-102804-*nn*)
- Terra Installation and Setup Guide for Transmitters and the Receivers (020-102814-*nn*)
- Terra Product Safety Guide (020-102786-*nn*)
- Terra User Manual (020-102838-*nn*)
- Terra XY Switcher User Manual (020-102883-*nn)*
- Terra XY Switcher API Reference Manual (020-102884-*nn*)

Transport and Framing

There are several transport methods allowed for this protocol. Message framing varies by transport protocol. All commands should respond within 15 seconds (round trip) unless otherwise specified.

Only one RS-232 connection is supported at one time.

TCP

The baseline communication method for JSON-RPC will be TCP. Each session will consist of one TCP session.

Port 23456 is the TCP/IP port used for JSON communications.

Messages must be framed using the CRLF characters. That is, messages must be separated by the {0x0D, 0x0A} characters. Any data before the two null characters that is not valid JSON is an invalid message and will generate an error messages.

Follow all commands with a Carriage Return followed by a New Line.

Asynchronous notifications may be supported by the server. Therefore, if a client sends overlapping requests, the server may return the replies out-of-order. The client must use the JSON-RPC `id` value to distinguish replies. Server notifications are inserted into the connection that registered for them, therefore a client waiting for a reply may receive other messages before the appropriate reply. A well-behaved client should not open more than 2 TCP connections.

Flow control is handled by TCP. When the TCP stream is broken, any pending registrations/notifications are cleared.

Serial Port (COM1) (RS-232)

JSON RPC can be used over a serial port link via the COM1 port on the controller. The communication parameters must be set as follows:

Baud Rate	9600
Parity	None
StopBits	1
DataBits	8
Handshake	None

Other than the hardware link, all messaging is identical to that of TCP.

The Serial RS232 or Network & RS232 option must be selected in the API Connection page in the Web Manager Global Settings page to enable this method of communication.

Standards and Conventions

1. RFC 4627, "The application/json Media Type for JavaScript Object Notation (JSON)", http://www.ietf.org/rfc/rfc4627
2. JSON-RPC 2.0 Specification, http://www.jsonrpc.org/specification
3. M Series Serial Commands Technical Reference 020-100224-*nn*

RFC 4627 defines the underlying data encoding format, while the JSON-RPC 2.0 specification outlines the message structure, message semantics, and request, response protocol.

The Terra protocol supports all features of JSON-RPC 2.0 including notifications and batch messages.

This document should not be read without reference to the above to specifications. In particular, the nuances of the JSON-RPC semantics and of JSON encoding are not described in this document.

Command and Event Names

All commands and event names are all lowercase letters. For ease of reading, this document uses initial capital letters in the document headings.

Parameters

For any message that specifies no parameters, the `params` member may either be omitted (as per JSON-RPC 2.0 spec), or transmitted as an empty array [] or object {}.

Batch Messages

Batch messages are supported as described by the JSON-RPC 2.0 specification with the caveat that they will be executed in order. That is, the second method in a batch request may rely on the first. Note: there may be some commands whose response does not necessarily indicate completion.

Languages

When the language for a Terra controller has been set, all labels/names are localized using UTF-8 encoding.

Reply and Acknowledgement

Reply and acknowledgement follow the JSON-RPC 2.0 standard.

Notifications have no `id` value and are used by the Terra controller for notifications.

Requests must have an `id` value, this value may be any JSON primitive value, though it should normally be a unique number within the range of a signed int32. JSON requests may be used for either a request/query or for a command, where an acknowledgement is desired.

JSON-RPC 2.0 requests are nominally asynchronous, though a client may employ synchronous communication by not sending a subsequent request until previous replies have been received.

Error Messages

Protocol-level errors must comply with the JSON-RPC 2.0 standard error codes and formats.

Generic errors such as invalid (or missing) parameters, etc. are not explicitly specified in this document. However, runtime errors are occasionally spelled out, where the specific message may be of importance.

Code	Message	Meaning
-32700	Parse error	Invalid JSON was received by the server. An error occurred on the server while parsing the JSON text.
-32600	Invalid request	The JSON sent is not a valid Request object.
-32601	Method not found	The method does not exist / is not available.
-32602	Invalid params	Invalid method parameter(s).
-32603	Internal error	Internal JSON-RPC error.

Generic Examples

Most methods described in this document include examples, using the following notation:

To illustrate a particular client request and server response:

Request example

```
{"id": 1, "jsonrpc": "2.0", "method": "abc", "params": {}}
```

Response example

```
{"id": 1, "jsonrpc": "2.0", "result": 123}
```

Asynchronous Notifications (Events)

The Terra protocol is designed to use asynchronous messaging where appropriate. Clients can register and unregister for various notification events, which are messages sent asynchronously from the server, typically upon a state change in the server.

Each namespace contains `addListener` and `removeListener` methods, and often it is possible to register for multiple (related) methods in a single call. Unless noted, the default behavior of the server is to immediately send the initial state in the form of an asynchronous notification message.

If `addListener` has been called more times than its corresponding `removeListener` method, notifications will continue. This implies that the server will count registration instances for each client.

Events

This section describe the different types of notification events that can occur for the system or system components. Events are sent with the "method" property stating the type of event. The parameters specify the entity that was impacted by the event.

After receiving the ID from an event, use the appropriate API command to get the updated details for the ID.

Summary of Events

The following is a summary of events and a description of when they occur:

Event Type (method)	Description	Parameters (params)
deviceadded	Device was added.	id
devicedeleted	Device was deleted and its IP address was released.	id
deviceupdated	Device was updated.	id
displayarrayadded	Display array was added.	id
displayarraydeleted	Display array was deleted.	id
displayarrayupdated	Configuration for the display array was updated.	id
layoutadded	New layout was added.	id
layoutappliedtodisplayarray	Layout was applied to specified display array.	layoutid, displayarrayid
layoutdeleted	Layout was deleted.	id
layoutupdated	Layout was updated.	id
systemrestart	Controller was restarted. Optionally, the Transmitters and Receivers can be restarted.	
systemshutdown	System controller was powered off.	
txusbunpaired	TX and RX USB is unpaired.	source, destinations
usbnotpaired	Simultaneous User Mode is enabled and a TX is not paired to an RX since the limit of seven allowable pairs has been reached.	usbnotpaired (list of sources and their destinations)
userlockout	User locked out.	userid, ip (list of IP addresses)

Event Response Format

{"jsonrpc": "2.0", "method": "*method*", "params": { "id": "*entity id*" }, "id": 1}

System events do not have any parameters.

Event Response Examples

This example is a notification that the device with id DEVICE1 was added:

{"jsonrpc": "2.0", "method": "deviceadded", "params": { "id": "DEVICE1" }, "id": 1}

This example is a notification that LAYOUTWIDE was applied to the display array with the id of DISPLAYARRAY1.

{"jsonrpc": "2.0", "method": "layoutappliedtodisplayarray", "params": { "layoutid": "LAYOUTWIDE", "displayarrayid": "DISPLAYARRAY1" }, "id": 1}

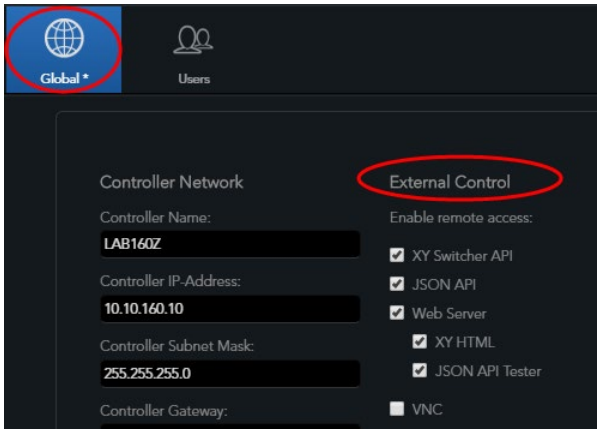
This example is a notification that a user has been locked out:

{"jsonrpc":"2.0","method":"userlockout","params":{"userid":"TESTER1","ip":["192.168.13.2","10.10.50.41"]},"id":1}

Enable Access and Device Configuration

Use the External Controller settings on the Terra Manager Global page to enable remote access to Terra using the JSON commands, Terra XY Switcher software, and the JSON API Tester.

Access can be set up using TCP protocol or a RS-232 connection via the serial port on the Controller.



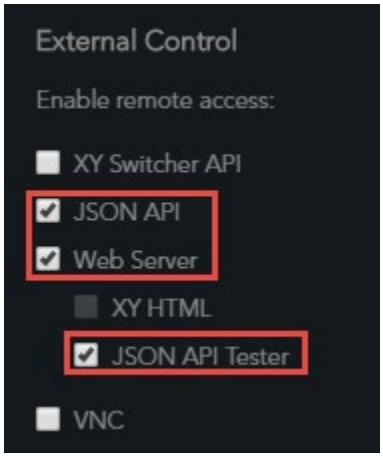
All Terra devices must be configured using Terra Manager.

Terra JSON API Tester

The Terra JSON API Tester is a tool for verifying Terra API commands when using a third-party control system. The Tester lists the parameters for each command and displays the command format that is sent and received. The tool also tracks a history of the tested commands.

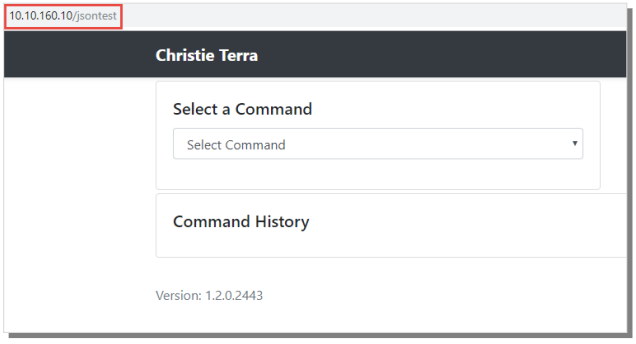
Enabling Access to the Tester

Access to the Terra API Tester is enabled on the Terra Manager Global Page by checking the JSON API, Web Server, and JSON API Tester check boxes.

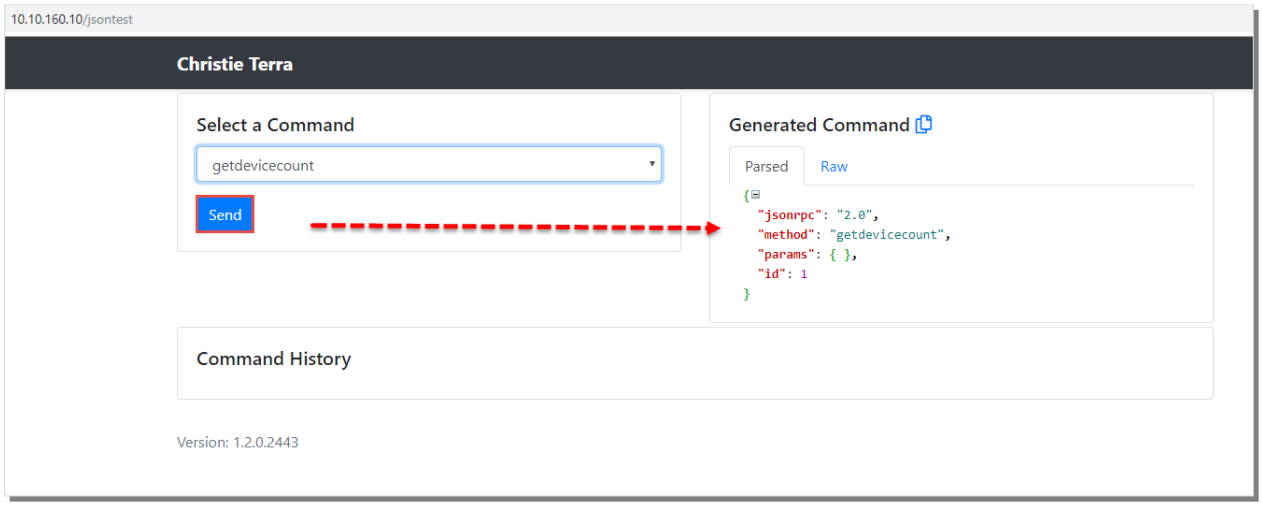


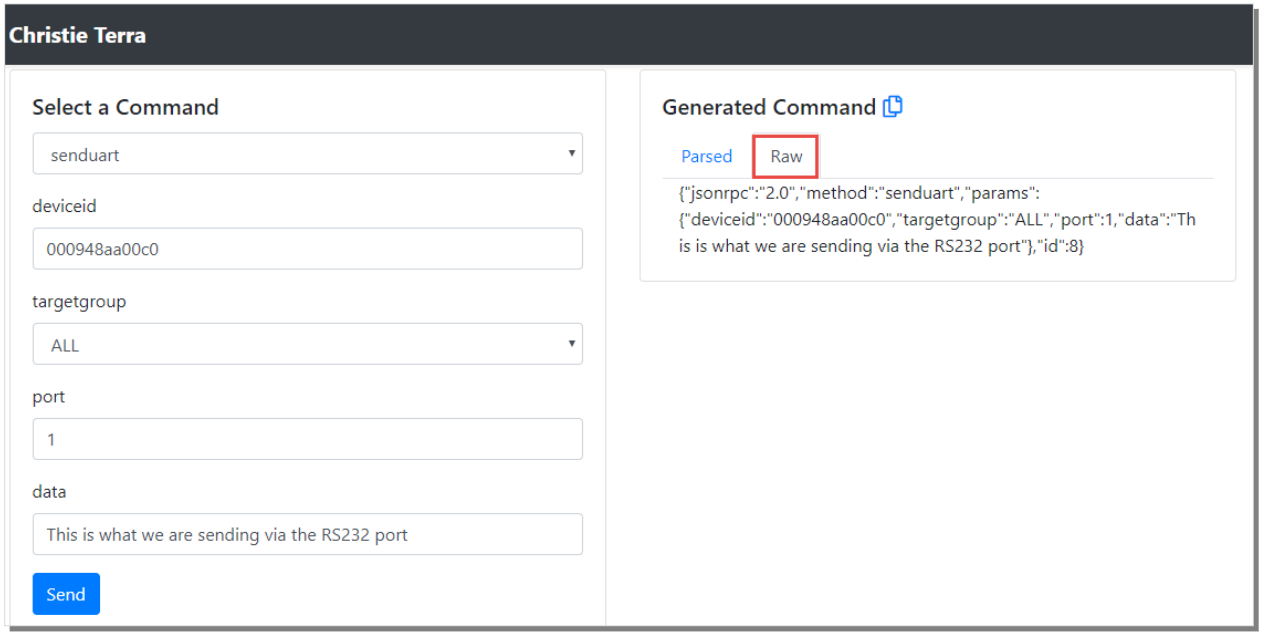
Using the Tester

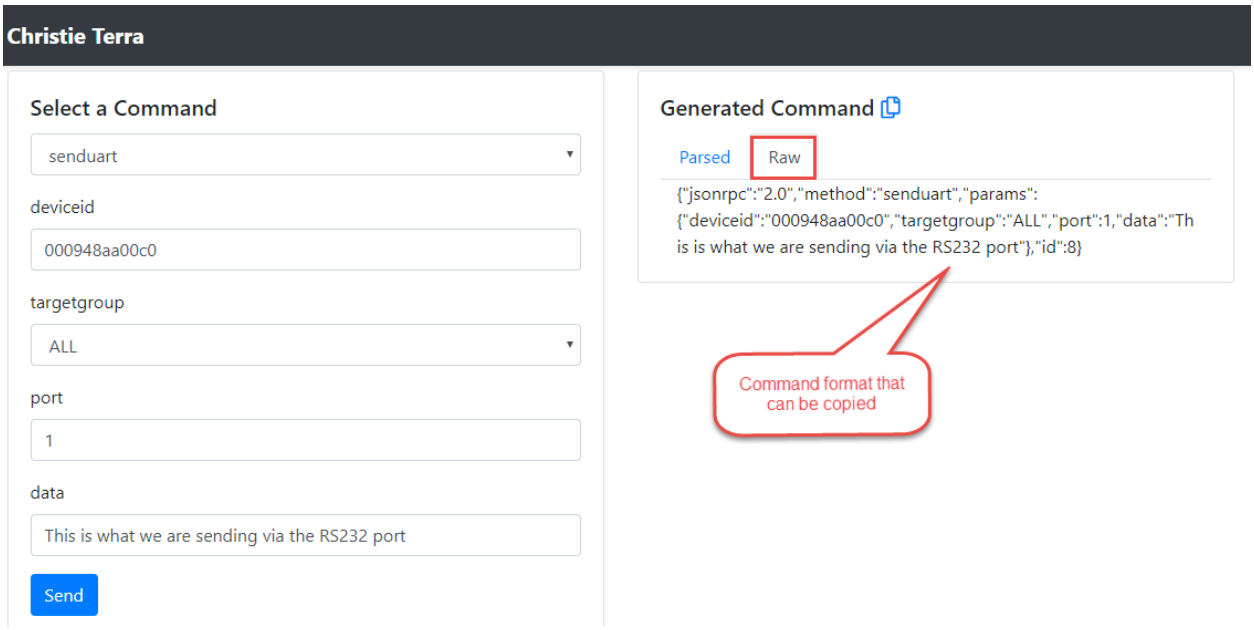
1. Access the tester by browsing to http://*IPAddressOfController*/jsontest
   For example: http://10.10.160.10/jsontest
   The following page is displayed:



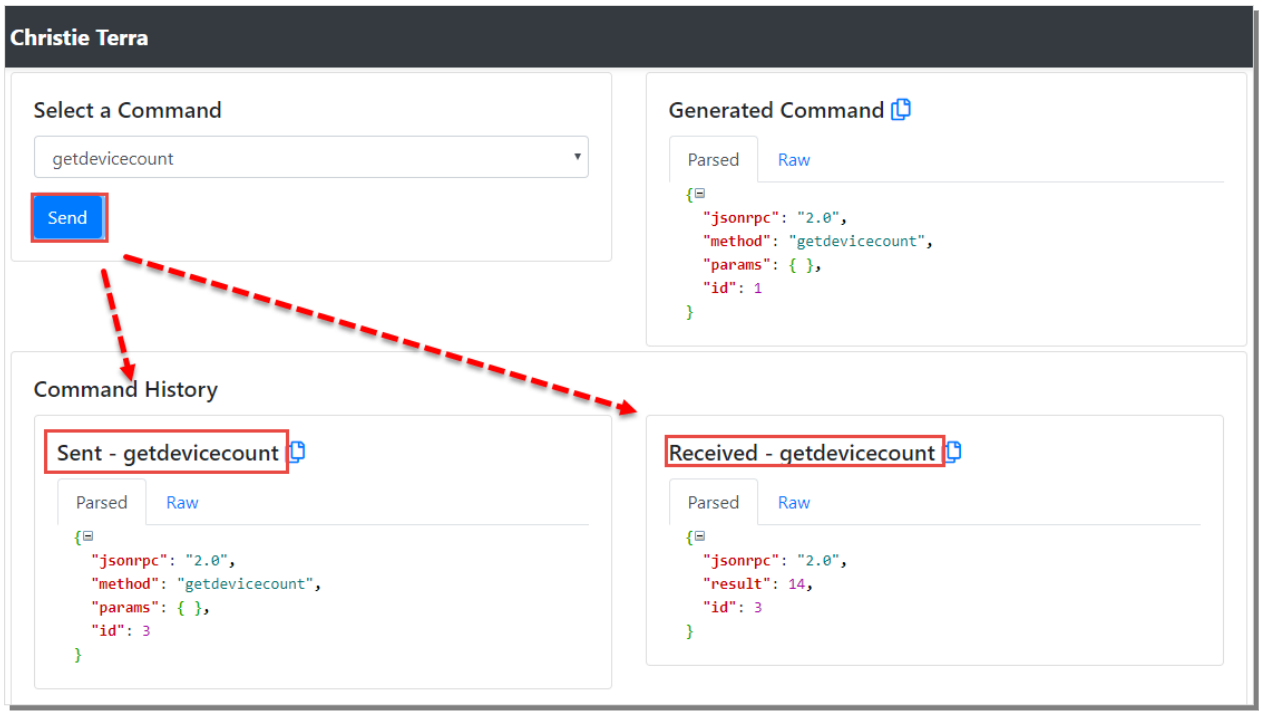
2. Select the command you want to test from the dropdown list, input associated parameters (if listed), and click **Send**. The generated command is displayed in the Generated Command panel. Optionally, click **Raw** on the Generated Command panel to display the actual JSON script to copy for your command implementation. See the following examples (with and without parameters).







When **Send** is clicked, the Command History is displayed. This history is updated with each executed command during a JSON API Tester session. The history is cleared when the session is closed.



Commands

The commands are categorized by types: layout, device control and switching, device, display array, listener, and system.

Layout Commands

This section describes the layout commands.

ApplyLayout

Applies the specified layout to the display array is was created on.

Request parameters

layoutid	The ID of the layout you want to apply.

Request example

{"jsonrpc": "2.0", "method": "applylayout", "params": { "layoutid": "ENTITYID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

ApplyLayoutByName

Applies the specified layout to the display array.

Request parameters

name	The name of the layout you want to apply.

Request example

{"jsonrpc": "2.0", "method": "applylayoutbyname", "params": { "name": "LAYOUTNAME" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

ApplyVideoSourceByViewIndex

Using the source index number (Window ID), switches the source displayed in the current window. The modification to the layout is not saved.

Terra Window ID Conventions

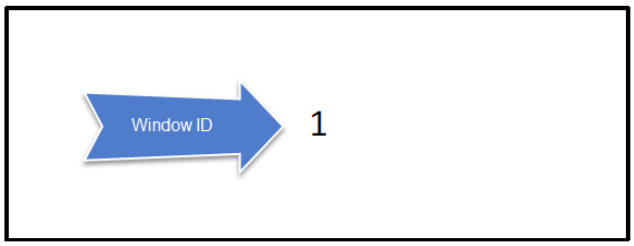
Each window position is assigned a number starting at base 1. The XY position of the upper left pixel is the decisive factor for the numbering.

The window ID sequence starts at the top left, going left to right; then goes down one level, again starting from left to right, etc.

Conventions for Non-Multiview Layouts

Each Window gets the next sequential number assigned. See the two examples below.

1X2 Display Array



4X4 Display Array

1	2	3	4
5	6 4 displays with one image across the 2X2		7
8			9
10	11	12	13

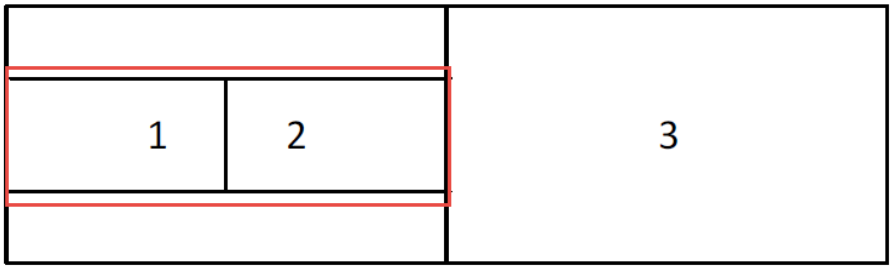
Conventions for Multiview Layouts

Multiviewer follows the same convention within the multiview layout; however, a multiview layout is treated as a standalone display when mixed with other Display Arrays. See the examples below.

1X2 Display Array with 1 **Multiviewer on the right**

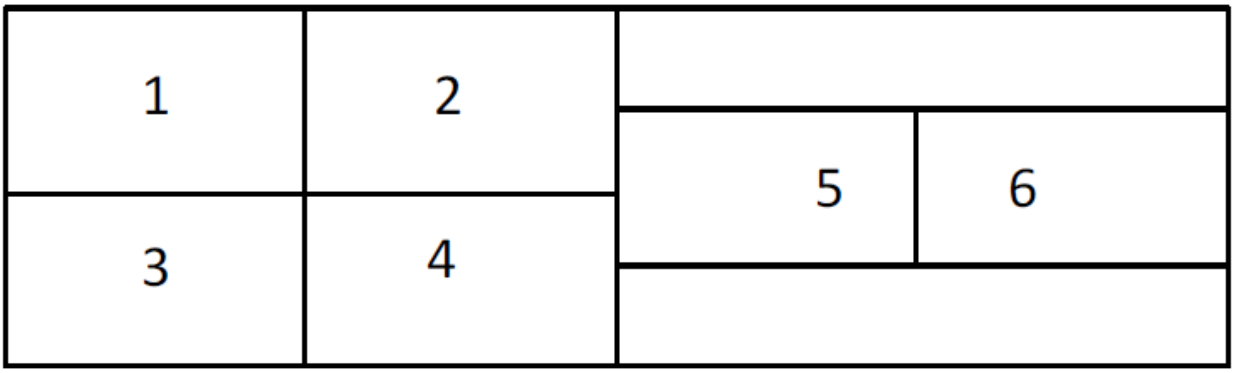
1	2	3
	4	5

1X2 Display Array with 1 **Multiviewer on the left**

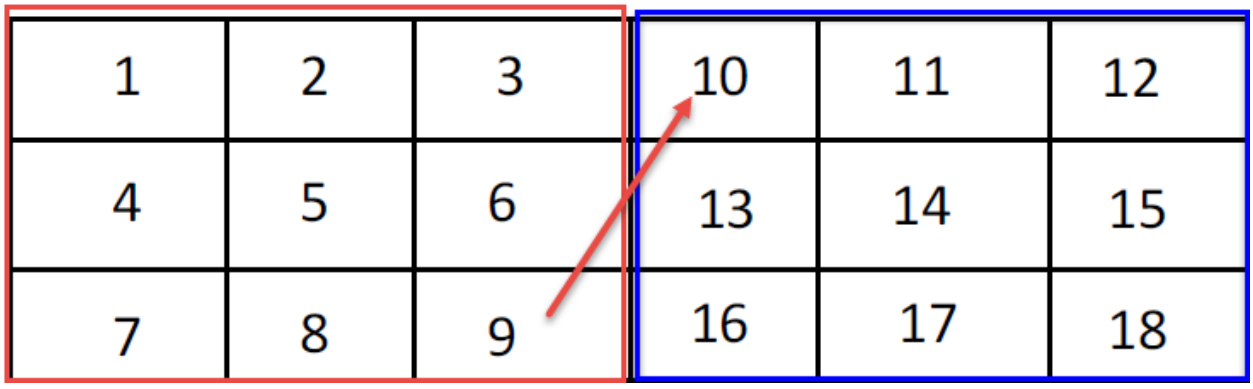


1X2 Display Array with 2 **Multiviewers**

Windows # 3 and 4 take a higher window priority over 5 and 6, although 5 and 6 have a higher vertical pixel position because multiview sources are all grouped together as one source group.



1X2 Display Arrays with 2 - **Multiviewers**



Request parameters

displayarrayid	ID of the Display Array.
layoutid	ID of the layout.
viewindex	Index for the Window where the new source will be displayed.
sourceid	ID of the source (TX).

Request example

```
{"jsonrpc": "2.0", "method": "applyvideosourcebyviewindex", "params": { "displayarrayid": "ENTITYID", "layoutid": "ENTITYID", "viewindex": "ENTITYID", "sourceid": "ENTITYID" }, "id": 1}
```

Response parameters

Success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

GetLayoutByID

Gets the specified layout.

Request parameters

id	The ID of the layout you want to get.

Request example

{"jsonrpc": "2.0", "method": "getlayoutbyid", "params": { "id": "ENTITYID" }, "id": 1}

Response parameters

id	The ID of the layout that was retrieved.
name	Name of the layout that was retrieved.
displayarrayid	The ID of the display array where the layout was created.

Response example

{"jsonrpc":"2.0","result": { "id": "ENTITYID", "name": "ENTITYNAME", "displayarrayid": "DISPLAYARRAYID"},"id":1}

GetLayoutsByRange

Returns a list of layouts within a specified range.

Request parameters

startindex	The number of the first layout to retrieve.
endindex	The number of the last layout to retrieve.

Request example

{"jsonrpc": "2.0", "method": "getlayoutsbyrange", "params": { "startindex": index, "endindex": index }, "id": 1}

Response parameters

id	Layout ID that was retrieved.
name	Layout Name that was retrieved.
displayarrayid	The ID of the display array where the layout was created.

Response example

{"jsonrpc":"2.0","result": { [{"id": "ENTITYID", "name": "ENTITYNAME"}, {"id": "ENTITYID", "name": "ENTITYNAME"}, "displayarrayid": "DISPLAYARRAYID"] },"id":1}

GetLayoutCount

Returns the total number of Layouts.

Request parameters

None.

Request example

{"jsonrpc": "2.0", "method": "getlayoutcount", "params": { }, "id": 1}

Response parameters

Number of layouts.

Response example

{"jsonrpc":"2.0","result":NUMBER_OF_LAYOUTS,"id":1}

GetLayoutByName

Returns a list of layouts by name.

Request parameters

name	The name of the layout you want to get.

Request example

{"jsonrpc": "2.0", "method": "getlayoutbyname", "params": { "name": "LAYOUTNAME" }, "id": 1}

Response parameters

id	The system id for the layout.
name	The name of the layout.
displayarrayid	The ID of the display array where the layout was created.

Response example

{"jsonrpc":"2.0","result": { "id": "ENTITYID", "name": "ENTITYNAME", "displayarrayid": "DISPLAYARRAYID" },"id":1}

SaveLayoutByName

Saves the current layout on the specified Display Array.

Request parameters

displayarrayid	The ID of the display array that contains the desired layout.
name	The name of the layout to add/update.
Overwrite (check box in JSON tester)	True or false (default). True overwrites a layout that has the same name in layoutname, otherwise a new layout is created. False checks to see if the layout name exists. If it does not, a new layout with the name in layoutname is created. If a layout with the same name exists, an error is returned.

Request example

{"jsonrpc": "2.0", "method": "savelayoutbyname", "params": { "displayarrayid": "DISPLAYARRAYID", "name" : "NAME", "overwrite" : false }, "id": 1}

Response parameters

id	The name of the layout that was created.
layoutid	The ID of the layout that was created.
displayarrayid	The ID of the display array the layout was created on.

Response example

{"jsonrpc":"2.0","result":{"success": true, "name" : "NAME", "layoutid" : "LAYOUT_ID", "displayarrayid": "DISPLAYARRAYID" },"id":1}

DeleteLayoutByName

Deletes the layout using the specified layout name.

Request parameters

name	The name of the layout to delete.

Request example

{"jsonrpc": "2.0", "method": "deletelayoutbyname", "params": { "name" : "NAME" }, "id": 1}

Response parameters

None.

Response example

{"jsonrpc":"2.0","result":{"success": true },"id":1}

DeleteLayout

Deletes a layout using the specified layout ID.

Request parameters

layoutid	The ID of the layout to delete.

Request example

{"jsonrpc": "2.0", "method": "deletelayout", "params": { "layoutid" : "LAYOUTID" }, "id": 1}

Response parameters

None.

Response example

{"jsonrpc":"2.0","result":{"success": true },"id":1}

Master Layout Commands

This section describes the layout commands for master layouts.

ApplyMasterLayout

Applies the specified master layout.

Request parameters

masterlayoutid	The ID of the master layout you want to apply.

Request example

{"jsonrpc":"2.0","method":"applymasterlayout","params":{"masterlayoutid":"c74cab67-cd23-4d31-b5fe-7d26d632ebce"},"id":16}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":9}

ApplyMasterLayoutByName

Applies the specified master layout using the specified name.

Request parameters

name	The name of the master layout you want to apply.

Request example

{"jsonrpc":"2.0","method":"applymasterlayoutbyname","params":{"name": "master layout name 1"},"id":9}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":9}

GetMasterLayoutByID

Gets the specified master layout using the specified ID.

Request parameters

masterlayoutid	The ID of the master layout you want to get.

Request example

{"jsonrpc":"2.0","method":"getmasterlayoutbyid","params":{"masterlayoutid":"c74cab67-cd23-4d31-b5fe-7d26d632ebce"},"id":19}

Response parameters

id	Master Layout ID that was retrieved.
name	Master Layout Name that was retrieved.
layouts	Array of layout ids that make up the master layout that were retrieved.

Response example

{"jsonrpc":"2.0","result":{"id":"c74cab67-cd23-4d31-b5fe-7d26d632ebce","name":"1","layouts":["a43f8b20-d171-4dda-b308-61ca1c676579","5ed74332-8d74-470e-981e-bdcc65cbc719"]},"id":19}

GetMasterLayoutByName

Returns a list of master layouts using a specific name.

Request parameters

name	The name of the master layout you want to get.

Request example

{"jsonrpc":"2.0","method":"getmasterlayoutbyname","params":{"name":"master layout name"},"id":22}

Response parameters

id	Master Layout ID that was retrieved.
name	Master Layout Name that was retrieved.
layouts	Array of layout ids that make up the master layout that were retrieved.

Request example

{"jsonrpc":"2.0","result":{"id":"c74cab67-cd23-4d31-b5fe-7d26d632ebce","name":"1","layouts":["a43f8b20-d171-4dda-b308-61ca1c676579","5ed74332-8d74-470e-981e-bdcc65cbc719"]},"id":22}

DeleteMasterLayout

Deletes a master layout using the specified layout ID.

Request parameters

masterlayoutid	The ID of the master layout you want to delete.

Request example

{"jsonrpc":"2.0","method":"deletemasterlayout","params":{"masterlayoutid":"c74cab67-cd23-4d31-b5fe-7d26d632ebce"},"id":23}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true },"id":23}

DeleteMasterLayoutByName

Deletes the master layout using the specified layout name.

Request parameters

name	The name of the master layout you want to delete.

Request example

{"jsonrpc":"2.0","method":"deletemasterlayoutbyname","params":{"name":"master layout name"},"id":24}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true },"id":}

ClearMasterLayout

Clears layouts for the master layout using a specific ID.

Request parameters

masterlayoutid	The ID of the master layout you want to clear.

Request example

{"jsonrpc":"2.0","method":"clearmasterlayout","params":{"masterlayoutid":"c74cab67-cd23-4d31-b5fe-7d26d632ebce"},"id":25}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true },"id":1}

ClearMasterLayoutByName

Clears layouts for the master layout using a specific name.

Request parameters

name	The name of the master layout you want to clear.

Request example

{"jsonrpc":"2.0","method":"clearmasterlayoutbyname","params":{"name":"master layout name"},"id":26}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true },"id":26}

GetMasterLayoutCount

Returns the total number of Master Layouts.

Request parameters

None.

Request example

{"jsonrpc":"2.0","method":"getmasterlayoutcount","params":{},"id":6}

Response parameters

None.

Response example

{"jsonrpc":"2.0","result":2,"id":6}

GetMasterLayoutsByRange

Returns a list of master layouts within a specified range.

Request parameters

startindex	The number of the first layout to retrieve.
endindex	The number of the last layout to retrieve.

Request example

{"jsonrpc":"2.0","method":"getmasterlayoutsbyrange","params":{"startindex":0,"endindex":100},"id":4}

Response parameters

id	Master Layout ID that was retrieved.
name	Master Layout Name that was retrieved.
layouts	Array of layout ids that make up the master layout that were retrieved.

Request example

{"jsonrpc":"2.0","result":[{"id":"c74cab67-cd23-4d31-b5fe-7d26d632ebce","name":"Test Master Layout 1","layouts":["a43f8b20-d171-4dda-b308-61ca1c676579","5ed74332-8d74-470e-981e-bdcc65cbc719"]},{"id":"c043bd7c-d4df-41eb-b6f1-3b2f2c9d2e2d","name":"Test Master Layout 2","layouts":["a43f8b20-d171-4dda-b308-61ca1c676579","5ed74332-8d74-470e-981e-bdcc65cbc719"]}],"id":4}

SaveMasterLayoutByName

Saves the current applied layouts to a new Master Layout.

Request parameters

name	The name of the master layout to add or update.
Overwrite (check box in JSON tester)	True (1) or false (0). True overwrites a master layout that has the same name, otherwise a new master layout is created. False (default) checks to see if the master layout name exists. If it does not, a new master layout using the name is created. If a master layout with the same name exists, an error is returned.

Request example

{"jsonrpc":"2.0","method":"savemasterlayoutbyname","params":{"name":"new master layout name","overwrite":0},"id":27}

Response parameters

id	ID of the Master Layout ID that was saved.
name	Name of the Master Layout Name that was saved.
layouts	List of layout ids that make up the master layout.

Device Control Commands

This section describes the device control commands.

GenericVideoAudioSwitch

Switches to the transmitter's video and audio.

Request parameters

fromdeviceid	The source device ID (TX).
todeviceid	The target device ID (RX).

Request example

{"jsonrpc": "2.0", "method": "genericvideoaudioswitch", "params": { "fromdeviceid": "DEVICEID", "todeviceid": "DEVICEID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{""success": true},"id":1}

ApplyVideoSource

Switches a video source of a window in a layout.

Request parameters

displayarrayid	ID of the Display Array.
layoutid	ID of the layout.
windowid	ID of the window in the Display Array.
sourceid	ID of the source (TX).

Request example

{"jsonrpc": "2.0", "method": "applyvideosource", "params": { "displayarrayid": "ENTITYID", "layoutid": "ENTITYID", "windowid": "ENTITYID", "sourceid": "ENTITYID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

ApplyVideoSourceByViewIndex

Refer to page *17*.

SendUART

Sending UART (universal asynchronous receiver-transmitter) data to one or more devices.

- If sending to one device, specify the deviceid. When deviceid is specified, the targetgroup should not be specified and is ignored.
- If sending to more than one device, use one of the targetgroup parameters (NONE, ALL, ALL_TX, ALL_RX).

Escape character rules for sending data via RS232

The string must be escaped if it contains ASCII control characters or non-ASCII byte values.

Space characters must also be escaped.

The following escape sequences are recognized:

Special Character	Escaped Output	Purpose	Hex
Space character	\(space)	Replaces the ASCII space character.	20
Single backslash (\) character	\\	Replaces the ASCII backslash character.	5C
Quotation mark	\”	Replaces a double quote.	22
Null character	\0	Replaces a null character.	00
Backspace	\b	Replaces the ASCII backspace character.	08
Form feed	\f	Replaces the ASCII form feed	0C
Line feed (new line)	\n	Replaces the ASCII line feed character.	0A
Carriage return	\r	Replaces the ASCII carriage return character.	0D
Tab (horizontal)	\t	Replaces the ASCII tab character.	09
Hexadecimal values	\x*nn*	Where *nn* are two hexadecimal values (two hexadecimal digits) of the character, replaces: • All ASCII control characters (hex. 1F and below) not already handled above • The ASCII DEL character (hex 7f) • All non-ASCII characters (hex 80 and above)	

To guarantee delivery of all RS-232 data between devices, the baud rate should be set to ≤19,200 bit/S. For passing RS-232 data between the endpoint and the API Server (Terra Controller), a baud rate up to 115.2 bit/S is sustained.

Request parameters

deviceid	Send data to specific device.
targetgroup	Send data to more than one device: ALL, ALL_TX, or ALL_RX.
port	Port on the receiving devices, 0 (RS-232 port) or 1 (1G Ethernet port).
data	RS232 data.

Request example

{"jsonrpc": "2.0", "method": "senduart", "params": { "deviceid": "ENTITYID", "targetgroup": "NONE", "port": port, "data": "UART DATA" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success"": true},"id":1}

Examples

Example set 1: Setting a Mirage Test pattern using the Terra JSON messaging

The command to SET the Engine test pattern would look like this:

```
{"jsonrpc": "2.0", "method": "senduart", "params": { "deviceid":
"000948dd023c", "targetgroup": "NONE", "port": 1, "data": "ETP 1" }, "id": 2}
```

The highlighted text is the actual string being sent to the projector by the3rd party device.

Result: Sets test pattern to green .

The command to SET the Engine test pattern would look like this:

```
{"jsonrpc": "2.0", "method": "senduart", "params": { "deviceid":
"000948dd023c", "targetgroup": "NONE", "port": 1, "data": "ETP 2" }, "id": 2}
```

The highlighted text is the actual string being sent to the projector by the3rd party device.

Result: Sets test pattern to red.

The command to SET the Engine test pattern would look like this:

```
{"jsonrpc": "2.0", "method": "senduart", "params": { "deviceid":
"000948dd023c", "targetgroup": "NONE", "port": 1, "data": "ETP 3" }, "id": 2}
```

The highlighted text is the actual string being sent to the projector by the3rd party device.

Result: Sets test pattern to blue.

Example set 2: Powering Mirage on/off using Terra JSON messaging

The command to TURN ON the Mirage would look like this:

```
{"jsonrpc": "2.0", "method": "senduart", "params": { "deviceid": "000948dd023c",
"targetgroup": "NONE", "port": 1, "data": "SET PWR1" }, "id": 2}
```

The highlighted text is the actual string being sent to the projector by the3rd party device.

Result: Mirage powers on.

Turning off the Mirage using Terra JSON messaging.

The command to TURN OFF the Mirage would look like this

```
{"jsonrpc": "2.0", "method": "senduart", "params": { "deviceid": "000948dd023c",
"targetgroup": "NONE", "port": 1, "data": "SET PWR0" }, "id": 2}
```

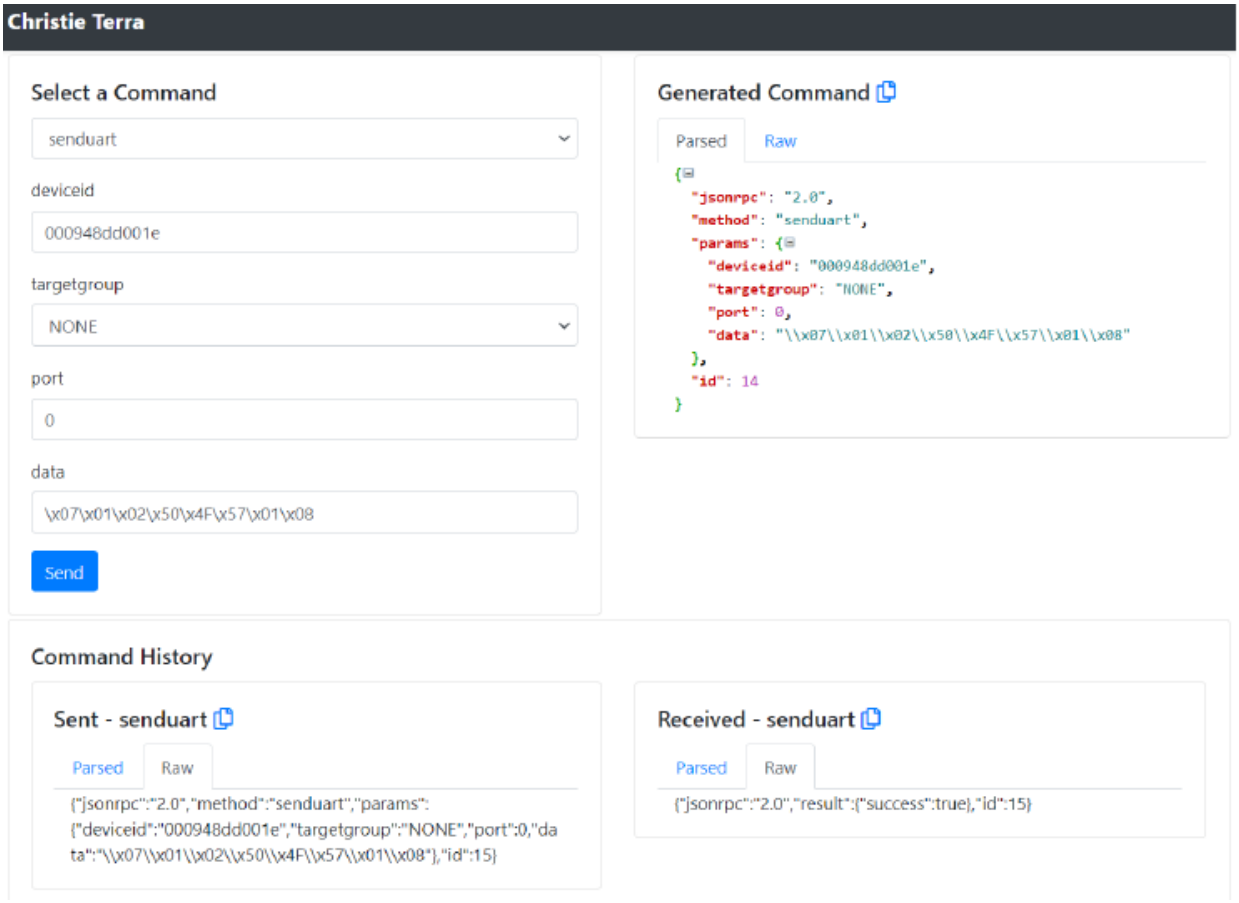
The highlighted text is the actual string being sent to the projector by the3rd party device.

Result: Mirage powers off.

Example set 3: HEX data to turn off a Christie Extreme Series LCD Panel

The hex data to TURN OFF a Christie Extreme Series LCD Panel would look like this:

```
07 01 02 50 4F 57 00 08
HEX Data after adhering to Terra's Escape rules:
\x07\x01\x02\x50\x4F\x57\x01\x08
If a Carriage Return is needed, add either:
\r or \x0D
The resulting SendUART Command:
{"jsonrpc":"2.0","method":"senduart","params":{"deviceid":"000948dd001e","targetg
roup":"NONE","port":0,"data":"\\x07\\x01\\x02\\x50\\x4F\\x57\\x01\\x08"},"id":15}
```



SendInfrared

Sending infrared data to one or more devices.

- If sending to one device, specify the deviceid. When deviceid is specified, the targetgroup should not be specified and is ignored.
- If sending to more than one device, use one of the targetgroup parameters (NONE, ALL, ALL_TX, ALL_RX).

Request parameters

deviceid	Send data to specific device.
targetgroup	Send data to more than one device: ALL, ALL_TX, or ALL_RX.
port	Port on the receiving devices, 0 or 1.
data	RS232 data.

Request example

{"jsonrpc": "2.0", "method": "sendinfrared", "params": { "deviceid": "ENTITYID", "targetgroup": "NONE", "port": port, "data": "UART DATA" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success"": true},"id":1}

BlinkDevice

Starts or stops blinking on all the LEDs on the specified device(s).

Request parameters

id	ID of the device(s) to blink.
Blink (check box in the JSON Tester)	Check box to start or stop blinking. Checked starts blinking, unchecked stops blinking.

Request example

{"jsonrpc":"2.0","method":"blinkdevice","params":{"id":"000948aa0012","blink":true},"id":8}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success":true},"id":9}

Switching Commands

This section describes the switching commands.

SwitchKVM

Switching the USB signals from a Transmitters to a Receiver.

Request parameters

fromdeviceid	The source device ID (TX).
todeviceid	The target device ID (RX).

Request example

{"jsonrpc": "2.0", "method": "switchkvm", "params": { "fromdeviceid": "DEVICEID", "todeviceid": "DEVICEID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

SwitchVideo

Switches the video source for a device.

Request parameters

fromdeviceid	The source device ID (TX).
todeviceid	The target device ID (RX).

Request example

{"jsonrpc": "2.0", "method": "switchvideo", "params": { "fromdeviceid": "DEVICEID", "todeviceid": "DEVICEID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

SwitchAudio

Switches the audio source for a device.

Request parameters

fromdeviceid	The source device ID (TX).
todeviceid	The target device ID (RX).
type	0 =digital (HDMI embedded audio) 1 = analog

Request example

{"jsonrpc":"2.0","method":"switchaudio","params":{"fromdeviceid":"000948aa00e8","todeviceid":"000948aa03e6","type":0},"id":15}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

"jsonrpc":"2.0","result":{"success":true},"id":15}

SwitchUART

Switching the UART (universal asynchronous receiver-transmitter) signals from a Transmitter to a Receiver.

Request parameters

fromdeviceid	The source device ID (TX).
todeviceid	The target device ID (RX).

Request example

{"jsonrpc": "2.0", "method": "switchuart", "params": { "fromdeviceid": "DEVICEID", "todeviceid": "DEVICEID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

SwitchInputConnector

Switches the input (HDMI or DisplayPort) on a Transmitter.

Request parameters

deviceid	HDMI=0 DisplayPort=1

Request example

{"jsonrpc": "2.0", "method": "switchinputconnector", "params": { "deviceid": "DEVICEID", "connector": 0/1 }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

Device Commands

This section describes the device commands.

GetDeviceByID

Gets the specified device.

Request parameters

id	The ID of the device you want to get.

Request example

{"jsonrpc": "2.0", "method": "getdevicebyid", "params": { "id": "ENTITYID" }, "id": 1}

Response parameters

id	Device ID that was retrieved.
name	Device Name that was retrieved.
height	Height in pixels.
width	Width in pixels.
fps	Frames per second.
isreceiver	Device is a receiver.
istransmitter	Device is a transmitter.
inputconnector	Type of input connector:

none	Device is a receiver (no input connector available).
DISPLAY_PORT	Device is connected to a DISPLAY_PORT port.
HDMI	Device is connected to a HDMI port.

Response example

For Receiver:
{"id":"000948dd004a","name":"1234567","height":1080,"width":1920,"fps":60,"isreceiver":true,"istransmitter":false,"inputconnector":"none"}

For Transmitter:
{"id":"f82285000855","name":"F8228500085","height":1080,"width":1920,"fps":60,"isreceiver":false,"istransmitter":true,"inputconnector":"HDMI"}

GetDevicesByRange

Returns a list of devices within a specified range.

Request parameters

startindex	The number of the first device to retrieve.
endindex	The number of the last device to retrieve.

Request example

{"jsonrpc": "2.0", "method": "getdevicesbyrange", "params": { "startindex": index, "endindex": index }, "id": 1}

Response parameters

id	Device ID that was retrieved.
name	Device Name that was retrieved.
height	Height in pixels.
width	Width in pixels.
fps	Frames per second.
isreceiver	Device is a receiver.
Istransmitter	Device is a transmitter.
inputconnector	Type of input connector:

none	Device is a receiver (no input connector available).
DISPLAY_PORT	Device is connected to a DISPLAY_PORT port.
HDMI	Device is connected to a HDMI port.

Response example

For Receiver:
{"id":"000948dd004a","name":"1234567","height":1080,"width":1920,"fps":60,"isreceiver":true,"istransmitter":false,"inputconnector":"none"}

For Transmitter:
{"id":"f82285000855","name":"F8228500085","height":1080,"width":1920,"fps":60,"isreceiver":false,"istransmitter":true,"inputconnector":"HDMI"}

StopDevice

Stops the process of sending (TX) or receiving (RX) video (with no parameter selected) and the specified stream. Use the ApplyLayout or ApplyMasterLayout command to restart the stream.

Request parameters

deviceid (string)	ID of the device.
type	0=Stop All Video streams (default) 1=Stop Native Video 2=Stop Scaled Video 3=Stop Audio 4=Stop USB 5=Stop RS232 6=Stop IR 99=Stop All

Request example

{"jsonrpc":"2.0","method":"stopdevice","params":{"deviceid":"000948aa0012","type":0},"id":42}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success":true},"id":43}

GetDeviceCount

Returns the total number of devices.

Request parameters

None.

Request example

{"jsonrpc": "2.0", "method": "getdevicecount", "params": { }, "id": 1}

Response parameters

None.

Response example

{"jsonrpc":"2.0","result":NUMBER_OF_DEVICES,"id":1}

GetDevicesSubscriptions

Lists all streams that the device is using. For receivers, lists all devices the receiver is currently streaming. For transmitters, lists all streams that are being used by each receiver.

Request parameters

Id	ID of the device.

Request example

{"jsonrpc": "2.0", "method": "getdevicesubscriptions", "params": { "id": "000948dd004a" }, "id": 1}

Response parameters

stream type	NONE	No stream available.
	HDMI	Uses the stream from the HDMI port.
	HDMI AUDIO	Uses the audio from the HDMI source.
	AUDIO	Uses the analog audio stream is embedded into the HDMI output.
	RS232	Uses the stream from the RS-232 port.
	INFRARED	Uses the stream from the IR port.
	IS2 AUDIO	Uses uncompressed audio for the I2S bus.
	USB	Uses stream from the USB port.

Response example

{"jsonrpc":"2.0","result":[{"id":"f82285000855","streamtype":"HDMI","xyindex":110},
{"id":"f82285000855","streamtype":"HDMI_AUDIO","xyindex":110}],"id":1}

Display Array Commands

This section describes the Display Array commands.

GetDisplayArrayByID

Request parameters

id	The ID of the display array to retrieve.

Request example

{"jsonrpc": "2.0", "method": "getdisplayarraybyid", "params": { "id": "ENTITYID" }, "id": 1}

Response parameters

id	The ID of the display array that was retrieved.
name	Name of the display array that was retrieved.
layout_id	The ID of the layout that was retrieved.
rows	Number of rows in the display array.
columns	Number of columns in the display array.
aspect_ratio	The ratio of the width to the height for the displays within the array.

Response example

{"jsonrpc":"2.0","result": { "id": "ENTITYID", "name": "ENTITYNAME", "layout_id": "LAYOUTID", rows: 1, columns: 1, aspect_ratio: 1.7777777777777777 },"id":1}

GetDisplayArraysByRange

Returns a list of display arrays within a specified range

Request parameters

startindex	The number of the first display array to retrieve.
endindex	The number of the last display array to retrieve.

Request example

{"jsonrpc": "2.0", "method": "getdisplayarraysbyrange", "params": { "startindex": index, "endindex": index }, "id": 1}

Response parameters

id	The ID of the display array that was retrieved.
name	Name of the display array that was retrieved.
layout_id	The ID of the layout that was retrieved.
rows	Number of rows in the display array.
columns	Number of columns in the display array.
aspect_ratio	The ratio of the width to the height for the displays within the array.

Response example

{"jsonrpc":"2.0","result": { [{"id": "ENTITYID", "name": "ENTITYNAME", "layout_id": "LAYOUTID", rows: 1, columns: 1, aspect_ratio: 1.7777777777777777}

ClearDisplayArray

Clears all sources from the display array.

Request parameters

displayarrayid	The ID of the display array to clear.

Request example

{"jsonrpc": "2.0", "method": "cleardisplayarray", "params": { "displayarrayid": "ENTITYID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

GetDisplayArrayCount

Returns the total number of Display Arrays.

Request parameters

None.

Request example

{"jsonrpc": "2.0", "method": "getdisplayarraycount", "params": { }, "id": 1}

Response parameters

None.

Response example

{"jsonrpc":"2.0","result":NUMBER_OF_DISPLAYARRAY,"id":1}

GetWindowsForDisplayArray

Gets a list of windows showing on a Display Array. Returns an array with Display Array ID, Layout ID, Window ID, and Source ID. Used to get information needed for the ApplyVideoSource command.

Request parameters

id	The IDs of the windows to retrieve from the Display Array. This ID can be used with ApplyVideoSource.	
Condensed (check box in JSON tester)	Sets response to condensed or uncondensed.	
	False	Sends full response. Default.
	True	Sends condensed response.

Request example

Uncondensed

{"jsonrpc":"2.0","method":"getwindowsfordisplayarray","params":{"id":"8e809ae8-e957-4fea-bdd3-70458b3c214a","condensed":false},"id":31}

Condensed

{"jsonrpc":"2.0","method":"getwindowsfordisplayarray","params":{"id":"1370fd63-79c1-4e1f-b407-c2d4091403f1","condensed":true},"id":43}

Response parameters

Array of windows that contains the displayarrayid, layoutid, sourceid, sourcename, viewindex, and windowid.

Response example

Uncondensed

{"jsonrpc":"2.0","result":{"windows":[{"DisplayArrayID":"1370fd63-79c1-4e1f-b407-c2d4091403f1","LayoutID":"45df47e4-392f-45dd-b760-2fa04c7ba081","WindowID":":0:0:0:1:0:0:0:100:100","Source":"000948aa00ea","SourceName":"SecurityPi","ViewIndex":1},{"DisplayArrayID":"1370fd63-79c1-4e1f-b407-c2d4091403f1","LayoutID":"45df47e4-392f-45dd-b760-2fa04c7ba081","WindowID":":0:1:0:1:0:0:1:100:100","Source":"000948aa03e6","SourceName":"BluRay","ViewIndex":2}]},"id":40}

Condensed

{"jsonrpc":"2.0","result":{"DisplayArrayID":"1370fd63-79c1-4e1f-b407-c2d4091403f1","LayoutID":"45df47e4-392f-45dd-b760-2fa04c7ba081","windows":[{"WindowID":":0:0:0:1:0:0:0:100:100","Source":"000948aa00ea","ViewIndex":1,"SourceName":"SecurityPi"},{"WindowID":":0:1:0:1:0:0:1:100:100","Source":"000948aa03e6","ViewIndex":2,"SourceName":"BluRay"}]},"id":44}

GetWindowsForDisplayArrayCount

Returns the total number of windows currently showing on a display array.

Request parameters

id	The ID of the display array.

Request example

{"jsonrpc": "2.0", "method": "getwindowsfordisplayarraycount", "params": { "id": "9c65fd3f-912b-4f1e-94ec-c20a016fced7" }, "id": 1}

Response parameters

result	The number of windows on the display array.

Response example

{"jsonrpc":"2.0","result":4,"id":1}

GetWindowsForDisplayArrayByRange

Returns a list of windows currently on the display array.

Request parameters

id	The ID of the display array.
startindex	The number of the first display array to retrieve.
endindex	The number of the last display array to retrieve.
Condensed (check box in the JSON Tester)	Sets response to condensed or uncondensed. False — Sends full response. Default. True — Sends condensed response.

Request example

Uncondensed

{"jsonrpc":"2.0","method":"getwindowsfordisplayarraybyrange","params":{"id":"1370fd63-79c1-4e1f-b407-c2d4091403f1","startindex":0,"endindex":5,"condensed":false},"id":52}

Condensed

{"jsonrpc":"2.0","method":"getwindowsfordisplayarraybyrange","params":{"id":"1370fd63-79c1-4e1f-b407-c2d4091403f1","startindex":0,"endindex":5,"condensed":true},"id":50}

Response parameters

None.

Response example

Uncondensed

{"jsonrpc":"2.0","result":{"windows":[{"DisplayArrayID":"1370fd63-79c1-4e1f-b407-c2d4091403f1","LayoutID":"45df47e4-392f-45dd-b760-2fa04c7ba081","WindowID":":0:0:0:1:0:0:0:100:100","Source":"000948aa00ea","SourceName":"SecurityPi","ViewIndex":1},{"DisplayArrayID":"1370fd63-79c1-4e1f-b407-c2d4091403f1","LayoutID":"45df47e4-392f-45dd-b760-2fa04c7ba081","WindowID":":0:1:0:1:0:0:1:100:100","Source":"000948aa03e6","SourceName":"BluRay","ViewIndex":2}]},"id":49}

Condensed

{"jsonrpc":"2.0","result":{"DisplayArrayID":"1370fd63-79c1-4e1f-b407-c2d4091403f1","LayoutID":"45df47e4-392f-45dd-b760-2fa04c7ba081","windows":[{"WindowID":":0:0:0:1:0:0:0:100:100","Source":"000948aa00ea","ViewIndex":1,"SourceName":"SecurityPi"},{"WindowID":":0:1:0:1:0:0:1:100:100","Source":"000948aa03e6","ViewIndex":2,"SourceName":"BluRay"}]},"id":51}

Listener Commands

This section describes the Listener commands. Listener commands are used to receive notifications for subscribed events.

addlistener

Adds a listener for the type specified in the request. Listenertypes are one of the following:

listenertypes

Data I/O	irdata rs232data
Device	deviceadded devicedeleted devicedisconnected deviceupdated
Display Array	displayarrayadded displayarraydeleted displayarrayupdated
Layout	layoutadded layoutappliedtodisplayarray layoutdeleted layoutupdated
System	systemrestart systemshutdown
USB	txusbunpaired usbnotpaired
User	useradded userlockout userlogin userupdated
User Group	usergroupadded usergroupdeleted usergroupupdated

Request parameters

listenertype	The type of listener to subscribe to receive notifications. Refer to *listenertypes*, page *44*.
devices	Valid only for irdata and rs232data listenertypes. All other listenertypes will ignore this parameter. A string array with the device IDs to filter on. If the string is empty, the API will send an event for all rs232data or irdata that is received.

Request example

The following example format can be used for any of the listener commands by replacing the *listenertype* in the example format.

```
{"jsonrpc": "2.0", "method": "addlistener", "params": { "listenertype": "LISTENERTYPE"
}, "id": 1}
```

For IR data and RS232 data only:

```
{"jsonrpc": "2.0", "method": "addlistener", "params": { "listenertype": "rs232data",
"deviceids" : ["DEVICEID "," DEVICEID "] }, "id": 1}
{"jsonrpc": "2.0", "method": "removelistener", "params": { "listenertype":
"rs232data", "deviceids" : ["DEVICEID "," DEVICEID "] }, "id": 1}
```

Response parameters

None. See *Error Messages* (page *9).*

Response example

{"jsonrpc":"2.0","result":{},"id":1}

removelistener

This method unregisters from notifications.

Request Parameters

listenertype	The type of listener to unsubscribe.
devices	Valid only for irdata and rs232data listenertypes. All other listenertypes will ignore this parameter. A string array with the device IDs to filter on. If the string is empty, the API will send an event for all rs232data or irdata that is received.

Request example

The following example format can be used for any of the listener commands by replacing the *listenertype* in the example format.

{"jsonrpc": "2.0", "method": "removelistener", "params": { "listenertype": "LISTENERTYPE" }, "id": 1}

For IR data and RS232 data only:

{"jsonrpc": "2.0", "method": "addlistener", "params": { "listenertype": "rs232data", "deviceids" : ["DEVICEID "," DEVICEID "] }, "id": 1}
{"jsonrpc": "2.0", "method": "removelistener", "params": { "listenertype": "rs232data", "deviceids" : ["DEVICEID "," DEVICEID "] }, "id": 1}

Response parameters

None. See *Error Messages* (page *9).*

Response example

request:

{"jsonrpc": "2.0", "method": "removelistener", "params": { "listenertype": "LISTENERTYPE" }, "id": 1}

response:

{"jsonrpc":"2.0","result":{},"id":1}

removealllisteners

The top level `removealllisteners` method completely unregisters all notifications currently registered to the client. It is recommended that this is called just before a client exits.

Request parameters

None.

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

request:

{"jsonrpc": "2.0", "method": "removealllisteners", "params": {}, "id": 1}

response:

{"jsonrpc":"2.0","result":{},"id":1}

System Commands

This section describes the system commands. System commands do not have any parameters and the response is the success status.

RestartSystem

Restarts the controller.

Request parameters

None.

Request example

{"jsonrpc": "2.0", "method": "restartsystem", "params": { }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

ShutdownSystem

Powers off the controller. The associated devices attached to the controller are not powered off.

Request parameters

None.

Request example

{"jsonrpc": "2.0", "method": "shutdownsystem", "params": { "id": "ENTITYID" }, "id": 1}

Response parameters

success	True if successful. False if Terra is unable to complete the command.

Response example

{"jsonrpc":"2.0","result":{"success": true},"id":1}

Index

Corporate offices

USA – Cypress
ph: 714-236-8610

Canada – Kitchener
ph: 519-744-8005

Consultant offices

Italy
ph: +39 (0) 2 9902 1161

Worldwide offices

Australia
ph: +61 (0) 7 3624 4888

Brazil
ph: +55 (11) 2548 4753

China (Beijing)
ph: +86 10 6561 0240

China (Shanghai)
ph: +86 21 6278 7708

Eastern Europe and
Russian Federation
ph: +36 (0) 1 47 48 100

France
ph: +33 (0) 1 41 21 44 04

Germany
ph: +49 2161 664540

India
ph: +91 (080) 6708 9999

Japan (Tokyo)
ph: 81 3 3599 7481

Korea (Seoul)
ph: +82 2 702 1601

Republic of South Africa
ph: +27 (0)11 510 0094

Singapore
ph: +65 6877-8737

Spain
ph: + 34 91 633 9990

United Arab Emirates
ph: +971 4 3206688

United Kingdom
ph: +44 (0) 118 977 8000