

CVIPv2

Camera Video Interface Protocol

XeronVision Inc.

Ver.2

09. Sep. 2015

Change History

Issue	Date	Description
0.1	2-Jan-09	Draft
0.2	16-Jan-09	Add User and Camera Management
0.3	30-Apr-09	Add Sequences
0.4	22-May-09	Add Map
0.5	23-June-09	Add PTZ service
0.6	22-July-09	Add Event service
0.7	04-Aug-09	Add acknowledge response, and playback control
0.8	22-Jan-10	Revise event notification, acknowledgement
0.9	03-Feb-10	Support intercom device
1.0	12-March-10	Event customization and event playback query.
1.1	7-Sep-15	Add user login encryption

Table of Contents

Table of Contents	i
Introduction	1
Protocol Basics	2
1. Client-Server Architecture	2
2. Request – Response Pairs	2
3. Event	2
4. XML-RPC	3
Protocol Use Case Diagram	3
1. Device Protocol Use Case	3
2. NVR Protocol Use Case	4
3. Master Server Protocol Use Case	4
CVIPv2 Protocol Use Cases	5
1. Common Protocol Use Case	5
1) Discover Device	5
2) Discover Services	5
3) Configure Device	6
4) Manage Event Service	7
5) Update Device Firmware	9
2. Device Protocol Use cases:	9
1) Manage Serial Port Service	9
2) Manage Streaming Service	10
3) Manage Display Service	12
3. NVR Protocol Use Cases	13
1) Manage Recording Service	13
2) Manage Playback Service	15
3) Manage Media Service	16
4. Master Server Protocol Use Cases	18
1) Manage Device Profile Service	18
2) Manage Device Entity Service	19
3) Manage User Service	23
4) Manage Application Service	27
CVIPv2 command details	29
1. CVIPv2 Message Transportation Protocol	29
2. CVIPv2 Message Header	29
3. Common Protocol	30

1) Acknowledge Response	30
2) Discover Device	30
Discover Services	32
3) Configure Device	40
4) Update Device Firmware	41
4. Device Protocol	44
1) Manage Serial Port Service	44
2) Manage Streaming Service	46
3) Manage Display Service	47
4) Manage Analytics Service	49
5) Manage PTZ Service	51
5. NVR Protocol	54
1) Manage Recording Service	54
2) Manage Playback Service	56
3) Manage Media Service	61
4) Manage Camera Event Service	62
6. Master Server Protocol	64
1) Manage Device Profile Service	64
2) Manage Device Entity Service	66
3) Manage User Service	73
4) Manage Application Service	78
Service and Properties	81
1. Service	81
2. Properties	82

Introduction

This document introduces an IP based configuration and control protocol CVIPv2 (Camera Video Interface Protocol), which facilitates the management of a large amount of devices in the security business. Using this protocol, a client side application could easily configure an encoder, start a streamer, stop a recorder, listen to an event on video motion, display a stream or even update the firmware.

To provide an easy, scalable, flexible and self-descriptive protocol, CVIPv2 is a TCP XML-RPC based protocol, which is an XML wrapper that contains the XML based request. The protocol consists of three parts: Device protocol, NVR protocol and Master Server protocol.

Protocol Basics

1. Client-Server Architecture

Client-server is a network architecture which separates the client from the server. It is intended to provide a scalable architecture, whereby each computer or processes on the network are either a client or a server. Each instance of the client software can send requests to a server. (i.e.: a review station to a NVR, or a control center to a Master Server).

2. Request – Response Pairs

Most CVIPv2 operations use pairs of messages in the format “Request - Response”, which is illustrated in Figure 1. A CVIPv2 client requests some information and CVIPv2 server returns the information requested. Some commands ask the device to perform an action rather than to provide information (e.g. starts a streamer). For such commands, CVIPv2 still uses a Request-Response approach, but in this case the response simply indicates whether the command was performed successfully.

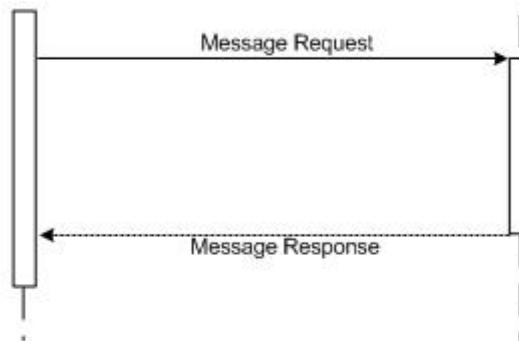


Figure 1. Request-Response pairs

3. Event

An event is an asynchronous notification sent by the server to the client to inform that something append. This is the only message issued from the server and also the only asynchronous element of the protocol.

4. XML-RPC

XML-RPC is a XML container, which wraps a Remote Procedure Command. There are three types of message defined in CVIPv2, method call, method response and event.

- 1) Method call is an XML wrapper that contains the XML based request sent from the client application to the server.
- 2) Method response is an XML wrapper that contains the XML based response sent to the client application from the server.
- 3) Event is an XML wrapper that contains an asynchronous XML based event sent by the server to the client side application.

Protocol Use Case Diagram

1. Device Protocol Use Case

The following use case diagram illustrates a typical scenario for the communication between a review station client and edge device.

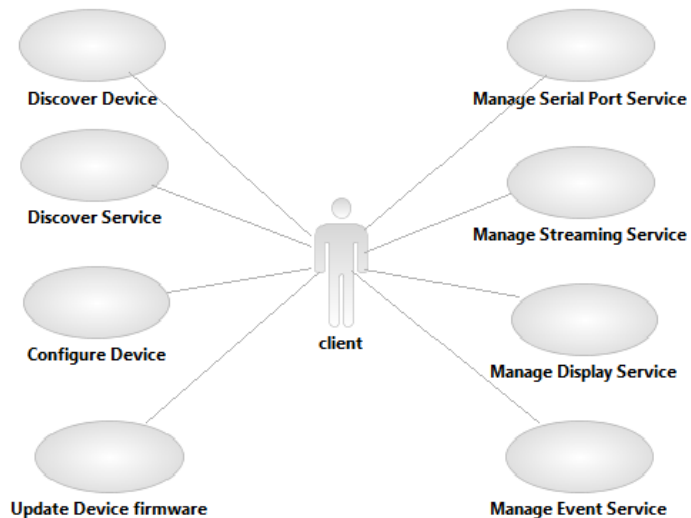


Figure 2. Device protocol use case diagram

2. NVR Protocol Use Case

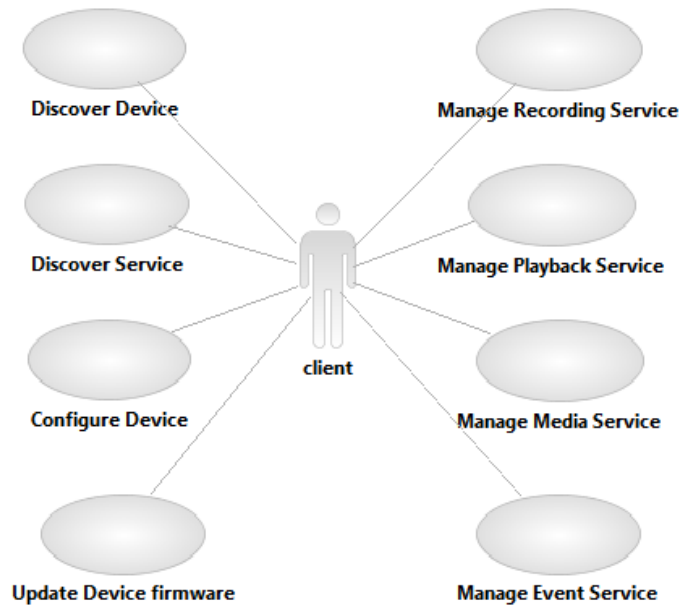


Figure 3. NVR Protocol Use Case diagram

3. Master Server Protocol Use Case

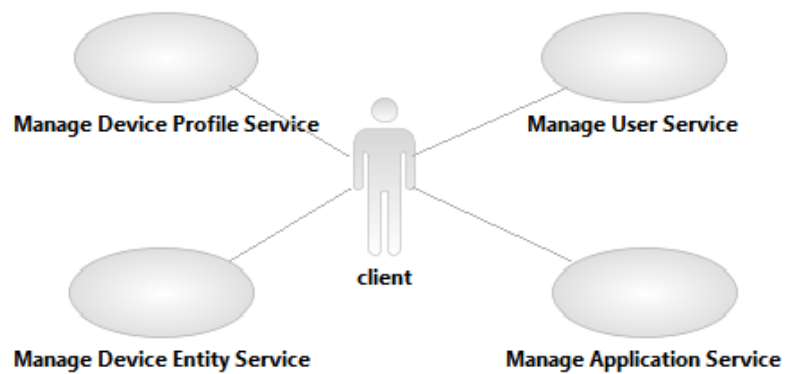


Figure 4. Master Server Protocol Use cases diagram

CVIPv2 Protocol Use Cases

1. Common Protocol Use Case

1) Discover Device

The discover Device use case consist of finding a unit on the network in order to get the IP address and port. Once those values are known, it is possible to perform any other use case.

Discover Device use Case

Context	A device is added to the network and the client wants to discover the device IP address and control port.	
Pre-conditions	• The device is booted up. • The device is connected to the same network segment as the client.	
Post-conditions	Success The device IP address and port is returned.	
Scenario	Step	Action
	1	The client performs a CVIP discovery.
	2	The device answers with his current IP address and control port.
	3	The client uses the provided IP and port to communicate with the device

2) Discover Services

Discover Services use Case

Context	A device is discovered on the network and the client wants to discover the services of the device.	
Pre-conditions	• The device is booted up. • The client is configured with the device IP address and CVIP port.	
Post-conditions	Success A list of all the available services is returned.	
Scenario	Step	Action
	1	The client connects to the unit.
	2	The client asks for the list of available event

	3	The device returns the list of available event group
	4	For each group, the client asks for a list of available event
	5	The device returns the list of available event on a per group basis
	6	The client requests for the list of available services.
	7	The device returns the list of available services
	8	For each service, the client asks for the service description that provide functions and properties support.
	9	The device returns the description on a per service basis
	10	For each service that support properties, the client asks for a list of properties
	11	The device returns the list of properties on a per service basis
	12	For each service that support custom functions, the client asks for a list of custom function.
	13	The device returns the list of custom function on a per service basis
	14	For each custom function, the client asks for a list of function parameters
	15	The device returns the list of function parameter on a per function basis

3) Configure Device

3.1 Network configuration

Configure Device use Case: Network

Context

The client can view and modify the network configuration.

Pre-conditions

- A Device with a valid configuration

Post-conditions

Success

The new network configuration value is applied

Scenario

Step Action

- | Step | Action |
|------|---|
| 1 | The client requests the current network configuration value. |
| 2 | The device returns the current network configuration value with details: <ul style="list-style-type: none"> • XMLRPC id • Friendly name |

	• Validation information • Access right • Value unit
3	The client specifies the new network configuration value within the valid range.
4	The device applies the new network configuration value.
5	The device returns an acknowledgement

3.2 Video configuration

Configure device use Case: Video

Context	The client can view and modify the video configuration of the device.												
Pre-conditions	• A device with video input												
Post-conditions	Success The new video configuration is applied												
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>The client requests the current video value.</td></tr> <tr> <td>2</td><td>The device returns the current video value with details: • XMLRPC id • Friendly name • Validation information • Access right • Value unit </td></tr> <tr> <td>3</td><td>The client specifies the new video value within the valid range.</td></tr> <tr> <td>4</td><td>The device applies the new video value.</td></tr> <tr> <td>5</td><td>The device returns an acknowledgement</td></tr> </table>	Step	Action	1	The client requests the current video value.	2	The device returns the current video value with details: • XMLRPC id • Friendly name • Validation information • Access right • Value unit	3	The client specifies the new video value within the valid range.	4	The device applies the new video value.	5	The device returns an acknowledgement
Step	Action												
1	The client requests the current video value.												
2	The device returns the current video value with details: • XMLRPC id • Friendly name • Validation information • Access right • Value unit												
3	The client specifies the new video value within the valid range.												
4	The device applies the new video value.												
5	The device returns an acknowledgement												

4) Manage Event Service

4.1 Request event notification

Manage Event Service use Case: Request

Context	The client want to be notify when a specific event occur on the device.
----------------	---

Pre-conditions	• A device with a event service								
Post-conditions	Success The client is added to the list of notification to be performed when an event triggers.								
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>The client requests for a specific event notification.</td></tr> <tr> <td>2</td><td>The device adds the client to the notification list</td></tr> <tr> <td>3</td><td>The device returns an acknowledgement</td></tr> </table>	Step	Action	1	The client requests for a specific event notification.	2	The device adds the client to the notification list	3	The device returns an acknowledgement
Step	Action								
1	The client requests for a specific event notification.								
2	The device adds the client to the notification list								
3	The device returns an acknowledgement								

4.2 Remove event notification

Manage Event Service use Case: Remove									
Context	The client can request to be removed from the event notification list.								
Pre-conditions	• A device with a event service • The client is part of a notification list								
Post-conditions	Success The client is removed from the list of notification to be performed when an event triggers.								
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>The client requests to be removed from a specific event notification.</td></tr> <tr> <td>2</td><td>The device removes the client from the notification list</td></tr> <tr> <td>3</td><td>The device returns an acknowledgement</td></tr> </table>	Step	Action	1	The client requests to be removed from a specific event notification.	2	The device removes the client from the notification list	3	The device returns an acknowledgement
Step	Action								
1	The client requests to be removed from a specific event notification.								
2	The device removes the client from the notification list								
3	The device returns an acknowledgement								

4.3 Receive event notification

Manage Event Service use Case: Receive									
Context	The device notifies the client on an event.								
Pre-conditions	• A device with an event service • A client in the notification list								
Post-conditions	Success The client receives a notification								
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>The device detects an event.</td></tr> <tr> <td>2</td><td>The device sends a notification based on the notification list</td></tr> <tr> <td>3</td><td>The client receives the notification</td></tr> </table>	Step	Action	1	The device detects an event.	2	The device sends a notification based on the notification list	3	The client receives the notification
Step	Action								
1	The device detects an event.								
2	The device sends a notification based on the notification list								
3	The client receives the notification								

5) Update Device Firmware

Update Device firmware use Case

Context	The client can update the device firmware.	
Pre-conditions	The device runs an old firmware version	
Post-conditions	Success The device reboots and run the new firmware	
Scenario	Step	Action
	1	The client requests for a firmware update.
	2	The client specifies the new firmware to use.
	3	The device downloads the new firmware
	4	The device installs the new firmware
	5	The device reboots

2. Device Protocol Use cases:

1) Manage Serial Port Service

1.1 Start streaming

Manage Serial Port Service use Case: start streaming

Context	A device with serial port service can start a streaming session to any serial port device.	
Pre-conditions	The limit of streaming is not reached	
Post-conditions	Success A streaming session has been started.	
Scenario	Step	Action
	1	The client requests to start a serial port streaming session.
	2	The client specifies the data source for the streaming session.
	3	The device returns an acknowledgement

1.2 Stop streaming

Manage Serial Port Service use Case: stop streaming

Context	A device with serial port service can stop a current serial port streaming session.	
Pre-conditions	The serial port streaming session to stop is currently running	
Post-conditions	Success A serial port streaming session has been stopped.	
Scenario	Step	Action
	1	The client requests to stop a current serial port streaming session.
	2	The device returns an acknowledgement

1.3 List streaming

Manage Serial Port Service use Case: list streaming

Context	A device with serial port service can list the current serial port streaming session.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of current serial port streaming session.
	2	The device returns the list of all the current serial port streaming session with all the details for each session.

2) Manage Streaming Service

2.1 Start streaming

Manage Streaming Service use Case: start streaming

Context	A device with streaming service can start a streaming session from any media source.	
Pre-conditions	The limit of streaming is not reached	
Post-	Success	

conditions	A streaming session has been started.	
Scenario	Step	Action
	1	The client requests to start a streaming session.
	2	The client specifies the media source for the streaming session.
	3	The device returns an acknowledgement

2.2 Stop streaming

Manage Streaming Service use Case: stop streaming		
Context	A device with streaming service can stop a current streaming session.	
Pre-conditions	The streaming session to stop is currently running	
Post-conditions	Success A streaming session has been stopped.	
Scenario	Step	Action
	1	The client requests to stop a current streaming session.
	2	The client specifies the media source for the current streaming session to stop.
	3	The device returns an acknowledgement

2.3 List streaming

Manage Streaming Service use Case: list streaming		
Context	A device with streaming service can list the current streaming session.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of current streaming session.
	2	The device returns the list of all the current streaming session with all the media details for each session.

2.4 Get a key frame

Manage Streaming Service use Case: get a key frame

Context	A device with streaming service can get a key frame from a current streaming session.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a key frame from a current streaming session.
	2	The device returns the next key frame from the current streaming session.

3) Manage Display Service

3.1 Start Display

Manage Playback Display Service use Case: Start

Context	A device with display service can start display session from any media source.	
Pre-conditions	The display limit has not been reached	
Post-conditions	Success A display session has been started.	
Scenario	Step	Action
	1	The client requests to start a display session.
	2	The client specifies the media source for the display session.
	3	The device returns an acknowledgement

3.2 Stop Display

Manage Display Service use Case: Stop

Context	A device with display service can stop a current display session.	
Pre-conditions	The display session to stop is currently running	

Post-conditions	Success A display session has been stopped.	
Scenario	Step	Action
	1	The client requests to stop a current display session.
	2	The client specifies the media source for the current display session to stop.
	3	The device returns an acknowledgement

3.3 List Display

Manage Display Service use Case: List		
Context	A device with display service can list the current display session.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of current display session.
	2	The device returns the list of all the current display session with all the media details for each session.

3. NVR Protocol Use Cases

1) Manage Recording Service

1.1 Start recording

Manage Recording Service use Case: Start	
Context	A NVR with recording service can start a recording session from any media source with a schedule. A schedule consists of a timeframe
Pre-conditions	The limit of recording is not reached
Post-conditions	Success A recording session has been started.

Scenario	Step	Action
	1	The client requests to start a recording session with a schedule.
	2	The client specifies the media source for the recording session.
	3	The NVR returns an acknowledgement

1.2 Stop recording

Manage Recording Service use Case: Stop

Context	A NVR with recording service can stop a current recording session.	
Pre-conditions	The recording session to stop is currently running	
Post-conditions	Success A recording session has been stopped.	
Scenario	Step	Action
	1	The client requests to stop a current recording session.
	2	The client specifies the media source for the current recording session to stop.
	3	The NVR returns an acknowledgement

1.3 List recording

Manage Recording Service use Case: List Recording

Context	A NVR with recording service can list the current recording session.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of current recording session.
	2	The NVR returns the list of all the current recording session with all the media details for each session.

1.4 Update a schedule

Manage Recording Service use Case: UpdateSchedule

Context	The client sends a schedule to update a current running recording session on the NVR.	
Pre-conditions	N/A	
Post-conditions	Success The schedule is applied according to the timeframe	
Scenario	Step	Action
	1	The client sends the schedule to the NVR.
	2	The NVR apply the schedule to the current recording session.

2) Manage Playback Service

2.1 Start playback

Manage Playback Service use Case: Start

Context	A NVR with playback service can start a playback session from any media source.	
Pre-conditions	The limit of playback is not reached	
Post-conditions	Success A playback session has been started.	
Scenario	Step	Action
	1	The client requests to start a playback session.
	2	The client specifies the media source for the playback session.
	3	The NVR returns an acknowledgement

2.2 Stop playback

Manage Playback Service use Case: Stop

Context	A NVR with playback service can stop a current playback session.	
Pre-conditions	The playback session to stop is currently running	

Post-conditions	Success A playback session has been stopped.	
Scenario	Step	Action
	1	The client requests to stop a current playback session.
	2	The client specifies the media source for the current playback session to stop.
	3	The NVR returns an acknowledgement

2.3 List playback

Manage Playback Service use Case: List Playback		
Context	A NVR with playback service can list the current playback session.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of current playback session.
	2	The NVR returns the list of all the current playback session with all the media details for each session.

3) Manage Media Service

3.1 Time based browsing

Manage Media Service use Case: Time based		
Context	A NVR with recording service can return the list of recorded media source within a specified time interval.	
Pre-conditions	At least one recording session has been running previously	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of recorded media source.
	2	The client specifies the time interval.

- 3 The DVR returns a list of media source recorded inside the time interval

3.2 Media source based browsing

Manage Media Service use cases: Media source based

Context	A NVR with recording service can return the list of recorded media source based on media source criteria.	
Pre-conditions	At least one recording session has been running previously	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of recorded media source.
	2	The client specifies the media source criteria.
	3	The NVR returns a list of media source recorded matching the media source criteria specified

3.3 Event based browsing

Manage Media Service use Case: Event based

Context	A NVR with recording service can return the list of recorded media source during an event.	
Pre-conditions	- At least one recording session has been running previously - An event has occurred during recording	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests for a list of recorded media source.
	2	The client specifies the event.
	3	The NVR returns a list of media source recorded during the event

3.4 Export Recorded Media

Manage Media Service use Case: Download recorded media

Context	The client can download from a NVR any recorded media.
----------------	--

Pre-conditions	The NVR contains recorded media	
Post-conditions	Success The media has been uploaded from the NVR to the client	
Scenario	Step	Action
	1	The client requests for a media to be downloaded.
	2	The NVR transfers the media to the client.

4. Master Server Protocol Use Cases

1) Manage Device Profile Service

The profile could be serial port profile, video profile, audio profile and schedule profile.

1.1 List <Profile>

Manage Device Profile Service use Case: list<Profile>		
Context	The Master Server can list profiles.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to get a list of profiles description.
	2	The Master Server returns the list

1.2 Add <Profile>

Manage Device Profile Service use Case: add<Profile>		
Context	Master Server can add a new profile.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to add a new profile.
	2	Master Server adds the profile and returns an acknowledgement and profile ID.

1.3 Update <Profile>

Manage Device Profile Service use Case: update<Profile>

Context	Master Server can update an existing profile.	
Pre-conditions	The updating profile exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to update an existing profile.
	2	Master Server updates the profile and returns an acknowledgement.

1.4 Delete <Profile>

Manage Device Profile Service use Case: delete<Profile>

Context	Master Server can delete an existing profile.	
Pre-conditions	The deleting profile exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to delete an existing profile.
	2	Master Server deletes the profile and returns an acknowledgement.

2) Manage Device Entity Service

The entity could be Camera, Edge or Monitor

2.1 List <entity>Groups

Manage Device Entity Service use Case: list<entity>Groups

Context	The Master Server can list entity group description.
Pre-conditions	N/A
Post-conditions	N/A

Scenario	Step	Action
	1	An authorized client requests to get a list of entity group description.
	2	The Master Server returns the list

2.2 Add<entity>Group

Manage Device Entity Service use Case: add<entity>Group

Context	Master Server can add a new entity group.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to add a new entity group with specified group properties.
	2	Master Server adds the group and returns an acknowledgement and group ID.

2.3 Update<entity>Group

Manage Device Entity Service use Case: update<entity>Group

Context	Master Server can update an existing entity group property.	
Pre-conditions	The updating group exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to update an existing entity group with group ID and updated group properties.
	2	Master Server updates the group and returns an acknowledgement.

2.4 Delete<entity>Group

Manage Device Entity Service use Case: delete<entity>Group

Context	Master Server can delete an existing entity group.
Pre-conditions	The deleting group exists
Post-	N/A

conditions**Scenario****Step Action**

- | | |
|---|--|
| 1 | An authorized client requests to delete an existing group with specified group ID. |
| 2 | Master Server deletes the group and returns an acknowledgement. |

2.5 List <entity>**Manage Device Entity Service use Case: list<entity>****Context**

The Master Server can list entity description.

Pre-conditions

N/A

Post-conditions

N/A

Scenario**Step Action**

- | | |
|---|--|
| 1 | An authorized client requests to get a list of entity description with different page. |
| 2 | The Master Server returns the list |

2.6 Add<entity>**Manage Device Entity Service use Case: add<entity>****Context**

Master Server can add a new entity.

Pre-conditions

N/A

Post-conditions

N/A

Scenario**Step Action**

- | | |
|---|---|
| 1 | An authorized client requests to add a new entity with specified entity properties. |
| 2 | Master Server adds the entity and returns an acknowledgement and entity ID. |

2.7 Update<entity>**Manage Device Entity Service use Case: update<entity>**

Context	Master Server can update an existing entity.	
Pre-conditions	The updating entity exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to update an entity with entity ID and updated user properties.
	2	Master Server updates the entity and returns an acknowledgement.

2.8 Delete<entity>

Manage Device Entity Service use Case: delete<entity>		
Context	Master Server can delete an existing entity.	
Pre-conditions	The deleting entity exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to delete an existing entity with specified entity ID.
	2	Master Server deletes the entity and returns an acknowledgement.

2.9 List<entity>Service

Manage Device Entity Service use Case: list<entity>Service		
Context	Master Server can list the service of any entity.	
Pre-conditions	The entity exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to list an existing service of certain entity.
	2	Master Server returns the service list.

2.10 Update<entity>Service

Manage Device Entity Service use Case: Update<entity>Service

Context	Master Server can update the service of any entity.	
Pre-conditions	The entity exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to update certain service of an entity.
	2	Master Server returns the acknowledgement.

3) Manage User Service

3.1 LoginUser

Manage User Service use Case: loginUser

Context	Master Server can authorize an existing user login.	
Pre-conditions	The authorizing user exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	The client requests to login an existing user with specified user name and password.
	2	Master Server validates the user and returns an application GUID, which could be used for getting device information.

3.2 LogoutUser

Manage User Service use Case: logoutUser

Context	Master Server can permit an authorized user logout.	
Pre-conditions	The authorizing user exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to logout Master

	server with specified application GUID.
2	Master Server permit the logout processing .

3.3 List Groups

Manage User Service use Case: listGroups

Context	The Master Server can list user group description.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to get a list of user group description.
	2	The Master Server returns the list

3.4 AddGroup

Manage User Service use Case: addGroup

Context	Master Server can add a new user group.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to add a new user group with specified group properties.
	2	Master Server adds the user group and returns an acknowledgement and group ID.

3.5 UpdateGroup

Manage User Service use Case: updateGroup

Context	Master Server can update an existing user group property.	
Pre-conditions	The updating user group exists	
Post-conditions	N/A	
Scenario	Step	Action

- 1 An authorized client requests to update an existing user group with group ID and updated group properties.
- 2 Master Server updates the group and returns an acknowledgement.

3.6 DeleteGroup

Manage User Service use Case: deleteGroup

Context	Master Server can delete an existing user group.	
Pre-conditions	The deleting group exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to delete an existing user group with specified group ID.
	2	Master Server deletes the group and returns an acknowledgement.

3.7 List Users

Manage User Service use Case: listUsers

Context	The Master Server can list user description.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to get a list of user description.
	2	The Master Server returns the list

3.8 AddUser

Manage User Service use Case: addUser

Context	Master Server can add a new user.	
Pre-conditions	N/A	
Post-conditions	N/A	

Scenario	Step	Action
	1	An authorized client requests to add a new user with specified user properties.
	2	Master Server adds the user and returns an acknowledgement and user ID.

3.9 UpdateUser

Manage User Service use Case: updateUser

Context	Master Server can update an existing user.	
Pre-conditions	The updating user exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to update a user with user ID and updated user properties.
	2	Master Server updates the user and returns an acknowledgement.

3.10 DeleteUser

Manage User Service use Case: deleteUser

Context	Master Server can delete an existing user.	
Pre-conditions	The deleting user exists	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to delete an existing user with specified user ID.
	2	Master Server deletes the user and returns an acknowledgement.

3.11 Mount<entity>

The entity could be Camera, Recorder or Monitor

Manage User Service use Case: mount<entity>

Context	Master Server can mount an entity to an existing user.
Pre-	The assigning entity exists

conditions		
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to mount an entity to an existing user.
	2	Master Server returns an acknowledgement.

3.12 Unmount<entity>

The entity could be Camera, Recorder or Monitor

Manage User Service use Case: unmount<entity>		
Context	Master Server can unmount an entity to an existing user.	
Pre-conditions	The entity is mounted	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to unmount an entity to an existing user.
	2	Master Server returns an acknowledgement.

4) Manage Application Service

The application could be PTZ control, streaming, playback, recording and event action.

4.1 List<application>

Manage Application Service use Case: list<Application>		
Context	Master Server can list the application with a specified entity.	
Pre-conditions	N/A	
Post-conditions	N/A	
Scenario	Step	Action
	1	An authorized client requests to list the applications with a specified entity.
	2	Master Server lists the application description and returns an acknowledgement.

4.2 Add<application>

Manage Application Service use Case: add<Application>

Context	Master Server can add a new application.						
Pre-conditions	N/A						
Post-conditions	N/A						
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>An authorized client requests to add a new application to a specified camera.</td></tr> <tr> <td>2</td><td>Master Server adds the application and returns an acknowledgement and application ID.</td></tr> </table>	Step	Action	1	An authorized client requests to add a new application to a specified camera.	2	Master Server adds the application and returns an acknowledgement and application ID.
Step	Action						
1	An authorized client requests to add a new application to a specified camera.						
2	Master Server adds the application and returns an acknowledgement and application ID.						

4.3 Update<application>

Manage Application Service use Case: update<Application>

Context	Master Server can update an existing application.						
Pre-conditions	The updating application exists						
Post-conditions	N/A						
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>An authorized client requests to update an application.</td></tr> <tr> <td>2</td><td>Master Server updates the application and returns an acknowledgement.</td></tr> </table>	Step	Action	1	An authorized client requests to update an application.	2	Master Server updates the application and returns an acknowledgement.
Step	Action						
1	An authorized client requests to update an application.						
2	Master Server updates the application and returns an acknowledgement.						

4.4 Delete<application>

Manage Application Service use Case: delete<Application>

Context	Master Server can delete an existing application.						
Pre-conditions	The deleting application exists						
Post-conditions	N/A						
Scenario	<table> <tr> <th>Step</th><th>Action</th></tr> <tr> <td>1</td><td>An authorized client requests to delete an existing application with specified application ID.</td></tr> <tr> <td>2</td><td>Master Server deletes the application and returns an acknowledgement.</td></tr> </table>	Step	Action	1	An authorized client requests to delete an existing application with specified application ID.	2	Master Server deletes the application and returns an acknowledgement.
Step	Action						
1	An authorized client requests to delete an existing application with specified application ID.						
2	Master Server deletes the application and returns an acknowledgement.						

CVIPv2 command details

1. CVIPv2 Message Transportation Protocol

Except Ping request-response message, CVIPv2 message connection should be established via TCP protocol. Ping message could be established via TCP protocol as well as UDP protocol. The purpose is to make it possible that the client could discover all devices on the network by sending a single command to a specific address or a broadcast address such as 255.255.255.255.

2. CVIPv2 Message Header

Each message in the CVIPv2 protocol begins with a header containing its type, a transaction number, and total size in bytes. The principle behind transaction numbers is simple: A request and its corresponding response share a common (and unique) number. Transaction numbers allow the application to know which incoming responses correspond to which outgoing requests, and help discard irrelevant messages. The most significant bit of the transaction number indicates if the message is a request (0) or a response(1). The other 15 bits represent the transaction number itself.

CVIP Header 2.0		
Size	Description	Example
2	Protocol Version	0x0201
	The version of the CVIP protocol used in CVIPv2: 0x0201	
1	Message Type	0x00
	The type of the message only used in Discovery Ping message	
2	Transaction Number	0x0001
	The transaction number of the message.	
4	Packet Size	0x000000 0x28
	The size of the message (in bytes), including this header.	
The example shows the header for a Request message whose size is 40 bytes.		

3. Common Protocol

1) Acknowledge Response

In general, the acknowledge response is to acknowledge a request message.

The name is the same as the request message	
Response:	
Field	Details
ErrorCode	The error code to indicate whether the request is acknowledged successfully.

Error Code	Description
0x01000000 Success	The request operation successes
0x01000001 Fail	The request operation fails
0x01000003 Busy	The server is too busy to process request
0x01000004 Invalid	The syntax of the message is invalid

2) Discover Device

CVIP Ping Request

The Ping Request message asks all units to identify themselves.

CVIP Ping Request	
Size	Description
9	CVIP Header 2.0
	Message Type = CVIP_MSG_TYPE_D_PING_REQ = 0x01
4	IP Address
	The IP address of the client side application.
2	CVIP Port
	The CVIP port of the client side application.

1	Connection Type
	The type of CVIP connection to use. CVIP_UDP_UNICAST = 0x01 CVIP_UDP_MULTICAST = 0x02 CVIP_UDP_BROADCAST = 0x10 CVIP_TCP = 0x0C

CVIP Ping Response

After receiving a Ping Request, each server sends back a Ping Response message.

CVIP Ping Response	
Size	Description
9	CVIP Header 2.0
	Message Type = CVIP_MSG_TYPE_D_PING_RESP = 0x01
4	IP Address
	The IP address of the DVR.
2	CVIP Port
	The CVIP port of the DVR.
1	Connection Type
	The type of CVIP connection to use.
16	Unit GUID
	GUID of the unit.
2	Manufacturer ID
	The identifier of the company that manufactured the unit.
2	Product Type
	The type of the unit, as part of its manufacturer's product line.
1	Status
	The indication of whether the unit has rebooted (1) or not (0) since the last request.

Note: Default & common CVIP Port

The default CVIP port of all CVIP servers is 7110. That CVIP port can be changed in the configuration of a device at any time (the change takes effect only after the unit reboots, though). The CVIP port cannot be used for any other purposes in a CVIP device.

But if the user does not know or does not remember the CVIP port of a device, CVIP supports an option feature. It opens a second port, called the “common port”. On the common port, CVIP accepts but one command, CVIP Ping Request and ignore everything else. When a CVIP Ping Request is received on the common port, using its own CVIP port, the CVIP device send a CVIP Ping Response back. Then the user could know the current CVIP port. The common port is 9990.

Discover Services

The client send CVIPv2 request in order to discover all the device available services

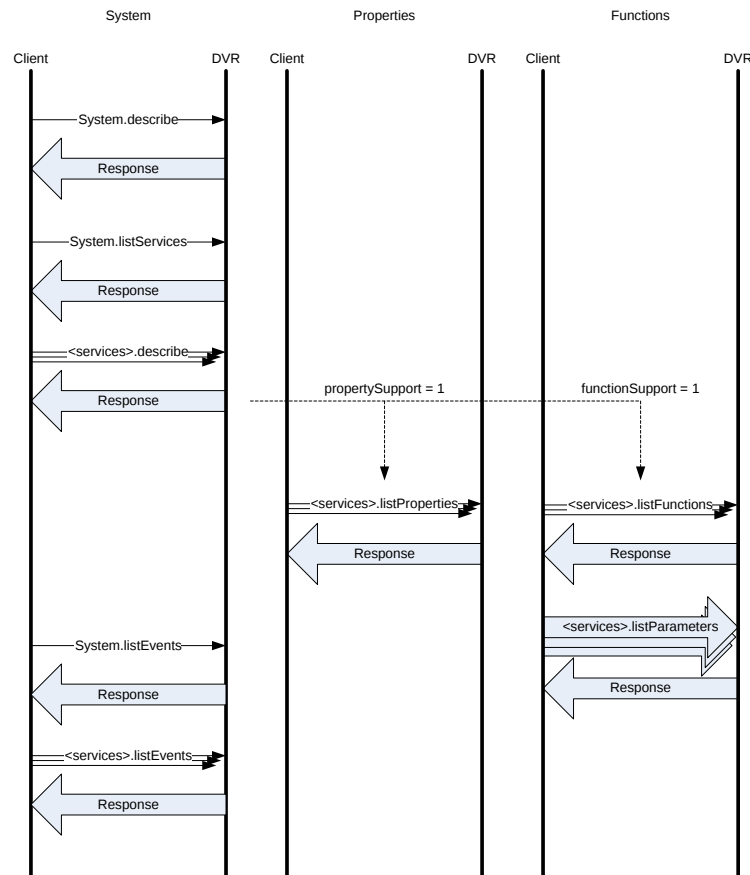


Figure 1 – Discover available services

1. *System.describe*

System.Describe	
Response:	
Field	Details
id	This is the XML unique name of the service.
name	This is the unit friendly name
version	This is the unit version.
propertySupport	Reserved
functionSupport	Reserved

[System.Describe Example]

call

```
<methodCall>
<methodName>system.describe</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><struct>
<member><name>id</name><value><string>NVR_I</string></value></member>
<member><name>name</name><value><string> NVR_I </string></value></member>
<member><name>version</name><value><string>1.0</string></value></member>
<member><name>propertySupport</name><value><uint>0</uint></value></member>
<member><name>functionsSupport</name><value><uint>0</uint></value></member>
</struct></value>
</param></params>
</methodResponse>
```

2. *System.listServices*

This call returns a list of available services

system.listServices	
Response:	
Field	Details
<services>	This is the name of an available services.

[System.listServices Example]

call

```
<methodCall>
<methodName>system.listServices</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><array><data>
<value><string>cvip</string></value>
<value><string>cviNetwork</string></value>
<value><string>cviNetwork.1</string></value>
<value><string>cviNetwork.2</string></value>
<value><string>cviNvr</string></value>
</data></array></value>
</param></params>
</methodResponse>
```

3. System.listEvents

This call returns a list of available event group.

system.listEvents	
Response:	
Field	Details
id	This is the XML unique name of the event.
name	This is the event friendly name

[System.listEvents Example]

call

```
<methodCall>
<methodName>system.listEvents</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><array><data>
<value><struct>
<member><name>id</name><value><string>cvipUnit</string></value></member>
<member><name>name</name><value><string>Unit Events</string></value></member>
</struct></value>
<value><struct>
<member><name>id</name><value><string>cviVideo</string></value></member>
<member><name>name</name><value><string>Video Events</string></value></member>
</struct></value>
</data></array></value>
</param></params>
</methodResponse>
```

4. <Service>.listEvents

This call returns a list of available event for this service.

<services>.listEvents	
Response:	
Field	Details
id	This is the XML unique name of the event.
name	This is the event friendly name

[<Service>.listEvents Example]

call

```
<methodCall>
<methodName>cviVideo.listEvents</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><array><data>
<value><struct>
<member><name>id</name><value><string>FRAME_SKIP</string></value></member>
<member><name>name</name><value><string>Video Frame skipped</string></value></member>
</struct></value>
<value><struct>
<member><name>id</name><value><string>VIDEO_LOST</string></value></member>
<member><name>name</name><value><string>Lost Video signal</string></value></member>
</struct></value>
</data></array></value>
</param></params>
</methodResponse>
```

5. <Service>.describe

<services>.describe	
Response:	
Field	Details
id	This is the XML unique name of the service.
name	This is the service friendly name
version	This is the service version

propertySupport	This flag tells if the service has properties. When properties are supported, the following method are available: • <service>.listProperties • <service>.getProperties • <service>.setProperties
functionSupport	This flag tells if the service has custom functions. When custom functions are supported, the following methods are available: • <service>.listFunctions • <service>.listParameters • <service>.<functionName>

[<Service>.Describe Example]

call

```
<methodCall>
<methodName>cviNetwork.describe</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><struct>
<member><name>id</name><value><string>cviNetwork</string></value></member>
<member><name>name</name><value><string> cviNetwork </string></value></member>
<member><name>version</name><value><string>1.0</string></value></member>
<member><name>propertySupport</name><value><uint>1</uint></value></member>
<member><name>functionsSupport</name><value><uint>0</uint></value></member>
</struct></value>
</param></params>
</methodResponse>
```

6. <Service>.listProperties

This call returns a list of supported properties for this service.

<services>.listProperties	
Response:	
Field	Details
id	This is the XML unique name of the property.
name	Property friendly name
value	Property current value
validation	This is the validation type
<rule type>	This is the validation rule to use

conditional	In case of a property dependency, this field specify the property that it depend on
<properties name>	This is the validation type for the conditional property
<valid rule type>	This is the validation rule for conditional property
access	This is the property access right
units	This is the property unit

[<Service>.listProperties Example]

call

```
<methodCall>
<methodName>cviNetwork.1.listProperties</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><array><data>
<value><struct>
<member><name>id</name><value><string>ip</string></value></member>
<member><name>name</name><value><string>IP address</string></value></member>
<member><name>value</name><value><string>192.168.1.1</string></value></member>
<member><name>validation</name><value><string>regexp</string></value></member>
<member><name>regexp</name><value><string>[0-2][0-9][0-9]*.</string></value></member>
<member><name>access</name><value><string>rw</string></value></member>
<member><name>units</name><value><string>n/a</string></value></member>
</value></struct>
<value><struct>
<member><name>id</name><value><string>port</string></value></member>
<member><name>name</name><value><string>Port</string></value></member>
<member><name>value</name><value><uint>1111</uint></value></member>
<member><name>validation</name><value><string>range</string></value></member>
<member><name>range</name><value><string>0,65000,1</string></value></member>
<member><name>access</name><value><string>rw</string></value></member>
<member><name>units</name><value><string>n/a</string></value></member>
</value></struct>
...
</data></array></value>
</param></params>
</methodResponse>
```

7. <Service>.listFunctions

This call returns a list of supported custom function for this service.

<services>.listFunctions

Response:

Field	Details
methodName	This is the name of the custom function.
multiCallSupport	This flag tells if the custom function call can contain multiple blocs of data
multiAnswerSupport	This flag tell if the custom function answer can contain multiple blocs of data

[<Service>.listFunctions Example]

call

```
<methodCall>
<methodName>cviNvr.listFunctions</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><array><data>
<value><struct>
<member><name>methodName</name><value><string>queryClip</string></value></member>
<member><name>multiCallSupport</name><value><uint>0</uint></value></member>
<member><name>multiAnswerSupport</name><value><uint>1</uint></value></member>
</struct></value>
</data></array></value>
</param></params>
</methodResponse>
```

8. <Services>.listParameters

This call needs a methodName value as input. The device will than return the description of the custom function call bloc and the custom function answer bloc.

<services>.listParameters	
Field	Details
<methodName>	This is the name of the custom function.
Response:	
Field	Details
methodCall	Contains the custom function call bloc definition.
id	This is the XML unique name of the parameter
name	The friendly name of the parameter
valueType	The data type of the value of the parameter
validation	This is the validation type

<rule type>	This is the validation rule
units	This is the parameter unit
paramType	This is the mandatory or optional flag
methodResponse	Contain the custom function answer bloc definition.
id	This is the XML unique name of the answer parameter
name	The friendly name of the answer
valueType	The data type of the value of the answer
units	This is the answer unit

[<Service>.listParameters Example]

call

```
<methodCall>
<methodName>cviNvr.listParameters</methodName>
<params><param>
<value><string>queryClip</string></value>
</param></params>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><struct>
<member><name>methodCall</name><value>
<array><data>
<value><struct>
<member><name>id</name><value><string>mediaType</string></value></member>
<member><name>name</name><value><string>Type of media</string></value></member>
<member><name>valueType</name><value><int></int></value></member>
<member><name>validation</name><value><string>list</string></value></member>
<member><name>list</name><value><string>Video</string></value></member>
<member><name>list</name><value><string>Audio</string></value></member>
<member><name>list</name><value><int>Motion</int></value></member>
<member><name>units</name><value><string>n/a</string></value></member>
<member><name>paramType</name><value><string>mandatory</string></value></member>
</value></struct>
<value><struct>
<member><name>id</name><value><string>major</string></value></member>
<member><name>name</name><value><string>Input of media</string></value></member>
<member><name>valueType</name><value><int></int></value></member>
<member><name>validation</name><value><string>range</string></value></member>
<member><name>range</name><value><string>1,16,1</string></value></member>
<member><name>units</name><value><string>n/a</string></value></member>
<member><name>paramType</name><value><string>mandatory</string></value></member>
</value></struct>
...
<value><struct>
<member><name>id</name><value><string>eventValue</string></value></member>
<member><name>name</name><value><string>Value of event</string></value></member>
<member><name>valueType</name><value><int></int></value></member>
```

```

<member><name>validation</name><value><string>range</string></value></member>
<member><name>range</name><value><string>0,1,1</string></value></member>
<member><name>units</name><value><string>n/a</string></value></member>
<member><name>paramType</name><value><string>conditional</string></value></member>
<member><name>conditional</name><value><string>eventEnable</string></value></member>
<member><name>eventEnable</name><value><string>validRange</string></value></member>
<member><name>validRange</name><value><string>1,1,0</string></value></member>
</value></struct>
</data></array></value></member>

<member><name>MethodResponse</name><value>
<array><data>
<value><struct>
<member><name>id</name><value><string>mediaType</string></value></member>
<member><name>name</name><value><string>Type of media </string></value> </member>
<member><name>valueType</name><value><int></int></value></member>
<member><name>units</name><value><string>n/a</string></value></member>
</value></struct>
...
<value><struct>
<member><name>id</name><value><string>clipTime</string></value></member>
<member><name>name</name><value><string>Clip Timestamp</string></value></member>
<member><name>valueType</name><value><dateTime.iso8601></dateTime.iso8601></value></mem
ber>
<member><name>units</name><value><string>n/a</string></value></member>
</value></struct>
</data></array></value></member>
</param></params>
</methodResponse>

```

3) Configure Device

Network configuration

The client sends a CVIPv2 request to change the value of a network property. This request must contain the property id and the new property value.

cviNetwork.setProperties	
Field	Details
<property id>	The property to modify with the new value
Response: acknowledgement	

Video configuration

The client sends a CVIPv2 request to change the value of a video property. This request must contain the property id and the new property value.

cviVideo.setProperties	
Field	Details
<property id>	The property to modify and the new value
Response: acknowledgement	

1. <Service>.getProperties

[<Service>.getProperties Example]

call

```
<methodCall>
<methodName>cviVideo.1.getProperties</methodName>
<params><param>
<array></data>
<value><string>parameter_1_AsExemple</string></value>
</data></array>
</param></params>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<struct>
<member><name>parameter_1_AsExemple</name><value><int>1</int></value></member>
</struct>
</param></params>
</methodResponse>
```

2. <Service>.setProperties

[<Service>.setProperties Example]

call

```
<methodCall>
<methodName>cviVideo.setProperties</methodName>
<params><param><struct>
<member><name>parameter_1_AsExemple</name><value><uint>18</uint></value></member>
<member><name>parameter_2_AsExemple</name><value><int>64</int></value></member>
<member><name>parameter_3_AsExemple</string><value><string>AB</string></value></member>
</struct></param></params>
</methodCall>
```

response

Acknowledge answer

4) Update Device Firmware

The client sends a CVIPv2 request to start the firmware update. This request must contain information that describes the firmware file to upload to the device. The device returns an acknowledgment and waits for the firmware image to be transferred.

Subsequently, the client can send CVIPv2 requests to be informed of the current process state and status.

1. *cviUpdateFirmware.startFirmwareUpdate*

cviUpdateFirmware.startFirmwareUpdate	
Field	Details
ImageFileId	New firmware file name.
AuthenticationId	Unique Id used to open the data channel
Response: acknowledgement	

[cviUpdateFirmware.startFirmwareUpdate Example]

call

```
<methodCall>
<methodName>cviUpdateFirmware.startFirmwareUpdate</methodName>
<params><param>
<value><array>
<data><value><struct>
<member><name>imageFileId</name><value><string>"image file name"</string></value></member>
</struct></value>
<value><struct>
<member><name>authenticationId</name><value><string>"any Id" </string></value></member>
</struct></value></data>
</array></value>
</param></params>
</methodCall>
```

response

Acknowledge answer

2. *cviUpdateFirmware.getProgressInfo*

cviUpdateFirmware.getProgressInfo	
Response	
Field	Details
state	Current step of the firmware update procedure.
status	Error status.
bytesProgress	Current Number of bytes either having been transferred to the unit during a file transfer or the current number of bytes erased/written so far during a flash update.
totalBytesProgress	Total Number of bytes either being transferred to the unit during a file transfer or the total number of bytes to erase/write during a flash update.

[cviUpdateFirmware.getProgressInfo Example]

call

```
<methodCall>
<methodName>cviUpdateFirmware.getProgressInfo</methodName>
<params/>
</methodCall>
```

response

```
<methodResponse>
<params><param>
<value><array>
<data><value><struct>
<member><name>state</name><value><string>FWUPDUNINITIALIZED</string></value></member>
<member><name>status</name><value><string>FWUPD_STATUS_OK</string></value></member>
<member><name>bytesProgress</name><value><string>0</string></value></member>
<member><name>totalBytesProgress</name><value><string>0</string></value></member>
</struct></value></data>
</array></value>
</param></params>
</methodResponse>
```

4. Device Protocol

1) Manage Serial Port Service

1. Set Serial Port piping configuration

The client sends a CVIPv2 request. The device will then, return an acknowledgement, once the serial port piping configuration is set.

cviSerialPort.setProperties	
Field	Details
ID	The serial port to use as an output
OperationMode	The operation mode of the serial port
BaudRate	The baud rate of the serial port
DataBits	The number of bits in transmitted data
PortParity	The type of parity check
StopBits	The number of stop bits in each transmission
LineDriver	The way the device detects the electrical line interface of the serial port.
Response: acknowledgement	

2. Start Serial Port piping

The client sends a CVIPv2 request. The device will then, return an acknowledgement, once the serial port piping session is established.

cviSerialPort.create	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
ID	The serial port to use as an output
OperationMode	The operation mode of the serial port
BaudRate	The baud rate of the serial port
DataBits	The number of bits in transmitted data
PortParity	The type of parity check
StopBits	The number of stop bits in each transimission
LineDriver	The way the device detects the electrical line interface of the serial port.

Persistent	Restart or not this streaming after reboot
Response: acknowledgement	

3. Stop Serial Port piping

The client sends a CVIPv2 request. The device will then, return an acknowledgement once the serial port piping session is closed.

cviSerialPort.delete	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
ID	The serial port to use as an output
Response: acknowledgement	

4. List Serial Port Piping

The client sends a CVIPv2 request. The device will then return a list of all the running serial port piping session. Each session in the list is returned with all the details.

cviSerialPort.list	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
ID	The serial port to use as an output
OperationMode	The operation mode of the serial port
BaudRate	The baud rate of the serial port
DataBits	The number of bits in transmitted data
PortParity	The type of parity check
StopBits	The number of stop bits in each transimission
LineDriver	The way the device detects the electrical line interface of the serial port.
ConnectionType	The type of connection, TCP/UDP
ConnectionPort	The port of connection
Persistent	Restart or not this streaming after reboot

2) Manage Streaming Service

1. Start streaming

The client sends a CVIPv2 request. The device will then, return an acknowledgement once the streaming session is established.

cviStream.create	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The input to use as a media source
minor	The encoder to use as a media source
persistent	Restart or not this streaming after reboot
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol
Response: acknowledgement	
GUID	Numeric value unique to every device. Used to control the origin of the media
StreamID	Numeric value unique to every streaming
MediaType	The streaming type

2. Stop streaming

The client sends a CVIPv2 request. The device will then, return an acknowledgement once the streaming session is closed.

cviStream.delete	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
StreamID	Numeric value unique to every streaming
Response: acknowledgement	

3. List streaming

The client sends a CVIPv2 request. The device will then return a list of all the running streaming session. Each session in the list is returned with all the details.

cviStream.list	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.

cviStream.list	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
StreamID	Numeric value unique to every streaming
major	The input to use as a media source
minor	The encoder to use as a media source
mediatype	The media source type
persistant	Restart or not this streaming after reboot
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol

3) Manage Display Service

1. Start display

The client sends a CVIPv2 request. The device will then return an acknowledgement once the display session is established.

cviDisplay.create	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The output to use as a media source
minor	The decoder to use as a media source

mediatype	The media source type
persistent	Restart or not this streaming after reboot
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source
srcTransport	Remote media source transport protocol
Response: acknowledgement	

2. Stop display

The client sends a CVIPv2 request. The device will then return an acknowledgement once the display session is closed.

cviDisplay.delete	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The output to use as a media source
minor	The decoder to use as a media source
mediatype	The media source type
persistent	Restart or not this display after reboot
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source
srcTransport	Remote media source transport protocol
Response: acknowledgement	

3. List display

The client sends a CVIPv2 request. The device will then return a list of all the running display session. Each session in the list is returned with all the details.

cviDisplay.list	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The output to use as a media source
minor	The decoder to use as a media source

mediatype	The media source type
persistant	Restart or not this display after reboot
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source
srcTransport	Remote media source transport protocol

4) Manage Analytics Service

1. Start Analytics

The client sends a CVIPv2 request. The device will then return an acknowledgement once the analytics session is established.

cviAnalytics.create	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The input to use as a media source
minor	The encoder to use as a media source
mediatype	The media source type
persistent	Restart or not this streaming after reboot
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol
Response: acknowledgement	

2. Stop Analytics

The client sends a CVIPv2 request. The device will then return an acknowledgement once the analytics session is closed.

cviDisplay.delete	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The input to use as a media source
minor	The encoder to use as a media source
mediatype	The media source type

persistent	Restart or not this streaming after reboot
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol
Response: acknowledgement	

3. List Analytics

The client sends a CVIPv2 request. The device will then return a list of all the running analytics session. Each session in the list is returned with all the details.

cviDisplay.list	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The input to use as a media source
minor	The encoder to use as a media source
mediatype	The media source type
persistent	Restart or not this streaming after reboot
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol

The analytics data will be streamed in a XML format, which is described in details in the following.

[Motion Detection Data]:

the DataSourceGUID is the GUID of the video source

the DataTime is the timestamp of the video source.

The first parameter is a flag to indicate whether the bounding box data is validate, followed by an arrays of bounding box coordinates (left, top, right, bottom).

The other parameters are the array of object presented in block coordinates (x, y).

DataExample

```

<AnalyticsData>
<DataSourceGUID>00000001-0000-0000-0000-001EC901E30B</DataSourceGUID>
<DateTime>0</DateTime>
<params>
<param>
<value><boolean>1</int></boolean>
<value><array><data>
<value><struct>
<member> <name>Left</name><value><int>0</int></value></member>
<member> <name>Top</name><value><int>0</int></value></member>
<member> <name>Right</name><value><int>50</int></value></member>
<member> <name>Bottom</name><value><int>50</int></value></member>
</struct></value>
</data></array></value>
</param>
<param>
<value><array><data>
<value><int>10</int></value>
<value><int>15</int></value>
<value><int>10</int></value>
<value><int>16</int></value>
</data></array></value>
</param>
</params>
</ AnalyticsData >

```

5) Manage PTZ Service

1. Set PTZ Pipe

The client sends a CVIPv2 request. The device will then return an acknowledgement once the PTZ property is set.

cviPtz.setProperties	
Field	Details
PtzID	The id of PTZ command port
ProtocolType	The protocol type
UserPriority	The user priority
Response: acknowledgement	

Protocol Type	Description
0x01 Pelco P	Pelco P protocol
0x02 Pelco D	Pelco D protocol
0x03 Panasonic	Panasonic Protocol

2. Send PTZ command

The client sends a CVIPv2 request. The device will then return an acknowledgement once the PTZ command is executed.

cviPtz.cmd	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
PtzID	The id of PTZ command port
PhyAddr	The physical address of camera, please refer to the camera entity property
UserPriority	The user priority
CmdType	0x00: Stop 0x01: Pan 0x02: Tilt 0x03: Zoom 0x04: Set pre-position 0x05: Go to pre-position 0x06: LeftUp/RightDown 0x07: RightUp/LeftDown
CmdData	A signed value between 1 and 16, the sign presents the direction, left/right, up/down, zoom in/zoom out, LeftUp/RightDown and RightUp/LeftDown. The magnitude presents the tilt speed.
Response: acknowledgement	

[PTZ Command Data]:

the GUID is the client ID

the CamLogicalID could be retrieved from camera entity.

The user priority is reserved for future use.

Example

The following message illustrate a user panning the camera 15 left with speed 9

```
<xml>
<methodCall>
<methodName>cviPTZ.cmd</methodName>
<params>
<param>
<value><string>00000001-0000-0000-0000-001EC901E30B</string></value>
</param>
<param>
<value><int>1</int></value>
</param>
<param>
<value><int>15</int></value>
</param>

<param>
<value><int>0</int></value>
</param>
<param>
<value><int>1</int></value>
</param>
<param>
<value><int>9</int></value>
</param>
</params>
</methodCall>
</ xml >
```

5. NVR Protocol

1) Manage Recording Service

1.1 Start Recording

The client sends a CVIPv2 request. The NVR will then return an acknowledgement once the recording session is established.

cviRecord.create	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The channel ID to use as a media sink
minor	The location ID to use as a media sink
persistent	Restart or not this recording after reboot
mediatype	The media sink type
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source
srcTransport	Remote media source transport protocol
scheduleProfileID	Schedule profile ID
camEntityID	Camera entity id
Response: acknowledgement	

1.2 Stop Recording

The client sends a CVIPv2 request. The NVR will then, return an acknowledgement once the recording session is closed.

cviRecord.delete	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The channel ID to use as a media sink
minor	The location ID to use as a media sink
mediatype	The media sink type
persistant	Restart or not this recording after reboot
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source

srcTransport	Remote media source transport protocol
Response: acknowledgement	

1.3 List Recording

The client sends a CVIPv2 request. The NVR will then, return a list of all the running recording session. Each session in the list is returned with all the details.

cviRecord.list	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The channel ID to use as a media sink
minor	The location ID to use as a media sink
mediatype	The media sink type
persistant	Restart or not this recording after reboot
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source
srcTransport	Remote media source transport protocol
scheduleProfileID	Schedule profile ID

1.4 Update Recording Schedule

The client sends a CVIPv2 request. The NVR will then, update the schedule setting or the current record session.

cviRecord.update	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
major	The channel ID to use as a media sink
minor	The location ID to use as a media sink
mediatype	The media sink type

persistant	Restart or not this recording after reboot
srcIp	Ip address of a remote media source
srcPort	Port number of a remote media source
srcTransport	Remote media source transport protocol
scheduleProfileID	Schedule profile ID

2) Manage Playback Service

2.1 Start playback

The client sends a CVIPv2 request. The device will then, return an acknowledgement once the playback session is established.

cviPlayback.create	
Field	Details
Request:	
GUID	Numeric value unique to every device. Used to control the origin of the media.
Major	The channel ID to use as a media source
Minor	The location ID to use as a media source
StartTime	Start time of the recorded file, the time string format is YYYYMMDDHHMMSS
EndTime	End time of the recorded file, the time string format is YYYYMMDDHHMMSS
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol
Response:	
GUID	Numeric value unique to every device
PlaybackID	Numeric value unique to every playback stream
MediaType	The media sink type

Media Type	Description
0x01 H264	H264 stream
0x02 MP4V	MPEG4 stream
0x03 VSM4	Verint SM4 stream

2.2 Stop playback

The client sends a CVIPv2 request. The device will then, return an acknowledgement once the playback session is closed.

cviPlayback.delete	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
PlaybackID	Numeric value unique to every playback stream
Response: acknowledgement	

2.3 List playback

The client sends a CVIPv2 request. The device will then return a list of all the running playback session. Each session in the list is returned with all the details.

cviPlayback.list	
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
PlaybackID	Numeric value unique to every playback session.
major	The channel ID to use as a media source
minor	The location ID to use as a media source
MediaType	The media sink type
mode	The media play mode
startTime	Start time of the recorded file, the time string

	format is YYYYMMDDHHMMSS
endTime	End time of the recorded file, the time string format is YYYYMMDDHHMMSS
destIp	The ip address of the stream destination
destPort	The port of the stream destination
destTransport	Stream destination transport protocol

2.4 Set playback state and speed

The client sends a CVIPv2 request to change the playback session state (pause, stop, play, fast forward and back forward). The device will then return an acknowledgement once the playback session is configured.

cviPlayback.setstate	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
PlaybackID	Numeric value unique to every playback session.
State	The playback state, which is illustrated in the following table
Speed	The playback speed, 1x, 2x...

Play State	Description
0x00 PLAYBACK_STATE_PAUSE	Pause state
0x01 PLAYBACK_STATE_PLAY	Play state
0x02 PLAYBACK_STATE_STOP	Stop state
0x03 PLAYBACK_STATE_FASTFORWARD	Fast forward state
0x04 PLAYBACK_STATE_FASTBACKWARD	Fast backward state
0x08 PLAYBACK_STATE_STEPFORWARD	Step forward
0x09 PLAYBACK_STATE_STEPBACKWARD	Step backward

2.5 Seek playback position

The client sends a CVIPv2 request to seek the playback session to a specified position. The device will then return an acknowledgement once the playback session is seek successfully.

cviPlayback.seek	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
PlaybackID	Numeric value unique to every playback session.
Position	The position of the playback session, which unit is frame based.

2.6 Get playback length

The client sends a CVIPv2 request to get the length of the playback session. The device will then return the length of playback session.

cviPlayback.getlength	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
PlaybackID	Numeric value unique to every playback session.
Response:	
Field	Detail
Length	Length of the playback session, which unit is frame based.

2.7 Get playback status

The client sends a CVIPv2 request to get the status of the playback session. The device will then return the current status of playback session, play state, play speed and position.

cviPlayback.getstatus	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
PlaybackID	Numeric value unique to every playback session.
Response:	
Field	Detail
State	Playback state
Speed	Playback speed
Position	Current position, which unit is frame based

2.8 Query playback list with event type

The client sends a CVIPv2 request to query the list of the playback session, which has specified event type. The device will then return the list of playback start time, end time and event type.

cviPlayback.getstatus	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
MajorID	The major id of recorder channel
MinorID	The minor id of recorder channel
StartTime	The start time of the event
EndTime	The end time of the event
EventType	The bit or result of user specified event type. For example, if the user like to query Motion detection and video loss event at the same time. EventType is MOTIOM VIDEO_LOSS
Response:	
Field	Detail
StartTime	The start time of event
EndTime	The end time of the event
EventType	The event type

3) Manage Media Service

3.1 Media Browsing

The client sends a CVIPv2 request. . The NVR will then return information about all the recorded media sources stored on the NVR. The NVR is searching its media storage from a range of timestamp. The response can contain more than one item, depending on the numbers of tuples found.

cviMedia.query	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media
Major	The input to use as a media source
Minor	The encoder to use as a media source
startDate	Start date filter for search
endDate	end date filter for search
eventType	In case of event type browsing, the event type
eventSubType	In case of event type browsing, the event sub type
eventValue	In case of event type browsing, the event value
Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
IsLastPage	Flag to indicate whether current page is the last
ClipID	Numeric value unique to every clip.
ClipType	Clip media type
ClipTime	Clip date and time

3.2 Export Recorded Media

The client sends the CVIPv2 request containing the information needed to retrieve the clip and the transfer server identification string. The NVR returns an acknowledgment and start transferring data to the server specified by the client in the request. The download is performing using a separate data channel using binary transfer mode.

cviMedia.export	
Field	Details
GUID	Numeric value unique to every device. Used to control the origin of the media.
ClipID	Numeric value unique to every clip
serverIP	IP of the data transfer server
serverPort	Port of the data transfer server
authenticationId	Unique Id used to open the data channel
Response: acknowledgement	

4) Manage Camera Event Service

4.1 Subscribe event

Client sends a CVIPv2 request to subscribe to a camera event.

cviEvent.subscribe	
Field	Detail
Type	Bits or combination of the event type, please refer to event type table
Id	The channel index of encoder or recorder, 0 means subscribe all the channels
Response: acknowledgement	

Event type table

Event Type	Description
0x01 Keep Alive	Indicate the device is alive
0x02 External	External alarm from I/O pins
0x04 Motion	Video content motion detection alarm
0x08 Video Input	Video signal loss alarm
0x10 Record Loss	Recorder loss packet alarm
0x20 RecordDisc	Recorder disc failure alarm

4.2 Unsubscribe event

Client sends a CVIPv2 request to unsubscribe from a camera event.

cviEvent.unsubscribe	
Type	Bits or combination of the event type, please refer to event type table
Id	The channel index of encoder or recorder, 0 means unsubscribe all the channels
Response: acknowledgement	

4.3 Receive event Notification

On a device with event service, the unit will send a notification every time an event occurs. All the previously registered client will then receive a notification with all the event details.

cviEvent.notify	
Field	Detail
Type	The event type, please refer to event type table
Id	The channel index of encoder or recorder
Time	Time stamp of alarm. The format is as "YYYY-MM-DD HH:MM:SS"
EventGUID	The GUID to identify event (since CMS version 2.1.9)
info	Event specific data, 1: alarm on, 0: alarm off.

4.4 Acknowledge event

Any client could acknowledge the notified event to Master Server.

cviEvent.ack	
Field	Detail
AppGUID	The GUID to identify application
EventGUID	The GUID to identify event
AckTime	Time stamp of acknowledgement. The format is as "YYYY-MM-DD HH:MM:SS"
AckString	The string to describe acknowledgement specified by user
Response:	
AckReturnCode	The acknowledge return code from the server, 0x01000000: success, 0x01000001: fail, 0x01000002:server busy, 0x01000003: invalid packet.

6. Master Server Protocol

1) Manage Device Profile Service

1.1 List Profiles

The client sends a CVIPv2 request. The Master Server will then, return a list of all the profile details.

cviProfile.list Request:	
Field	Details
GUID	Numeric value unique to every device.
ProfileTypeID	The profile type.

cviProfile.list Response:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
ProfileTypeID	The profile type
ProfileID	The ID to identify profile
<ProfileProperty>	The property of the profile, which depends on certain profile

1.2 Add Profile

cviProfile.add Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
ProfileTypeID	The profile type
<ProfileProperty>	The property of the profile, which depends on certain profile

1.3 Update Profile

cviProfile.update	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
ProfileTypeID	The profile type
ProfileID	The profile ID
<ProfileProperty>	The property of the profile, which depends on certain profile

1.4 Delete Profile

cviProfile.delete	
Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
ProfileType	The profile type
ProfileID	The profile ID

Profile Type ID	Profile Property
0x01 SerialPortProfile	portType
	baudRate
	parity
	stopBit
	dataBit
	portMode
	dateValid
0x02 ScheduleProfile	startDateTime
	endDateTime
	iteration
	specificDateTime
	timeBlock
0x03 VideoEncoderProfile	targetBitRate
	maxBitRate
	rateControl
	compression
	intraInterval

0x04 AudioEncoderProfile	QMax
	QMin
	frameRate
	skipRate
	resolution
	channel
	samplingFreq
	samplingBits
	format

2) Manage Device Entity Service

2.1 List <Entity> Groups

The client sends a CVIPv2 request. The Master Server will then return a list of all the entity groups. Each group in the list is returned with all the details.

cviEntity.listgroups Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.
EntityTypeID	The entity type

cviEntity.listgroups Response:	
Field	Details
EntityTypeID	The entity type
GroupID	Group ID.
GroupName	Group Name
GroupDesc	Group Description

2.2 Add <Entity> group

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity group is added.

cviEntity.addgroup	
Field	Detail
GUID	Numeric value unique to every device.

AppGUID	The application GUID
EntityTypeID	The entity type
GroupName	The new group name
GroupDescription	Group description
Response: acknowledgement	

2.4 Update <Entity> group

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity group is updated.

cviEntity.updategroup	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
EntityTypeID	The entity type
GroupID	The group ID
GroupName	The new group name
GroupDescription	Group description
Response: acknowledgement	

2.5 Delete <Entity> group

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity group is deleted.

cviEntity.deletegroup	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
EntityTypeID	The entity type
GroupID	The group ID to be deleted
Response: acknowledgement	

2.6 Query <Entity>

The client sends a CVIPv2 request. The Master Server will then return a version number of current entity description and the last modified time. The IsLastPage field will indicate whether current page is the last one. Each entity in the list is returned with all the details.

cviEntity.query Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.
EntityTypeID	The entity type

cviEntity.query Response:	
Field	Details
EntityTypeID	Entity type
Version	The version current entity description
TimeStamp	The timestamp of the last modification

2.7 List <Entity>

The client sends a CVIPv2 request. The Master Server will then return a list of all the entity in separated page. The IsLastPage field will indicate whether current page is the last one. Each entity in the list is returned with all the details.

cviEntity.list Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.
EntityTypeID	The entity type

cviEntity.list Response:	
Field	Details
IsLastPage	Flag to indicate whether current page is the last page
EntityTypeID	Entity type

GroupID	The entity group ID
EntityID	Entity ID
EntityName	Entity name
EntityDesc	Entity description
<EntityProperty>	The property of entity, which depends on certain entity.

2.8 Add <Entity>

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity is added.

cviEntity.add	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
IsLastPage	Flag to indicate whether current page is the last page
EntityTypeID	The entity type
GroupID	The entity group ID
<EntityProperty>	The entity property
Response: acknowledgement	

2.9 Update <Entity>

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity is updated.

cviEntity.update	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
IsLastPage	Flag to indicate whether current page is the last page
EntityTypeID	The entity type
EntityID	Entity ID
<EntityProperty>	The entity property
Response: acknowledgement	

2.10 Delete <Entity>

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity is deleted.

cviEntity.delete	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
EntityTypeID	The entity type ID
EntityID	The entity ID
Response: acknowledgement	

Entity Type ID	Entity Property
0x01 Camera	Capability
	PhyAddr
	LogicalID
0x02 Monitor	
0x03 Device	Enable
	IPAddress
	Port
	Input/Output Number
	DeviceGuid
	DeviceType
	DeviceUpdateTime
0x04 Sequence	LogicID
	CameraID
	Duration
	PrePosition
0x05 Map	MapEntityTypeID
	MapEntityID
	MapEntityPosX
	MapEntityPosY
	MapEntityRotation
0x06 Image	ImageTypeID
	ImageSize
	ImageData
0x33 AlarmMessage	ObjectID (the event associated device camera or external alarm device)

	AlarmID (the alarm id of the system)
	AlarmType (the alarm type)
	BGColor (the background color)
	FGColor (the foreground color)
	Message (user customized message)
	Font (message font)
	FontSize (message font size)
	PosX (the x position of the message)
	PosY (the y position of the message)

The capability of camera is the bitwise or of SURPPOINT_PTZ (0x01) and SUPPORT_AUX (0x02)

The ImageData in a single packet will not greater than 1024Kbytes. Thus, in the case that the image is greater than 750 Kbytes, it will be split into several packets with the same entity id.

ImageTypeID	Image Type
0x01	8 bits bitmap
0x02	24 bits bitmap
0x03	32 bits bitmap
0x04	Png format
0x05	JPG format
0x06	Gif format

For the map entity, the MapEntityPosX and MapEntityPosY are the (x,y) coordinates on the first image type map entity. MapEntityTypeID is defined with following type element:

Map Entity TypeID	Entity
0x01 Map	Sub map of the current map
0x02 Image	Map image of the current map
0x03 Camera	Camera device
0x04 Monitor	Monitor device
0x05 Sensor	Sensor device

0x06 Intercom	Intercom device
0x07 Microphone	Microphone device
0x08 Speaker	Speaker device
0x09 Relayout	Relayout device
0x0A Relaybox	Relay box device
0x0B Encoder	Encoder device
0x0C Recorder	Recorder device

In addition, the device entity supports the following two messages.

2.11 List <entity> service

The client sends a CVIPv2 request. The Master Server will then return a list of all the entity service. Each entity service in the list is returned with all the details.

cviEntity.listservices Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.
EntityID	The entity ID

cviEntity.listservices Response:	
Field	Details
EntityID	Entity ID.
SericeID	Service ID
<ServiceProfile>	The profile of the service

2.12 Update <entity> service

The client sends a CVIPv2 request. The Master Server will then update the entity service.

cviEntity.updateservices Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.
EntityID	The entity ID
ServiceID	The service ID
<ServiceProfile>	The profile ID of the service

Service ID	Service Profile
0x01.[1..n] SerialPort	SerialPortProfile
0x02.[1..n] Recoder	RecoderProfile
0x03.[1..n] VideoEncoder	VideoEncoderProfile
0x04.[1..n] AudioEncoder	AudioEncoderProfile

3) Manage User Service

3.1 Login

The client sends a CVIPv2 request. The Master Server will then authorize a use to login and returns an application GUID.

cviUser.login	
Field	Details
GUID	Numeric value unique to every device
UserName PassWord	User Name Password of the username
LoginType	Login type whether it is via control center or review station

Note: Both username and password need encrypted in MD5.

3.2 Logout

The client sends a CVIPv2 request. The Master Server will then authorize a use to logout and returns an acknowledge command.

cviUser.logout	
Field	Details
GUID AppGUID	Numeric value unique to every device The application GUID

3.3 List Groups

The client sends a CVIPv2 request. The Master Server will then return a list of all the user groups. Each group in the list is returned with all the details.

cviUser.listgroups Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.

cviUser.listgroups Response:	
Field	Details
GroupID	Group ID.
GroupName	Group Name
GroupDescription	Group Description

3.4 Add group

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the user group is added.

cviUser.addgroup	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
GroupName	The new group name
GroupDescription	Group description
GroupPrivilege	Group privilege
Response: acknowledgement	

3.5 Update group

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the user group is updated.

cviUser.updategroup	
Field	Detail
GUID	Numeric value unique to every device.

AppGUID	The application GUID
GroupID GroupName	The group ID The new group name
GroupDescription	Group description
GroupPrivilege	Group privilege
Response: acknowledgement	

3.6 Delete group

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the user group is deleted.

cviUser.deletegroup	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
GroupID	The group ID to be deleted
Response: acknowledgement	

3.7 List Users

The client sends a CVIPv2 request. The Master Server will then return a list of all the users. Each user in the list is returned with all the details.

cviUser.listusers Request:	
Field	Details
GUID	Numeric value unique to every device.
AppGUID	The application GUID.

cviUser.listusers Response:	
Field	Details
UserID	Group ID.
UserName	User Name
UserDescription	User Description
UserPriority	User priority
UserPrivilege	User privilege
GroupID	The group ID

3.8 Add user

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the user is added.

cviUser.adduser	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
UserName	The new user name
UserDescription	user description
UserPrivilege	User privilege
UserPriority	User priority
GroupID	Group ID
Response: acknowledgement	

3.9 Update user

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the user is updated.

cviUser.updateuser	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
UserID	User ID
UserName	user name
UserPassword	User password
UserDescription	user description
UserPrivilege	User privilege
UserPriority	User priority
GroupID	Group ID
Response: acknowledgement	

3.10 Delete user

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the user is deleted.

cviUser.deleteuser	
Field	Detail
GUID	Numeric value unique to every device.
AppGUID	The application GUID
UserID	The user ID to be deleted
Response: acknowledgement	

User Privilege	User Privilege Description
0x0001 Streaming	Allow user review streaming from camera
0x0002 Playback	Allow user review files recorded
0x0004 Snapshot	Allow user do a snapshot when event occurs
0x0008 Record	Allow user do a instant recording when event occurs
0x0010 Housekeeping	Allow user do house keeping
0x0020 Template	Allow user keep the template of review windows
0x0040 SystemExit	Allow user quit the system

In addition, Camera and Monitor entity supports the following two user management messages.

3.11 Mount Entity

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity is mounted on the user.

cviUser.mount	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
UserID	User ID
EntityID	Entity ID
EntityType	Entity type
Response: acknowledgement	

3.12 Unmount Entity

The client sends a CVIPv2 request. The Master Server will then, return an acknowledgement once the entity is unmounted from the user.

cviUser.unmount	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
UserID	User ID
EntityID	Entity ID
EntityType	Entity Type
Response: acknowledgement	

4) Manage Application Service

4.1 List Application

The client sends a CVIPv2 request. The Master Server will then, return a list of all the application related with the entity. The entity could be Camera or Monitor. Each application in the list is returned with all the details.

cviApplication.list Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
EntityTypeID	Entity type
EntityID	Entity ID

cviApplication.list Response:	
Field	Detail
EntityTypeID	Entity type
EntityID	Entity ID.
ApplicationTypeID	The application type ID
<ApplicationProperty>	The application property

4.2 Add Application

The client sends a CVIPv2 request. The Master Server will add an application and associate it with the entity.

cviApplication.add Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
EntityTypeID	Entity Type
EntityID	Entity ID
ApplicationTypeID	The application type ID
<ApplicationProperty>	The application property

4.2 Update Application

The client sends a CVIPv2 request. The Master Server will update the application.

cviApplication.update Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
EntityTypeID	Entity type
EntityID	Entity ID
ApplicationTypeID	The application type ID
<ApplicationProperty>	The application property

4.3 Delete Application

The client sends a CVIPv2 request. The Master Server will delete the application.

cviApplication.delete Request:	
Field	Detail
GUID	Numeric value unique to every device. Used to control the origin of the media.
AppGUID	The application GUID
EntityTypeID	Entity type
EntityID	Entity ID

Application Type ID	Application Property
0x01 PTZ Control	SerialPortEntityIp
	SerialPortEntityPort
	SerialPortServiceID
0x02 Video Streaming	VideoSourceEntityIP
	VideoSourceEntityPort
	VideoEncoderServiceID
0x03 Video Playback	VideoSinkEntityIP
	VideoSinkEntityPort
	VideoRecorderServiceID
0x04 Video Recording	VideoSourceEntityIP
	VideoSourceEntityPort
	VideoEncoderServiceID
	VideoSinkEntityIP
	VideoSinkEntityPort
	VideoRecorderServiceID

Service and Properties

1. Service

Service	Use cases	Property support	Function Support
cvip	CVIP protocol property	X	-
cviUnit	Browse NVR informations	X	-
cviNVR	Browse recorded media	-	X
cviClipExporter	Download recorded media	-	X
cviRecord	Manage Recording service	-	X
cviDisplay	Manage Display service	-	X
cviStream	Manage Stream service	-	X
cviUpdateFirmware	Update NVR Firmware	-	X
cviNetwork	Configure NVR, Network configuration	X	-
cviNetwork.1	Configure NVR, Network configuration port 1	X	-
cviNetwork.2	Configure NVR, Network configuration port 2	X	-
cviVideo.01	Configure NVR, Video configuration input 01	X	-
cviVideo.01.1	Configure NVR, Video configuration input 01 encoder 1	X	-
cviVideo.01.2	Configure NVR, Video configuration input 01 encoder 2	X	-
cviVideo.02	Configure NVR, Video configuration input 02	X	-
cviVideo.02.1	Configure NVR, Video configuration input 02 encoder 1	X	-
cviVideo.02.2	Configure NVR, Video configuration input 02 encoder 2	X	-
cviVideo.03	Configure NVR, Video configuration input 03	X	-
cviVideo.03.1	Configure NVR, Video configuration input 03 encoder 1	X	-
cviVideo.03.2	Configure NVR, Video configuration input 03 encoder 2	X	-
cviVideo.04	Configure NVR, Video configuration input 04	X	-
cviVideo.04.1	Configure NVR, Video configuration input 04 encoder 1	X	-
cviVideo.04.2	Configure NVR, Video configuration input 04 encoder 2	X	-
cviVideo.05	Configure NVR, Video configuration input 05	X	-
cviVideo.05.1	Configure NVR, Video configuration input 05 encoder 1	X	-
cviVideo.05.2	Configure NVR, Video configuration input 05 encoder 2	X	-
cviVideo.06	Configure NVR, Video configuration input 06	X	-
cviVideo.06.1	Configure NVR, Video configuration input 06 encoder 1	X	-
cviVideo.06.2	Configure NVR, Video configuration input 06 encoder 2	X	-
cviVideo.07	Configure NVR, Video configuration input 07	X	-
cviVideo.07.1	Configure NVR, Video configuration input 07 encoder 1	X	-
cviVideo.07.2	Configure NVR, Video configuration input 07 encoder 2	X	-
cviVideo.08	Configure NVR, Video configuration input 08	X	-
cviVideo.08.1	Configure NVR, Video configuration input 08 encoder 1	X	-
cviVideo.08.2	Configure NVR, Video configuration input 08 encoder 2	X	-
cviUser	Manage User service		X

2. Properties

Service	Property Id	Access
cvip	cvipPort	Read / Write
cviUnit	unitProductName	Read only
	unitCompanyName	Read only
	unitMarketingName	Read only
	unitFirmwareVersion	Read only
	unitOsVersion	Read only
	unitCpldVersion	Read only
	unitBoardVersion	Read only
	unitUpTime	Read only
	unitTemperature	Read only
	unitFreeMemory	Read only
	unitCvipName	Read / Write
	unitSaveConfig	Read / Write
	unitSaveConfigMask	Read / Write
	unitLoadDefaultConfig	Read / Write
	unitLoadDefaultConfigMask	Read / Write
	unitVideoStandard	Read / Write
cviNetwork	unitNetworkName	Read / Write
	networkNtpServer	Read / Write
	unitDomainName	Read / Write
	networkDnsServer1	Read / Write
	networkDnsServer2	Read / Write
	networkGateway	Read / Write
cviNetwork.1	networkIpAddress	Read / Write
	networkSubnetMask	Read / Write
	networkOnBoot	Read / Write
	networkUseDhcp	Read / Write
	networkPeerNtp	Read / Write
	networkPeerDns	Read / Write
cviNetwork.2	networkIpAddress	Read / Write
	networkSubnetMask	Read / Write
	networkOnBoot	Read / Write
	networkUseDhcp	Read / Write
	networkPeerNtp	Read / Write
	networkPeerDns	Read / Write
cviVideo.n	videoInputState	Read / Write
	videoObstructionDetEnable	Read / Write
	videoObstructionDetSensitivity	Read / Write
	videoInputName	Read / Write
	videoInputBrightness	Read / Write
	videoInputContrast	Read / Write
	videoInputSaturation	Read / Write
	videoInputHue	Read / Write
cviVideo.n.m	videoEncoderState	Read / Write
	videoEncoderResolution	Read / Write
	videoEncoderSkipRate	Read / Write
	videoEncoderFrameRate	Read / Write
	videoEncoderQuantizerMin	Read / Write
	videoEncoderQuantizerMax	Read / Write
	videoEncoderIntraInterval	Read / Write
	videoEncoderCompression	Read / Write
	videoEncoderRateControl	Read / Write
	videoEncoderMaxBitRate	Read / Write
	videoEncoderTargetBitRate	Read / Write
cviMotionDetection.n	motiondetectionEnable	Read / Write
	motiondetectionSensitivity	Read / Write
	motiondetectionSizeBlock	Read / Write
	motiondetectionSizeMin	Read / Write
	motiondetectionRatioCar	Read / Write
	motiondetectionRatioHuman	Read / Write

CVIP V 2 P R O T O C O L

	motiondetectionStringRMA	Read / Write
	motiondetectionFormatContour	Read / Write