



# tvONE WebSocket API

Document version: 1.2.3  
Firmware version: M502 and above  
Supported products: CM2-547

# Contents

Connection .....	4
Event registration .....	4
Subscribe .....	4
Unsubscribe .....	5
Errors .....	6
Invalid JSON format .....	6
Unknown event category .....	6
Media Storage Events .....	7
USB_HOTPLUG_ARRIVED .....	7
USB_HOTPLUG_REMOVED .....	7
OPERATION_DONE .....	8
OPERATION_STARTED .....	8
Media Player Events .....	10
STATUS_UPDATE .....	10
ITEM_STATUS_CHANGED .....	11
Module Events .....	12
USB_POWER_ALERT .....	12
STATUS .....	12
UPDATE_TRANSFER_STARTED .....	13
UPDATE_TRANSFER_PROGRESS .....	13
UPDATE_TRANSFER_FINISHED .....	14
NETWORK_LINK_SPEED_CHANGED .....	14
NETWORK_SETTINGS_CHANGED .....	15
CORE_TEMPERATURE_ALERT .....	15
CHANGED .....	16
Front Panel Events .....	17
LOCKED .....	17
UNLOCKED .....	17
INFO_MODE_ON .....	17
INFO_MODE_OFF .....	18
Output Events .....	19
AUDIO_FOLLOW_WINDOW_CHANGED .....	19
STATUS_GROUP .....	20
PROPERTY_CHANGED .....	21
Input Events .....	22
STATUS_GROUP .....	22
Window Events .....	24
INPUT .....	24
System Events .....	25
UPDATE_STATUS .....	25
POWERMODE_CHANGED .....	25

HDMI Events.....	26
SINK_ATTACHED .....	26
SINK_UNPLUGGED .....	26
Preset Events .....	27
TAKE.....	27
COMPLETE .....	27
SAVE.....	28
REMOVE .....	28
Security Events.....	29
USERCERT_UPDATE_DECRYPTED .....	29
USERCERT_UPDATE_PARSE .....	29
USERCERT_UPDATE_VALIDATION.....	30
USERCERT_UPDATE_COMPLETE .....	30
Canvas Events .....	31
STBDCURRENT_CHANGED .....	31
PROPERTY_CHANGED .....	32
Storyboard Events .....	33
ISCURRENT_CHANGED.....	33
Region Events.....	34
EXCHANGING .....	34
EXCHANGED .....	34

## Connection

Clients should make a connection to:

```
[scheme]://[host]/ws/v1/
```

- scheme
  - ws – unsecure connection
  - wss – secure connection
- host
  - The IP Address of the device

For example, to connection using secure channel

```
wss://192.168.0.10/ws/v1/
```

## Event registration

To register or unregister for events, the client will need to specify which events they are interested in.

### Subscribe

To subscribe to event categories an array of event categories is sent to the device using the subscribe object

```
[  
  {  
    "subscribe" : {  
      "category" : String  
    }  
  },  
  {  
    "subscribe" : {  
      "category" : String  
    }  
  }  
]
```

- subscribe.category (String)
  - The name of the event category to subscribe to

## Unsubscribe

To unsubscribe from event categories an array of event categories is sent to the device using the unsubscribe object

```
[
  {
    "unsubscribe" : {
      "category" : String
    }
  },
  {
    "unsubscribe" : {
      "category" : String
    }
  }
]
```

- unsubscribe.category (String)
  - The name of the event category to unsubscribe from

## Errors

The following errors are handled by the WebSocket API

### Invalid JSON format

If you send a JSON request to the device that is not correctly formatted, then you will get an error.

The response is as follows:

```
{
  "Error" : "JSON parse failed"
}
```

### Unknown event category

If you try to subscribe for an event category that does not exist, then you will get an error.

The response is as follows:

```
{
  "Error" : "Event registration failed - <category>"
}
```

<category> = The category you attempted to subscribe to.

# Media Storage Events

Events associated with media storage on the AVIP module

## USB\_HOTPLUG\_ARRIVED

Event raised when a USB device is plugged into the AVIP module

```
{
  "category" : "MEDIA_STORAGE",
  "event" : "USB_HOTPLUG_ARRIVED",
  "data" : {
    "slot" : String,
    "label" : String,
    "totalsize" : Int,
    "freespace" : Int
  }
}
```

- data.slot (String)
  - The name of the Slot. For example "Slot5"
- data.lable (String)
  - The name of the USB device
- data.totalsize (Int)
  - Total size of the USB device in bytes
- data.freespace (Int)
  - Amount of free space on the USB device in bytes

## USB\_HOTPLUG\_REMOVED

Event raised when a USB device is removed from an AVIP module

```
{
  "category" : "MEDIA_STORAGE",
  "event" : "USB_HOTPLUG_REMOVED",
  "data" : {
    "slot" : String,
  }
}
```

- data.slot (String)
  - The name of the Slot. For example "Slot5"

## OPERATION\_DONE

Event raised when any file operation completes

```
{
  "category" : "MEDIA_STORAGE",
  "event" : "OPERATION_DONE",
  "data" : {
    "slot" : String,
    "exitCode" : Int
  }
}
```

- data.slot (String)
  - The name of the Slot. For example "Slot5"
- data.exitCode (Int)
  - Indicates the result of an asynchronous file operation. Value of 0 for success, all other values indicate failure

## OPERATION\_STARTED

Event raised when any file operation starts

```
{
  "category" : "MEDIA_STORAGE",
  "event" : "OPERATION_STARTED",
  "data" : {
    "slot" : String
  }
}
```

- data.slot (String)
  - The name of the Slot. For example "Slot5"



## SYNC\_STATUS

Event raised when any file operation completes

```
{
  "category" : "MEDIA_STORAGE",
  "event" : "SYNC_STATUS",
  "data" : {
    "slot" : String,
    "command" : String,
    "status" : ENUM (Started, Completed),
    "resultCode" : int
  }
}
```

- data.slot (String)
  - The name of the Slot. For example "Slot5"
- data.command (String)
  - Indicates the sync command that was executed. For example "StartFtpSync"
- data.status (Enum)
  - Return the Sync status of the AVIP module. For example "Started"
- data.resultCode (Int)
  - Indicates the result of an asynchronous file operation. Value of 0 for success, all other values indicate failure

## Media Player Events

Events associated with media playback on an AVIP module

### STATUS\_UPDATE

Event raise when the state of the AVIP media player changes

```
{
  "category" : "MEDIA_PLAYER",
  "event" : "STATUS_UPDATE",
  "data" : {
    "input" : String,
    "state" : ENUM (Idle, Configured, Connecting, Playing, Paused,
                   Disconnecting, Retrying),
    "index" : Int
  }
}
```

- data.input (String)
  - The full name of the input, which includes the slot. For example, "Slot1.In1"
- data.state (Enum)
  - Indicates the state the media palyer has entered
- data.index (Int)
  - The index of the active item in the play queue

## ITEM\_STATUS\_CHANGED

Event raise when the status of an item in the active play queue on an AVIP module changes

```
{
  "category" : "MEDIA_PLAYER",
  "event" : "ITEM_STATUS_CHANGED",
  "data" : {
    "input" : String,
    "itemNumber" : Int,
    "status" : ENUM (OK,Failed),
    "resultCode" : Int
  }
}
```

- data.input (String)
  - The full name of the input, which includes the slot. For example, "Slot1.In1"
- data.itemNumber (Int)
  - The index of the item within the play queue. From 1 to 20.
- data.status (Enum)
  - Status of the item.
- data.resultCode (Int)
  - Currently unused

# Module Events

Events associated with Modules

## USB\_POWER\_ALERT

Event raised to indicate whether the USB device's power requirements are met. Only applies to AVIP module

```
{
  "category" : "MODULE",
  "event" : "USB_POWER_ALERT",
  "data" : {
    "module" : String,
    "status" : ENUM (OK, OverCurrent)
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.status (Enum)
  - Return the power status of the device.

## STATUS

Event raised to indicate the status of an AVIP only

```
{
  "category" : "MODULE",
  "event" : "STATUS",
  "data" : {
    "module" : String,
    "status" : ENUM (READY, SHUTDOWN, BOOTING, UPDATING, BOOTFAILED,
                    UPDATEFAILED, WAITFORVERSION, CARDFAILED, SYNCING)
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.status (Enum)
  - Return the status of an AVIP module

## UPDATE\_TRANSFER\_STARTED

Event raised to indicate AVIP module has started to transfer a new update

```
{
  "category" : "MODULE",
  "event" : "UPDATE_TRANSFER_STARTED",
  "data" : {
    "module" : String
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"

## UPDATE\_TRANSFER\_PROGRESS

Event raised to indicate AVIP module update progress

```
{
  "category" : "MODULE",
  "event" : "UPDATE_TRANSFER_PROGRESS",
  "data" : {
    "module" : String,
    "percentageComplete" : Int,
    "transferredBytes" : Int
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.percentageComplete (Int)
  - Update progress as a percentage.
- data.transferredBytes (Int)
  - The total number of bytes transferred to the module

## UPDATE\_TRANSFER\_FINISHED

Event raised to indicate the update transfer to the AVIP module has finished

```
{
  "category" : "MODULE",
  "event" : "UPDATE_TRANSFER_FINISHED",
  "data" : {
    "module" : String,
    "updateTransferResult" : ENUM (NotSet,UpdateComplete,
                                  UpdateFailedOnModule,
                                  FileNotFound,BPCommsError)
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.updateTransferResult (Enum)
  - Return the result of an AVIP update transfer. Any value other than UpdateComplete indicates a failure.

## NETWORK\_LINK\_SPEED\_CHANGED

Event raised to indicate the network speed on an AVIP module ethernet adaptor has changed

```
{
  "category" : "MODULE",
  "event" : "NETWORK_LINK_SPEED_CHANGED",
  "data" : {
    "module" : String,
    "linkSpeed" : ENUM (0,100,1000)
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.linkSpeed (Enum)
  - Network link speed of the AVIP module's ethernet adaptor.

## NETWORK\_SETTINGS\_CHANGED

Event raised to indicate the network settings on an AVIP module have changed

```
{
  "category" : "MODULE",
  "event" : "NETWORK_SETTINGS_CHANGED",
  "data" : {
    "module" : String
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"

## CORE\_TEMPERATURE\_ALERT

Event raised to indicate the core temperature of an AVIP module has changed status

```
{
  "category" : "MODULE",
  "event" : "CORE_TEMPERATURE_ALERT",
  "data" : {
    "module" : String,
    "status" : ENUM (OK,RunningHot,OverTemperature),
    "coreTemperature" : Int
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.status (Enum)
  - The status of the core temperature on a AVIP Module. Value of OK indicates the module is within safe operating temperature, RunningHot indicates the temperature is higher than normal but should not affect performance. However, OverTemperature indicates the temperature is too high and performance will be affected.
- data.coreTemperature (Int)
  - The current temperature (in degrees Celsius) of the module. For example 75.

## CHANGED

Event raised to indicate the core temperature of an AVIP module has changed status

```
{
  "category" : "MODULE_CORE_TEMPERATURE",
  "event" : "CHANGED",
  "data" : {
    "module" : String,
    "coreTemperature" : Int
  }
}
```

- data.module (String)
  - The full name of the AVIP module. For example, "Slot1"
- data.coreTemperature (Int)
  - The current temperature (in degrees Celsius) of the module. For example 75.



## Front Panel Events

Events associated with the front panel (on certain products)

### LOCKED

Event raised when the front panel is locked

```
{
  "category" : "FRONTPANEL",
  "event" : "LOCKED",
  "data" : {
  }
}
```

Note: data object is empty as there are no parameters for this event

### UNLOCKED

Event raised when the front panel is unlocked

```
{
  "category" : "FRONTPANEL",
  "event" : "UNLOCKED",
  "data" : {
  }
}
```

Note: data object is empty as there are no parameters for this event

### INFO\_MODE\_ON

#### CORIOview only

Event raised when info labels are enabled

```
{
  "category" : "FRONTPANEL",
  "event" : "INFO_MODE_ON",
  "data" : {
  }
}
```

Note: data object is empty as there are no parameters for this event

## INFO\_MODE\_OFF

### CORIOview only

Event raised when info labels are disabled

```
{
  "category" : "FRONTPANEL",
  "event" : "INFO_MODE_OFF",
  "data" : {
  }
}
```

Note: data object is empty as there are no parameters for this event

## Output Events

Events associated with output modules

### AUDIO\_FOLLOW\_WINDOW\_CHANGED

Event raised when the audio source routing for an output is changed

```
{
  "category" : "OUTPUT",
  "event" : "AUDIO_FOLLOW_WINDOW_CHANGED",
  "data" : {
    "output" : String,
    "window" : String | NULL
  }
}
```

- data.output (String)
  - The full name of the output, which includes the slot. For example, "Slot16.Out1"
- data.window (String)
  - The name of the Window where that audio will come from. For example "Window1"

## STATUS\_GROUP

Event raised when the audio source routing for an output is changed

```
{
  "category" : "OUTPUT",
  "event" : "STATUS_GROUP",
  "data" : {
    "output" : String,
    "propertyName" : String,
    "value" : String
  }
}
```

- data.output (String)
  - The full name of the output, which includes the slot. For example, "Slot16.Out1"
- data.propertyName (String)
  - The name of the input property. For example "FramelockStatus"
- data.value (String)
  - The value of the input property specified in propertyName. For example "Locked"

The supported propertyName properties are:

- HDCP\_Active
  - Valid values: Active, Off
  - Not available on SDI output modules
- HDMI
  - Valid values: Found, Not\_Found
- FramelockStatus
  - Valid values: Locked, Unlocked
- Genlock
  - Valid values: Off, Locked
- Resolution
  - SDI output modules only.
  - Raised when the resolution is changed by HDCP requirements

## PROPERTY\_CHANGED

Event raised when any of the specified properties on an output change

```
{
  "category" : "OUTPUT",
  "event" : "PROPERTY_CHANGED",
  "data" : {
    "output" : String,
    "propertyName" : String,
    "value" : String
  }
}
```

- data.output (String)
  - The name of the output where the audio mode has changed, For example "Slot5.Out1"
- data.propertyName (String)
  - The name of the canvas property. For example "AudioMute"
- data.value (String)
  - The value of the canvas property specified in propertyName. For example, "On"

The supported propertyName properties are:

- AudioMute
  - Valid values: On, Off
- AudioEnable
  - Valid values: On, Off
- CutToBlack
  - Valid values: On, Off

# Input Events

Events associated with input modules

## STATUS\_GROUP

Event raised when the audio source routing for an output is changed

```
{
  "category" : "INPUT",
  "event" : "STATUS_GROUP",
  "data" : {
    "input" : String,
    "propertyName" : String,
    "value" : String
  }
}
```

- data.input (String)
  - The full name of the input, which includes the slot. For example, "Slot1.In1"
- data.propertyName (String)
  - The name of the input property. For example "Status"
- data.value (String)
  - The value of the input property specified in propertyName. For example "OK"

The supported propertyName properties are:

- Status
  - Valid values: OK, Invalid
- Measured\_Resolution
  - Valid values: a valid resolution like 1920x1080p60 or empty
- Set\_Resolution
  - Valid values: a valid resolution like 1920x1080p60 or empty
- CanFramelockTo
  - Valid values: Yes, No
- HDCP\_Required
  - Valid values: Required, Off
- HDCP\_Status
  - Valid values: None, v1\_4, v2\_2

- HDMI
  - Valid values: Found, Not\_found
- Audio
  - Valid values: Found, Off

## Window Events

Events associated with windows

### INPUT

Event raised when the input (video) source for a Window is changed

```
{
  "category" : "WINDOW",
  "event" : "INPUT",
  "data" : {
    "window" : String
    "input" : String | NULL
  }
}
```

- data.window (String)
  - The name of the window for which the input has changed. For example, "Window1"
- data.input (String)
  - The full name of the input, which includes the slot. For example, "Slot1.In1"



# System Events

Events associated with device system

## UPDATE\_STATUS

Event indicates the status of AVIP module updates across the system. The events represent the combined status of all AVIP modules present in the system

```
{
  "category" : "SYSTEM",
  "event" : "UPDATE_STATUS",
  "data" : {
    "status" : ENUM (Booting, Updating, Ready, UpdateFailed)
  }
}
```

- data.status (String)
  - Device wide update status of all AVIP modules in a system. Ready indicates all modules are ready to be used. UpdateFailed indicates that 1 or more modules failed to update.

## POWERMODE\_CHANGED

Event indicates the status of AVIP module updates across the system. The events represent the combined status of all AVIP modules present in the system

```
{
  "category" : "SYSTEM",
  "event" : "POWERMODE_CHANGED",
  "data" : {
    "powermode" : ENUM (Standby, Resuming, Resumed)
  }
}
```

- data.powermode (String)
  - Power mode the device is entering

## HDMI Events

Events associated with HDMI based modules

### SINK\_ATTACHED

Event raised when an HDMI connection is attached to an output

```
{
  "category" : "HDMI",
  "event" : "SINK_ATTACHED",
  "data" : {
    "output" : String
  }
}
```

- data.output (String)
  - The full name of the output, which includes the slot. For example, "Slot16.Out1"

### SINK\_UNPLUGGED

Event raised when an HDMI connection is unplugged from an output

```
{
  "category" : "HDMI",
  "event" : "SINK_UNPLUGGED",
  "data" : {
    "output" : String
  }
}
```

- data.output (String)
  - The full name of the output, which includes the slot. For example, "Slot16.Out1"

## Preset Events

Events associated with presets

### TAKE

Event raised when a preset is taken (applied)

```
{
  "category" : "PRESET",
  "event" : "TAKE",
  "data" : {
    "preset" : Int
  }
}
```

- data.preset (Int)
  - The ID of the preset which has been taken (applied). For example, "1" for preset 1.

### COMPLETE

Event raised when a preset is complete. This occurs once any transitions have completed.

```
{
  "category" : "PRESET",
  "event" : "COMPLETE",
  "data" : {
    "preset" : Int
  }
}
```

- data.preset (Int)
  - The ID of the preset which has been taken (applied). For example, "1" for preset 1.

## SAVE

Event raised when a preset is saved.

```
{
  "category" : "PRESET",
  "event" : "SAVE",
  "data" : {
    "preset" : Int
  }
}
```

- data.preset (Int)
  - The ID of the preset which has been taken (applied). For example, "1" for preset 1.

## REMOVE

Event raised when a preset is removed.

```
{
  "category" : "PRESET",
  "event" : "REMOVED",
  "data" : {
    "preset" : Int
  }
}
```

- data.preset (Int)
  - The ID of the preset which has been taken (applied). For example, "1" for preset 1.

## Security Events

Events associated with system security

### USERCERT\_UPDATE\_DECRYPTED

Event raised when the decrypt phase of a user certificate is completed

```
{
  "category" : "SECURITY",
  "event" : "USERCERT_UPDATE_DECRYPTED",
  "data" : {
    "result" : ENUM (OK,Fail)
  }
}
```

- data.result (Enum)
  - The result of the user certificate package update decryption phase

### USERCERT\_UPDATE\_PARSE

Event raised when the parse phase of a user certificate is completed

```
{
  "category" : "SECURITY",
  "event" : "USERCERT_UPDATE_PARSE",
  "data" : {
    "result" : ENUM (OK,Fail)
  }
}
```

- data.result (Enum)
  - The result of the user certificate package parsing phase

## USERCERT\_UPDATE\_VALIDATION

Event raised when the validation phase of a user certificate is completed

```
{
  "category" : "SECURITY",
  "event" : "USERCERT_UPDATE_VALIDATION",
  "data" : {
    "result" : ENUM (OK,Fail)
  }
}
```

- data.result (Enum)
  - The result of the user certificate package validation phase

## USERCERT\_UPDATE\_COMPLETE

Event raised when the validation phase of a user certificate is completed

```
{
  "category" : "SECURITY",
  "event" : "USERCERT_UPDATE_COMPLETE",
  "data" : {
    "result" : ENUM (UPDATECOMPLETE,CERTIFICATE_INVALID,FILENOTFOUND,
FILECORRUPTED,DIRECTORYMISSING,WRITEERROR,DECRYPTPROBLEM)
  }
}
```

- data.result (Enum)
  - The result of the user certificate package update. UPDATECOMPLETE is a successful update. All other results indicate a failure

## Canvas Events

Events associated with canvasses

### STBDCURRENT\_CHANGED

Event raised when the storyboard for a canvas changes

```
{
  "category" : "CANVAS",
  "event" : "STBDCURRENT_CHANGED",
  "data" : {
    "canvas" : String,
    "stbd" : String
  }
}
```

- data.canvas (String)
  - The name of the canvas where the storyboard has changed, For example "Canvas1"
- data.stbd (String)
  - The name of the storyboard that is now current for the specified canvas. For example "Stbd1"

## PROPERTY\_CHANGED

Event raised when any of the specified properties on a canvas change

```
{
  "category" : "CANVAS",
  "event" : "PROPERTY_CHANGED",
  "data" : {
    "canvas" : String,
    "propertyName" : String,
    "value" : String
  }
}
```

- data.canvas (String)
  - The name of the canvas where the audio mode has changed, For example "Canvas1"
- data.propertyName (String)
  - The name of the canvas property. For example "AudioMute"
- data.value (String)
  - The value of the canvas property specified in propertyName. For example, "On"

The supported propertyName properties are:

- AudioMute
  - Valid values: On, Off
- AudioMode
  - Valid values: FromSource, FollowWindow
- AudioFollowWindow
  - Valid values: The full name of the window, for example "Window1"
- AudioSource
  - Valid values: The full name of the input source, for example "Slot1.In1"
- AudioVolume
  - Valid values: 0 to 100



## Storyboard Events

Events associated with storyboards

### ISCURRENT\_CHANGED

Event raised when the storyboard is the either activated or deactivated

```
{
  "category" : "STBD",
  "event" : "ISCURRENT_CHANGED",
  "data" : {
    "stbd" : String,
    "isCurrent" : Boolean
  }
}
```

- data.stbd (String)
  - The name of the storyboard. For example, "Stbd1"
- data.isCurrent (Boolean)
  - True when specified storyboard is active, otherwise false.

## Region Events

### CORIOview only

Events associated with regions

## EXCHANGING

### CORIOview only

Event raised when the region exchange function has started

```
{
  "category" : "REGION",
  "event" : "EXCHANGING",
  "data" : {
    "srcRegion" : String,
    "destRegion" : String
  }
}
```

- data.srcRegion (String)
  - The name of the source region. For example, "RegionA"
- data.destRegion (String)
  - The name of the region you are switching to. For example, "RegionB"

## EXCHANGED

### CORIOview only

Event raised when the region exchange function has completed

```
{
  "category" : "REGION",
  "event" : "EXCHANGED",
  "data" : {
    "srcRegion" : String,
    "destRegion" : String
  }
}
```

- data.srcRegion (String)
  - The name of the source region. For example, "RegionA"
- data.destRegion (String)
  - The name of the region you are switching to. For example, "RegionB"