ONVIF™
Device IO Service Specification

Version 16.12
December, 2016

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

CONTENTS

1 Scope

This document defines the web service interface for all physical inputs and outputs. For most inputs and outputs this is a pure get interface while for e.g. relays also configuration and control is include.

Web service usage is outside of the scope of this document. Please refer to the ONVIF core specification.

2 Normative references

ONVIF Core Specification
<http://www.onvif.org/specs/core/ONVIF-Core-Specification-v220.pdf>

3 Terms and Definitions

3.1 Definitions

Input/Output (I/O)	Currently relay ports and Video/Audio Inputs/Outputs are handled.

3.2 Abbreviations

ONVIF	Open Network Video Interface Forum

4 Overview

The DevideIO service offers commands to retrieve and configure the settings of physical inputs and outputs of a device.

The DeviceIO service supports the configuration of the following device interfaces:

- VideoOutputs
- RelayOutputs
- DigitalInputs
- Send and/or Receive serial data communication

The following commands list existing interfaces:

- GetVideoOutputs – Gets all existing video outputs of the device.
- GetVideoSources – Gets all existing video sources of the device.
- GetAudioOutputs – Gets all existing audio outputs of the device.
- GetAudioSources – Gets all existing audio sources of the device
- GetRelayOutputs– Gets all existing relay outputs of the device
- GetDigitalInputs – Gets all existing digital inputs of the device
- GetSerialPorts - Gets a list of all available serial ports and their settings.

For VideoOutputs the following commands are supported:

- *Set<device name>Configuration* – Modifies the configuration of a specific interface.
- *Get< device name >Configuration* – Gets the configuration of a specific interface.
- *Get< device name >ConfigurationOptions* – Gets the supported property values for a specific interface.

RelayOutputs supports following commands:

- SetRelayOutputSettings – Modifies the configuration of a relay output
- SetRelayOutputState – Sets the logical state

SerialPorts additionally support the following command:

- Send and/or Receive serial command - Transmit/receive generic controlling data to/from a serial device

WSDL for the DeviceIO service is specified in http://www.onvif.org/ver10/deviceio.wsdl.

Table 1: Referenced namespaces (with prefix)

Prefix	Namespace URI
env	http://www.w3.org/2003/05/soap-envelope
ter	http://www.onvif.org/ver10/error
xs	http://www.w3.org/2001/XMLSchema
tt	http://www.onvif.org/ver10/schema
tmd	http://www.onvif.org/ver10/deviceIO/wsdl

5 Service

This service offers commands to retrieve and configure the physical Inputs and Outputs of a device.

Commands to request the available video and audio in- and outputs are defined as well as commands to request the available relays. This service also offers functions to request and change the configuration of these entities.

A device that has physical sources and outputs SHALL support this service as described in [DeviceIOService.wsdl].

5.1 VideoOutputs

The VideoOutput type represents the physical Video Outputs of a device that can be connected to a monitor to display the video signal. The structure contains the Layout Settings that can be configured using the Display Service.

5.1.1 GetVideoOutputs

This command lists all available video outputs of a device. A device that has one or more physical video outputs shall support listing of available video outputs through the GetVideoOutputs command.

Table 2: GetVideoOutputs command

GetVideoOutputs	Access Class: READ_MEDIA
Message name	**Description**
GetVideoOutputsRequest	*This is an empty message.*
GetVideoOutputsResponse	*Contains a list of tokenenumerating all available video outputs of the device. If a device has no VideoOutputs an empty list is returned.* tt:ReferenceToken **Token** [0][unbounded]
Fault codes	**Description**
No specific fault codes.	

5.2 VideoOutputConfiguration

5.2.1 GetVideoOutputConfiguration

This operation requests the configuration of a Video Output. A device that has one or more Video Outputs shall support the retrieval of the VideoOutputConfiguration through this command.

Table 3: GetVideoOutputConfiguration command

GetVideoOutputConfiguration	Access Class: READ_MEDIA
Message name	**Description**
GetVideoOutputConfigurationRequest	*This message contains the token of the VideoOutput.* Tt:ReferenceToken **VideoOutputToken** [1][1]
GetVideoOutputConfigurationResponse	*This message contains the requested VideoOutputConfiguration with the matching token.* Tt:VideoOutputConfiguration **VideoOutputConfiguration** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoVideoOutput	*The requested VideoOutput indicated with **VideoOutputToken** does not exist.*

5.2.2 SetVideoOutputConfiguration

This operation modifies a video output configuration. A device that has one or more video outputs shall support the setting of its video output configuration through this command.

Table 4: SetVideoOutputConfiguration command

SetVideoOutputConfiguration	Access Class: ACTUATE
Message name	**Description**
SetVideoOutputConfiguration-Request	*The **Configuration** element contains the modified VideoOutput configuration.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* Tt:VideoOutputConfiguration **Configuration** [1][1] xs:boolean **ForcePersistence** [1][1]
SetVideoOutputConfiguration-Response	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoVideoOutput	*The requested Video Output does not exist*
env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*

5.2.3 GetVideoOutputConfigurationOptions

This operation requests the VideoOutputConfigurationOptions of a VideoOutput. A device that has one or more video outputs shall support the retrieval of VideoOutputConfigurationOptions through this command.

Table 5: GetVideoOutputConfigurationOptions command

GetVideoOutputConfigurationOptions	Access Class: READ_MEDIA
Message name	**Description**
GetVideoOutputConfiguration-OptionsRequest	*The **VideoOutputToken** element specifies the VideoOutput whose options are requested. The VideoOutput shall exist in the device* tt:ReferenceToken **VideoOutputToken**[1][1]
GetVideoOutputConfiguration-OptionsResponse	*The response contains the VideoOutputOptions of the device.* Tt:VideoOutputConfigurationOptions **VideoOutputOptions**[1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoVideoOutput	*The requested Video Output does not exist*

5.3 VideoSources

A VideoSource represents physical video input. The structure contains the pixel resolution of the video, framerate and imaging settings. The imaging settings can be manipulated through the ImagingService if supported and contains parameters for focus, exposure and brightness, for example.

5.3.1 GetVideoSources

This operation lists all available video sources for the device. The device that has one or more video inputs shall support the listing of available video sources through the GetVideoSources command.

Table 6: GetVideoSources command

GetVideoSources	Access Class: READ_MEDIA
Message name	**Description**
GetVideoSourcesRequest	*This is an empty message.*
GetVideoSourcesResponse	*Contains a list of tokenenumerating all available video sources of the device. If a device has no video sources an empty list is returned.* tt:ReferenceToken **Token** [0][unbounded]
Fault codes	**Description**
No specific fault codes.	

5.4 AudioOutputs

The Audio Output represents the physical audio outputs that can be connected to a loudspeaker.

5.4.1 GetAudioOutputs

This command lists all available audio outputs of a device. A device that has one ore more physical audio outputs shall support listing of available audio outputs through the GetAudioOutputs command.

Table 7: GetAudioOutputs command

GetAudioOutputs	Access Class: READ_MEDIA
Message name	**Description**
GetAudioOutputsRequest	*This is an empty message.*
GetAudioOutputsResponse	*Contains a list of tokenenumerating all available audio outputs of the device. If a device has no audio outputs an empty list is returned.* tt:ReferenceToken **Token** [0][unbounded]
Fault codes	**Description**
env:Receiver *ter:ActionNotSupported* *ter:AudioOutputNotSupported*	*Audio or Audio Outputs are not supported by the Device*

5.5 AudioSources

An AudioSource represents unencoded audio input and states the number of input channels

5.5.1 GetAudioSources

This operation lists all available audio sources for the device. The device that has one or more audio sources shall support the listing of available audio inputs through the GetAudioSources command.

Table 8: GetAudioSources command

GetAudioSources	Access Class: READ_MEDIA
Message name	**Description**
GetAudioSourcesRequest	*This is an empty message.*
GetAudioSourcesResponse	*Contains a list of tokenenumerating all available audio sources of the device. If a device has no audio sources an empty list is returned.* tt:ReferenceToken **Token** [0][unbounded]
Fault codes	**Description**

env:Receiver ter:ActionNotSupported ter:AudioOutputNotSupported	*The devices does not support audio.*

5.6 Relay Outputs

The Input/Output (I/O) commands are used to control the state or observe the status of the I/O ports. If the device has I/O ports, then it shall support the I/O commands.

Handling of relay outputs is also defined in DeviceManagement, see ONVIF Core Specification secion Input/Output.

5.6.1 Get relay outputs

This operation gets a list of all available relay outputs and their settings.

Table 9: GetRelayOutputs command

GetRelayOutputs	Access Class: READ_MEDIA
Message name	**Description**
GetRelayOutputsRequest	*This is an empty message.*
GetRelayOutputsResponse	*This message contains an array of relay outputs.* Tt:RelayOutput **RelayOutputs** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.6.2 Get relay output options

Request the available settings and ranges for one or all relay outputs. The method shall returns the information for exactly one output when a RelayOutputToken is provided as request parameter. Otherwise the method shall return the information for all relay outputs.

A device that has one or more RelayOutputs should support this command..

Two examples:

2) Device supports PT1S to PT120S:

```
<tmd:RelayOutputOptions token='44'>
  <tmd:Mode>Monostable</tmd:Mode>
  <tmd:DelayTimes>1 120</tmd:DelayTimes>
</tmd:RelayOutputOptions>
```

2) Device supports values PT0.5S, PT1S, PT2s and PT1M:

```
<tmd:RelayOutputOptions token='123'>
  <tmd:Mode>Monostable</tmd:Mode>
  <tmd:DelayTimes Discrete='True'>0.5 1 2 60</tmd:DelayTimes>
</tmd:RelayOutputOptions>
```

Table 10: GetRelayOutputOptions command

GetRelayOutputOptions	Access Class: PRE_AUTH
Message name	**Description**
GetRelayOutputOptionsRequest	- *"RelayOutputToken": Optional token reference to the requested relay output.* Tt:ReferenceToken **RelayOutputToken** [0][1]
GetRelayOutputOptionsResponse	*This message contains an array of relay output options.* Tmd:RelayOutputOptions **RelayOutputOptions** [0][unbounded]
Fault codes	**Description**
	No command specific faults!

5.6.3 Set relay output settings

This operation sets the settings of a relay output.

The relay can work in two relay modes:

- Bistable – After setting the state, the relay remains in this state.

- Monostable – After setting the state, the relay returns to its idle state after the specified time.

The physical idle state of a relay output can be configured by setting the IdleState to 'open' or 'closed' (inversion of the relay behaviour).

Idle State 'open' means that the relay is open when the relay state is set to 'inactive' through the trigger command (see Section 5.6.4) and closed when the state is set to 'active' through the same command.

Idle State 'closed' means, that the relay is closed when the relay state is set to 'inactive' through the trigger command (see Section 5.6.4) and open when the state is set to 'active' through the same command.

The Duration parameter of the Properties field "DelayTime" describes the time after which the relay returns to its idle state if it is in monostable mode. If the relay is set to bistable mode the value of the parameter shall be ignored.

Table 11: SetRelayOutputSettings command.

SetRelayOutputSettings	Access Class: ACTUATE
Message name	**Description**
SetRelayOutputSettingsRequest	*This message contains:* • *“RelayOutputToken”: Token reference to the requested relay output.* • *“RelayOutputSettings”: The settings of the relay* . tt:ReferenceToken **RelayOutputToken** [1][1] tt:RelayOutputSettings **RelayOutputSettings** [1][1]
SetRelayOutputSettingsResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:RelayToken	*Unknown relay token reference.*
Env:Sender ter:InvalidArgVal ter:ModeError	*Monostable delay time not valid*

5.6.4 Trigger relay output

This operation triggers a relay output[1].

Table 12: SetRelayOutputState command

SetRelayOutputState	Access Class: ACTUATE
Message name	**Description**
SetRelayOutputStateRequest	*This message contains:* • *RelayOutputToken”: Token reference to the requested relay output.* • *“LogicalState”: Trigger request, i.e., active or inactive.* Tt:ReferenceToken **RelayOutputToken** [1][1] tt:RelayLogicalState **LogicalState** [1][1]
SetRelayOutputStateResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:RelayToken	*Unknown relay token reference.*

5.7 Digital Inputs

The DigitalInput type represents the integrated physical digital inputs of a device which enable connection to external devices, such as doorbells, detectors, lights or switches (device that can be toggled between an open and closed circuit).

[1] There is no GetRelayState command; the current logical state of the relay output is transmitted via notification and their properties.

5.7.1 GetDigitalInputs

This command lists all available digital inputs of a device. A device that signals support for digital inputs via its capabilities shall support listing of available inputs through the GetDigitalInputs command. A device having one or more digital inputs shall support the GetDigitalInputs command.

Table 13: GetDigitalInputs command

GetDigitalInputs	Access Class: READ_MEDIA
Message name	**Description**
GetDigitalInputsRequest	*This is an empty message.*
GetDigitalInputsResponse	*Contains a list of structures describing all available digital inputs of the device. If a device has no digital inputs an empty list is returned.* Tt:DigitalInput **DigitalInputs** [0][unbounded]
Fault codes	**Description**
No specific fault codes.	

5.7.2 GetDigitalInputConfigurationOptions

This operation retrieves the digital input configuration options when the digital input configuration token is known. If a specific digital input is specified, the options shall concern that particular configuration. If a token is not specified, the options shall be considered generic for the device. A device shall support the GetDigitalInputConfigurationOptions command if the device signals capability of digital input configuration via DigitalInputOptions capability.

Table 14: GetDigitalInputConfigurationOptions command

GetDigitalInputConfigurationOptions	Access Class: READ_SYSTEM
Message name	**Description**
GetDigitalInputConfigurationOptionsRequest	*This message contains an optional token of a digital input configuration. If no token is specified, the options are applicable for all digital inputs.* tt:ReferenceToken **Token**[0][1]
GetDigitalInputConfigurationOptionsResponse	*The message contains the configuration options.* tt:DigitalInputOptions **DigitalInputOptions**[1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested configuration indicated by Token does not exist.*

5.7.3 SetDigitalInputConfigurations

This operation modifies existing digital input configurations. When applying multiple configuration settings, the expected behaviour is to configure all or none. If the setting is invalid, the expected behaviour is to configure none, and a remark is provided in fault to indicate which digital input configuration is not acceptable. A device shall support the

SetDigitalInputConfigurations command if the device signals capability of digital input configuration via DigitalInputOptions capability.

Table 15: SetDigitalInputConfigurations command

SetDigitalInputConfigurations	Access Class: ACTUATE
Message name	**Description**
SetDigitalInputConfigurationsRequest	*The* **DigitalInputs** *element contains the modified digital input configurations containing the unique configuration token. The digital input configuration shall exist in the device.* tt:DigitalInput **DigitalInputs**[1][unbounded]
SetDigitalInputConfigurationsResponse	*This message is empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NoConfig	*The requested digital input configuration does not exist.*
env:Sender ter:InvalidArgVal ter:SettingsInvalid	*The requested settings are incorrect. Token need to be returned to indicate which digital input configuration is not acceptable.*

5.8 SerialPorts

The SerialPort type represents the physical serial port on the device and allows serial data to be read and written.

5.8.1 GetSerialPorts

This command lists all available serial ports of a device. A device that has one or more physical serial ports shall support listing of available serial ports through the GetSerialPorts command.

Table 16: GetSerialPorts command

GetSerialPorts	Access Class: READ_SYSTEM
Message name	**Description**
GetSerialPortsRequest	*This is an empty message.*
GetSerialPortsResponse	*Contains a list of structures describing all available serial ports of the device. If a device has no serial ports an empty list is returned* tmd:SerialPort **SerialPort**[0][unbounded]
Fault codes	**Description**
No specific fault codes.	

5.8.2 GetSerialPortConfiguration

This operation gets a list of all available Serial ports and their settings.

Table 17: GetSerialPortConfiguration command

GetSerialPortConfiguration	Access Class: READ_SYSTEM
Message name	**Description**
GetSerialPortConfigurationRequest	*This message contains the token of the serial port.* Tt:ReferenceToken **SerialPortToken**[1][1]
GetSerialPortConfigurationResponse	*This message contains an array of SerialPortConfiguration.* Tmd:SerialPortConfiguration **SerialPortConfiguration**[1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidSerialPort	*The supplied serial port token does not exist.*

5.8.3 SetSerialPortConfiguration

This operation sets the setting of serial port.

Table 18: SetSerialPortConfiguration command

SetSerialPortConfiguration	Access Class: WRITE_SYSTEM
Message name	**Description**
SetSerialPortConfigurationRequest	*The **SerialPortToken** element specifies the serial port whose configuration is to be modified.* *The **SerialPortConfiguration** element contains the modified serial port configuration.* *The **ForcePersistence** element determines if the configuration changes shall be stored and remain after reboot. If true, changes shall be persistent. If false, changes MAY revert to previous values after reboot.* Tt:ReferenceToken **SerialPortToken**[1][1] tmd:SerialPortConfiguration **SerialPortConfiguration** [1][1] xs:boolean **ForcePersistence**[1][1]
SetSerialPortConfigurationResponse	*This is an empty message.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidSerialPort	*The supplied serial port token does not exist.*
Env:Sender ter:InvalidArgVal ter:ConfigModify	*The configuration parameters are not possible to set.*

5.8.4 GetSerialPortConfigurationOptions

This operation requests the SerialPortConfigurationOptions of a SerialPort. A device that has one or more SerialPorts shall support this command.

Table 19: GetSerialPortConfigurationOptions command

GetSerialPortConfigurationOptions	Access Class: READ_SYSTEM
Message name	**Description**
GetSerialPortConfigurationOptions-Request	*The **SerialPortToken** element specifies the Serial Port whose options are requested.* Tt:ReferenceToken **SerialPortToken**[1][1]
GetSerialPortConfigurationOptions-Response	tmd:SerialPortConfigurationOptions **SerialPortConfigurationOptions** [1][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidSerialPort	*The supplied serial port token does not exist.*

5.8.5 Send and/or Receive serial command

This section describes operations to transmit/receive *generic* controlling data to/from a serial device that is connected to the serial port of the device.

This operation can be used for the following purposes.

- Transmitting arbitrary data to the connected serial device
- Receiving data from the connected serial device
- Transmitting arbitrary data to the connected serial device and then receiving its response data

In order to make use of this command for the above purpose, this specification defines the input parameter structure as follows.

- token

  This element shall be present in the request. It indicates the physical serial port reference to be used when this request is invoked.

- SerialData

  This element is optional to be put in the request. When transmitting serial data is needed, the request should contain the element.

- TimeOut

  This element is optional to be put in the request. Depending on the specified value, it is possible for various configurations as follows.

  (i) TimeOut > PT0S: Indicates that the command should be responded back within the specified period of time. In the case the device received the data which meets one of the following conditions of DataLength and Delimiter, the device should respond back with the received data instead of waiting for the specified time.

  (ii) TimeOut = PT0S: Indicates that the command should be responded back immediately (Non-blocking). It will be used in the case of only transmitting data.

  (iii) TimeOut = -PT1S: Indicates that the command should be responded after one of the following conditions (DataLength / Delimiter) is met. How long the device can hold the blocking state is *vendor specific*.

  If this element is not present in the request, the command should be responded after one of the following conditions (DataLength / Delimiter) is met. How long the device can hold the blocking state is *vendor specific*.

- DataLength

  This element is optional to be put in the request. This element may be put in the case that data length returned from the connected serial device is already determined as some fixed bytes length. It indicates the length of received data which can be regarded as available.

- Delimiter

  This element is optional to be put in the request. This element may be put in the case that the delimiter codes returned from the connected serial device is already known. It indicates the termination data sequence of the responded data. In case the string has more than one character a device shall interpret the whole string as a single delimiter. Furthermore a device shall return the delimiter character(s) to the client.

A device that indicates generic serial communication service capability shall support this command.

Table 20: Send and/or Receive serial command

SendReceiveSerialCommand	Access Class: ACTUATE
Message name	**Description**
SendReceiveSerialCommandRequest	*See abiove for information about the paramaters.* Tmd:SerialData**SerialData** [0][1] xs:duration **TimeOut** [0][1] xs:integer **DataLength** [0][1] xs:string **Delimiter** [0][1]
SendReceiveSerialCommandResponse	*This message contains the serial data.* Tmd:SerialData **SerialData** [0][1]
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidSerialPort	*The supplied serial port token does not exist.*
Env:Sender ter:OperationProhibited ter:DataLengthOver	*Number of available bytes exceeded.*
Env:Sender ter:OperationProhibited ter:DelimiterNotSupported	*Sequence of character (delimiter) is not supported.*

5.9 Capabilities

The capabilities reflect optional functions and functionality of a service. The information is static and does not change during device operation. The following capabilites are available:

VideoSources: Number of video sources (defaults to none).

VideoOutputs: Number of video outputs (defaults to none).

AudioSources: Number of audio sources (defaults to none).

AudioOutputs: Number of audio outputs (defaults to none).

RelayOutputs: Number of relay outputs (defaults to none).

DigitalInputs: Number of digital inputs (defaults to none).

SerialPorts: Number of serial ports (defaults to none).

DigitalInputOptions: Indicates support for DigitalInput configuration of the idle state (defaults to false)

Table 21: GetServiceCapabilities command

GetServiceCapabilities	Access Class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This is an empty message.*
GetServiceCapabilitiesResponse	*The capability response message contains the requested service capabilities using a hierarchical XML capability structure.* Tmd:Capabilities **Capabilities** [1][1]
Fault codes	**Description**
	No command specific faults!

5.10 Events

5.10.1 DigitalInput State Change

A device that signals support for digital inputs in its capabilities shall provide the following event whenever one of its input state changes:

```
Topic: tns1:Device/Trigger/DigitalInput

<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="InputToken" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="LogicalState" Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

Digital Input LogicalState can be either set at "true" to represent the circuit in the closed state or set at "false" to represent the circuit in the open state.

5.10.2 Relay Output Trigger

A device that signals RelayOutputs in its capabilities should provide the Trigger event whenever its relay output state is changed. A device shall use the following topic and message format:

```
Topic: tns1:Device/Trigger/Relay

<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="RelayToken" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="LogicalState" Type="tt:RelayLogicalState"/>
  </tt:Data>
</tt:MessageDescription>
```

5.11 Service specific fault codes

The table below lists the DeviceIO service specific fault codes. Additionally, each command can also generate a generic fault as defined in the ONVIF Core specification.

Table 22: DeviceIO service specific fault codes

Fault Code	Parent Subcode Subcode	Fault Reason	Description
env:Sender	ter:InvalidArgVal Ter:ConfigModify	Invalid configuration parameters	The configuration parameters are not possible to set.
env:Sender	ter:InvalidArgVal Ter:NoVideoOutput	Video output token does not exist.	The requested VideoOutput indicated with **VideoOutputToken** does not exist.
env:Sender	ter:InvalidArgVal Ter:NoVideoSource	Video source token does not exist.	The requested VideoSource indicated with **VideoSourceToken** does not exist.
env:Sender	ter:InvalidArgVal Ter:NoAudioOutput	Audio output token does not exist.	The requested AudioOutput indicated with **AudioOutputToken** does not exist.
env:Sender	ter:InvalidArgVal Ter:NoAudioSource	Audio source token does not exist.	The requested AudioSource indicated with **AudioSourceToken** does not exist.
env:Sender	ter:InvalidArgVal Ter:RelayToken	Unknown relay token reference	The requested RelayOutput indicated **RelayOutputToken** does not exist.
env:Sender	ter:InvalidArgVal ter:ModeError	Monostable delay time not valid	
env:Sender	ter:InvalidArgVal Ter:InvalidSerialPort	Serial port token not valid	The supplied serial port token does not exist.
env:Sender	ter:OperationProhibited Ter:DataLengthOver	Data length over	Number of available bytes exceeded.
env:Sender	ter:OperationProhibited Ter:DelimiterNotSupported	Delimiter is not supported	Sequence of character (delimiter) is not supported.
env:Sender	ter:InvalidArgVal ter:NoConfig	The requested digital input configuration does not exist.	The requested configuration indicated by the configuration token does not exist.
env:Sender	ter:InvalidArgVal	Invalid configuration.	The requested settings are incorrect.

Annex A. Deprecated Interfaces

A.1 Configuration of media source and output

The duplicated definition and interfaces for the handling of media source and output configuration between media service, have been deprecated with release 16.12. The following interfaces have been removed from the specification:

- GetVideoSourceConfiguration
- SetVideoSourceConfiguration
- GetVideoSourceConfigurationOptions
- GetAudioOutputConfiguration
- SetAudioOutputConfiguration
- GetAudioOutputConfigurationOptions
- GetAudioSourceConfiguration
- SetAudioSourceConfiguration
- GetAudioSourceConfigurationOptions

The definitions are available via the link http://www.onvif.org/specs/srv/io/ONVIF-DeviceIo-Service-Spec-v1606.pdf

Annex B. Revision History

Rev.	Date	Editor	Changes
2.1	Jul-2011	Hans Busch	Split from Core 2.0 Change Request 232
2.1.1	Jan-2012	Hans Busch	Change Requests 259, 291, 535
2.2	May-2012	M.Tonomura	Add serial port function
2.2.1	Dec-2012	Hans Busch	Change Request 708
2.4.1	Dec-2013	Michio Hirai	Change Request 1217
2.5	Dec-2014	Michio Hirai	Change Request 1492, 1516
2.6	Jun-2015	Michio Hirai	Change Request 1589, 1639, 1642
2.6.1	Aug-2015	Tom Yui Hans Busch	Add Digital Input Configuration Change Request 1633
16.06	Jun-2016	Hiroyuki Sano	Change Request 1802, 1853
16.12	Dec-2016	Hiroyuki Sano	Change Request 1985