

ONVIF™ Receiver Service Specification

Version 2.1.1
January, 2012



© 2008-2011 by ONVIF: Open Network Video Interface Forum Inc.. All rights reserved.

Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

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** **4**
- 2 Normative references** **4**
- 3 Terms and Definitions** **4**
 - 3.1 Definitions.....4
 - 3.2 Abbreviations4
- 4 Overview** **4**
 - 4.1.1 Synchronization Points5
- 5 Service** **6**
 - 5.1 Persistence6
 - 5.2 Receiver modes6
 - 5.3 Receiver commands6
 - 5.3.1 Get Receivers6
 - 5.3.2 Get Receiver7
 - 5.3.3 Create Receiver7
 - 5.3.4 Delete Receiver8
 - 5.3.5 Configure Receiver8
 - 5.3.6 SetReceiverMode8
 - 5.3.7 GetReceiverState9
 - 5.4 Capabilities.....9
 - 5.5 Events10
 - 5.5.1 ChangeState10
 - 5.5.2 Connection Failed.....10
 - 5.6 Service specific data types.....10
 - 5.6.1 • Receiver10
 - 5.6.2 • ReceiverConfiguration.....11
 - 5.6.3 • ReceiverStateInformation.....11
 - 5.7 Service specific fault codes.....12
- Annex A. Revision History** **13**

1 Scope

This document defines the web service interface for configuration a network streaming receiver. The receiver is used for displaying and recording audio video streams. Additionally the associated events are defined.

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-v211.pdf>>

ONVIF DeviceIO Service Specification

<<http://www.onvif.org/specs/srv/io/ONVIF-DeviceIo-Service-Spec-v211.pdf>>

ONVIF Streaming Specification

<<http://www.onvif.org/specs/stream/ONVIF-Streaming-Spec-v211.pdf>>

3 Terms and Definitions

3.1 Definitions

Metadata All streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications).

3.2 Abbreviations

JPEG	Joint Photographic Expert Group
MPEG-4	Moving Picture Experts Group - 4
RTSP	Real Time Streaming Protocol

4 Overview

A receiver is an object that acts as an RTSP client endpoint. Receivers are used by other services that consume media streams, such as the Display, Recording and Analytics Device services. A receiver has a configuration that determines the RTSP endpoint to which it should connect and the connection parameters it should use.

A receiver can operate in three distinct modes:

AlwaysConnect. The receiver attempts to maintain a persistent connection to the configured endpoint.

NeverConnect. The receiver does not attempt to connect.

AutoConnect. The receiver connects on demand, as required by consumers of the media streams.

A single receiver may be used by more than one consumer. For example, in order to record a stream and also perform analytics on it, both a recording job and an analytics engine could be attached to the same receiver. If the receiver uses the “Auto Connect” mode, it will connect whenever either the recording job or the analytics engine is active, and disconnect when neither of them are active.

Receivers may be created and deleted either manually, by calling the CreateReceiver and DeleteReceiver operations in the Receiver Service, or automatically by other services. For

example, if a recording job is created with the “AutoCreateReceiver” option, it will automatically create and attach to a Receiver. Deleting the recording job will also delete the receiver.

WSDL for this service is specified in <http://www.onvif.org/onvif/ver10/receiver.wsdl>.

4.1.1 Synchronization Points

Because receivers use RTSP addresses to specify the source of the stream, they do not necessarily have access to the web services interface of the transmitter. This means that they cannot use the SetSynchronizationPoint command described in the section “Synchronization Point” of the ONVIF Streaming Specification..

Instead, receivers should use the PLI message described in [RFC 4585] to request a synchronization point.

5 Service

This service offers commands to manage Receiver objects, which are used to receive media streams from other devices. A Receiver object contains the information how to setup the stream, the mode of the receiver and the Stream Uri (MediaUri). A device shall at least support Media Uris of 128 octet length. The Receiver - MaximumRTSPURILength capability indicates the maximum length supported by the device. The Receiver Service shall be implemented by devices that can receive media streams.

The IP or DNS address in the transmit URI given to the receiver, is the address that the device hosting the receiver service will use to access the transmit device. If, for example, the client has to communicate through a NAT router to access the transmitter and the receiver, the transmitter address that the client gives the receiver (in this case a local network address) may not be the same address that the client would use to access the transmitter (in this case an external network address).

A device shall support RTP transfer via RTP and RTP transfer via RTSP/HTTP/TCP, see ONVIF Streaming Specification. A device may support other RTP transport protocols and shall indicate what it supports with the appropriate capability, see 5.4.

5.1 Persistence

All the objects created within the receiver service shall be persistent – i.e. they shall survive a power cycle. Likewise, all the configuration data in the objects shall be persistent.

5.2 Receiver modes

A receiver can operate in three distinct modes:

AlwaysConnect. The receiver attempts to maintain a persistent connection to the configured endpoint.

NeverConnect. The receiver does not attempt to connect.

AutoConnect. The receiver connects on demand, as required by consumers of the media streams.

5.3 Receiver commands

This section describes the commands offered by the Receiver Service.

5.3.1 Get Receivers

This operation lists all receivers that currently exist on the device. The Receiver Service shall support this command.

Table 1: GetReceivers command

GetReceivers		Access Class: READ_MEDIA
Message name	Description	
GetReceiversRequest	<i>This message is empty.</i>	

GetReceiversResponse	Contains a list of receivers. tt:Receiver Receivers [0][unbounded]
Fault codes	Description
No specific fault codes.	

5.3.2 Get Receiver

This operation retrieves the details of a specific receiver whose token is known to the client. The Receiver Service shall support this command.

Table 2: GetReceiver command

GetReceiver	Access Class: READ_MEDIA
Message name	Description
GetReceiverRequest	Contains the token of the requested receiver. tt:ReferenceToken ReceiverToken [1][1]
GetReceiverResponse	Contains the details of the requested receiver. tt:Receiver Receiver [1][1]
Fault codes	Description
env:Sender ter:InvalidArgVal ter:UnknownToken	The receiver indicated by ReceiverToken does not exist.

5.3.3 Create Receiver

This operation creates a new receiver. The Receiver Service shall support this command.

Table 3: CreateReceiver command

CreateReceiver	Access Class: ACTUATE
Message name	Description
CreateReceiverRequest	Contains the initial configuration of the receiver. tt:ReceiverConfiguration Configuration [1][1]
CreateReceiverResponse	Contains the details of the receiver that was created. tt:Receiver Receiver [1][1]
Fault codes	Description
env:Sender ter:InvalidArgVal ter:BadConfiguration	The specified configuration is invalid.
env:Receiver ter:Action ter:MaxReceivers	The maximum supported number of receivers has been reached.

5.3.4 Delete Receiver

This operation deletes an existing receiver. A receiver MAY NOT be deleted if it is in use. The Receiver Service shall support this command.

Table 4: DeleteReceiver command

DeleteReceiver		Access Class: ACTUATE
Message name	Description	
DeleteReceiverRequest	<i>Contains the token of the receiver to be deleted.</i>	
	tt:ReferenceToken ReceiverToken [1][1]	
DeleteReceiverResponse	<i>This message is empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:UnknownToken	<i>The receiver indicated by ReceiverToken does not exist.</i>	
env: Receiver ter:Action ter:CannotDeleteReceiver	<i>It is not possible to delete the specified receiver, for example because it is currently in use.</i>	

5.3.5 Configure Receiver

This operation configures a receiver. The Receiver Service shall support this command.

Table 5: ConfigureReceiver command

ConfigureReceiver		Access Class: ACTUATE
Message name	Description	
ConfigureReceiverRequest	<i>Contains the token of the requested receiver and the new configuration.</i>	
	tt:ReferenceToken ReceiverToken [1][1] tt:ReceiverConfiguration Configuration [1][1]	
ConfigureReceiverResponse	<i>This message is empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:UnknownToken	<i>The receiver indicated by ReceiverToken does not exist.</i>	
env:Sender ter:InvalidArgVal ter:BadConfiguration	<i>The specified configuration is invalid.</i>	

5.3.6 SetReceiverMode

This operation may be used to set the mode of the receiver independently of the rest of its configuration. The Receiver Service shall support this command.

Table 6: SetReceiverMode command

SetReceiverMode		Access Class: ACTUATE
Message name	Description	
SetReceiverModeRequest	<i>Contains the token of the requested receiver and the new mode.</i>	
	tt:ReferenceToken ReceiverToken [1][1] tt:ReceiverMode ReceiverMode [1][1]	
SetReceiverModeResponse	<i>This message is empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:UnknownToken	<i>The receiver indicated by ReceiverToken does not exist.</i>	

5.3.7 GetReceiverState

This operation determines whether the receiver is currently disconnected, connected or attempting to connect. The Receiver Service shall support this command.

Table 7: GetReceiverState command

GetReceiverState		Access Class: READ_MEDIA
Message name	Description	
GetReceiverStateRequest	<i>Contains the token of the requested receiver.</i>	
	tt:ReferenceToken ReceiverToken [1][1]	
GetReceiverStateResponse	<i>Contains the current state of the receiver.</i>	
	tt:ReceiverState State [1][1]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:UnknownToken	<i>The receiver indicated by ReceiverToken does not exist.</i>	

5.4 Capabilities

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

- RTP_Multicast:** Indication if the device supports receiving of RTP Multicast.
- RTP_TCP:** Indication if the device supports receiving of RTP over TCP.
- RTP_RTSP_TCP:** Indication if the device supports receiving of RTP over RTSP over TCP.
- SupportedReceivers:** The maximum number of receivers the device supports..

MaximumRTSPURILength: The maximum length allowed for RTSP URIs.

Table 8: GetServiceCapabilities command

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

5.5 Events

The receiver service shall dispatch events through the event service. It shall be capable of generating the events listed in this chapter whenever the condition that fires the event occurs.

5.5.1 ChangeState

Whenever a receiver changes state, the device shall dispatch the following event:

```
Topic: tns1: Receiver/ChangeState
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="ReceiverToken" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:SimpleItemDescription Name="NewState" Type="tt:ReceiverState"/>
    <tt:SimpleItemDescription Name="MediaUri" Type="tt:MediaUri" minOccurs="0"/>
  </tt>Data>
</tt:MessageDescription>
```

5.5.2 Connection Failed

If a receiver fails to establish a connection, the device shall dispatch the following event:

```
Topic: tns1: Receiver/ConnectionFailed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="ReceiverToken" Type="tt:ReferenceToken"/>
  </tt:Source>
  <tt>Data>
    <tt:SimpleItemDescription Name="MediaUri" Type="tt:MediaUri"/>
  </tt>Data>
</tt:MessageDescription>
```

5.6 Service specific data types

5.6.1 • Receiver

Description of a receiver, including its token and configuration.

```
<xs:complexType name="Receiver"/>
```

```
<xs:element name="Token" type="tt:ReferenceToken"/>
<xs:element name="Configuration" type="tt:ReceiverConfiguration"/>
</xs:complexType>
```

- **Token**
Unique identifier of the receiver.
- **Configuration**
Describes the configuration of the receiver.

5.6.2 • ReceiverConfiguration

Describes the configuration of a receiver.

```
<xs:complexType name="ReceiverConfiguration"/>
  <xs:element name="Mode" type="tt:ReceiverMode"/>
  <xs:element name="MediaUri" type="xs:anyURI"/>
  <xs:element name="StreamSetup" type="tt:StreamSetup"/>
</xs:complexType>
```

- **Mode**
The following connection modes are defined: AlwaysConnect, NeverConnect, AutoConnect.
- **MediaUri**
Details of the URI to which the receiver should connect.
- **StreamSetup**
Stream connection parameters.

5.6.3 • ReceiverStateInformation

Contains information about a receiver's current state.

```
<xs:complexType name="ReceiverStateInformation"/>
  <xs:element name="State" type="tt:ReceiverState"/>
  <xs:element name="AutoCreated" type="xs:boolean"/>
</xs:complexType>
```

- **State**
The connection state of the receiver may have one of the following states: NotConnected, Connecting, Connected.
- **AutoCreated**
Indicates whether or not the receiver was created automatically.

5.7 Service specific fault codes

Table 9 lists the receiver service specific fault codes. Additionally, each command can also generate a generic fault.

The specific faults are defined as subcode of a generic fault. The parent generic subcode is the *subcode* at the top of each row below and the specific fault *subcode* is at the bottom of the cell.

Table 9: Service specific fault codes

Fault Code	Parent Subcode	Fault Reason	Description
	Subcode		
Env:Sender	ter:InvalidArgVal	Receiver does not exist.	The receiver indicated by ReceiverToken does not exist.
	ter:UnknownToken		
Env:Sender	ter:Action	Maximum number of receivers has been reached.	The maximum supported number of receivers has been reached.
	ter:MaxReceivers		
Env:Sender	ter:InvalidArgVal	The StreamSetup is not supported.	Specification of StreamType or Transport part in ReceiverConfiguration StreamSetup is not supported.
	ter:BadConfiguration		
Env:Sender	ter:Action	It is not possible to delete the receiver.	It is not possible to delete the specified receiver, for example because it is currently in use.
	ter:CannotDeleteReceiver		

Annex A. Revision History

Rev.	Date	Editor	Changes
2.1	Jul-2011	Hans Busch	Split from Core 2.0 Change Request 173
2.1.1	Jan-2012	Hans Busch	Change Request 535