



# XMPP

## XEP-0180: Jingle Video via RTP

Peter Saint-Andre  
<mailto:peter@andyet.net>  
<xmpp:stpeter@stpeter.im>  
<https://stpeter.im/>

Milton Chen  
<mailto:Milton.Chen@vseelab.com>

2008-06-04  
Version 0.13

Status	Type	Short Name
Retracted	Standards Track	NOT_YET_ASSIGNED

Note: This specification has been retracted in favor of XEP-0167, which now consolidates both audio and video chat via RTP and therefore contains the content originally published in this specification; please refer to XEP-0167 for the most up-to-date definition of XMPP video chat. This specification defines a Jingle application type for negotiating a video chat or other video session. The application type uses the Real-time Transport Protocol (RTP) for the underlying media exchange and provides a straightforward mapping to Session Description Protocol (SDP) for interworking with SIP media endpoints.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 by the [XMPP Standards Foundation](#) (XSF).

## Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

## Warranty

## NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. ##

## Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

## Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>1</b>
<b>3</b>	<b>Jingle Conformance</b>	<b>1</b>
<b>4</b>	<b>Application Format</b>	<b>2</b>
<b>5</b>	<b>Negotiating a Jingle Video Session</b>	<b>4</b>
<b>6</b>	<b>Mapping to Session Description Protocol</b>	<b>7</b>
<b>7</b>	<b>Error Handling</b>	<b>8</b>
<b>8</b>	<b>Determining Support</b>	<b>9</b>
<b>9</b>	<b>Informational Messages</b>	<b>10</b>
<b>10</b>	<b>Implementation Notes</b>	<b>10</b>
	10.1 Codecs . . . . .	10
<b>11</b>	<b>Security Considerations</b>	<b>10</b>
<b>12</b>	<b>IANA Considerations</b>	<b>10</b>
<b>13</b>	<b>XMPP Registrar Considerations</b>	<b>10</b>
	13.1 Protocol Namespaces . . . . .	10
	13.2 Jingle Application Formats . . . . .	11
<b>14</b>	<b>XML Schemas</b>	<b>11</b>
	14.1 Application Format . . . . .	11
	14.2 Errors . . . . .	12

## 1 Introduction

*Note: This specification has been retracted in favor of [Jingle RTP Sessions \(XEP-0167\)](#)<sup>1</sup>, which now consolidates both audio and video chat via RTP and therefore contains the content originally published in this specification; please refer to XEP-0167 for the most up-to-date definition of XMPP video chat.*

[Jingle \(XEP-0166\)](#)<sup>2</sup> can be used to initiate and negotiate a wide range of peer-to-peer sessions. One session type of interest is video chat. This document specifies a format for describing Jingle video sessions, where the media exchange occurs using the Real-time Transport Protocol (see [RFC 3550](#)<sup>3</sup>).

## 2 Requirements

The Jingle application format defined herein is designed to meet the following requirements:

1. Enable negotiation of parameters necessary for video chat.
2. Map these parameters to the Session Description Protocol (SDP; see [RFC 4566](#)<sup>4</sup>) to enable interoperability.
3. Define informational messages related to video chat.

## 3 Jingle Conformance

In accordance with Section 8 of XEP-0166, this document specifies the following information related to the Jingle Video via RTP application type:

1. The application format negotiation process is defined in the [Negotiating a Jingle Video Session](#) section of this document.
2. The semantics of the <description/> element are defined in the [Application Format](#) section of this document.
3. A mapping of Jingle semantics to the Session Description Protocol is provided in the [Mapping to Session Description Protocol](#) section of this document.

---

<sup>1</sup>XEP-0167: Jingle RTP Sessions <<https://xmpp.org/extensions/xep-0167.html>>.

<sup>2</sup>XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

<sup>3</sup>RFC 3550: RTP: A Transport Protocol for Real-Time Applications <<http://tools.ietf.org/html/rfc3550>>.

<sup>4</sup>RFC 4566: SDP: Session Description Protocol <<http://tools.ietf.org/html/rfc4566>>.

4. A Jingle video session SHOULD use a lossy transport method such as [Jingle Raw UDP Transport Method \(XEP-0177\)](#)<sup>5</sup> or the "ice-udp" method specified in [Jingle ICE-UDP Transport Method \(XEP-0176\)](#)<sup>6</sup>.
5. Content is to be sent and received as follows:
  - Outbound video content shall be encoded into RTP packets and each packet shall be sent individually over the transport. Each inbound packet received over the transport is an RTP packet.

## 4 Application Format

A Jingle video session is described by a content type that contains one application format and one transport method. The application format consists of one or more encodings contained within a wrapper <description/> element qualified by the 'urn:xmpp:tmp:jingle:apps:video-rtp' namespace (see [Protocol Namespaces](#) regarding issuance of one or more permanent namespaces). In the language of RFC 4566 each encoding is a payload-type; therefore, each <payload-type/> element specifies an encoding that can be used for the audio stream, as illustrated in the following example.

Listing 1: Video description format

```
<description xmlns='urn:xmpp:tmp:jingle:apps:video-rtp'>
  <payload-type id='96' name='theora' clockrate='90000'>
    <parameter name='height' value='720' />
    <parameter name='width' value='1280' />
    <parameter name='delivery-method' value='inline' />
    <parameter name='configuration' value='somebase16string' />
    <parameter name='sampling' value='YCbCr-4:2:2' />
  </payload-type>
  <payload-type id='28' name='nv' clockrate='90000' />
  <payload-type id='25' name='CelB' clockrate='90000' />
  <payload-type id='32' name='MPV' clockrate='90000' />
</description>
```

The <description/> element is intended to be a child of a <content/> element as specified in XEP-0166.

The <description/> element SHOULD possess a 'profile' attribute that specifies the profile of RTP in use as would be encapsulated in SDP (e.g., "RTP/AVP" or "UDP/TLS/RTP/SAVP"). If not included, the default value of "RTP/AVP" MUST be assumed.

The encodings SHOULD be provided in order of preference by placing the most-preferred

<sup>5</sup>XEP-0177: Jingle Raw UDP Transport Method <<https://xmpp.org/extensions/xep-0177.html>>.

<sup>6</sup>XEP-0176: Jingle ICE-UDP Transport Method <<https://xmpp.org/extensions/xep-0176.html>>.

<payload-type/> element as the first child of the <description/> element (etc.). The allowable attributes of the <payload-type/> element are as follows:

Attribute	Description	Datatype/Units	Inclusion
channels	The number of channels (e.g., 2 for stereoscopic video)	positiveInteger faults to 1)	(de- OPTIONAL
clockrate	The sampling frequency in Hertz	positiveInteger	RECOMMENDED
id	A unique identifier for the payload type	positiveInteger	REQUIRED
name	A name for the payload type	string	RECOMMENDED for static payload types, REQUIRED for dynamic payload types

In Jingle Video, the encodings are used in the context of RTP. The most common encodings for the Audio/Video Profile (AVP) of RTP are listed in [RFC 3551](#)<sup>7</sup> (these "static" types are reserved from payload ID 0 through payload ID 95), although other encodings are allowed (these "dynamic" types use payload IDs 96 to 127) in accordance with the dynamic assignment rules described in Section 3 of RFC 3551. The payload IDs are represented in the 'id' attribute. Each <payload-type/> element MAY contain one or more child elements that specify particular parameters related to the payload. For example, as described in [RTP Payload Format for Theora Encoded Video](#)<sup>8</sup>, the "configuration", "configuration-uri", "delivery-method", "height", "sampling", and "width" parameters may be specified in relation to usage of the Theora<sup>9</sup> codec. Where such parameters are encoded via the "fmp" SDP attribute, they shall be represented in Jingle via the following format:

```
<parameter name='foo' value='bar' />
```

Note: The parameter names are effectively guaranteed to be unique, since the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>10</sup> maintains a registry of SDP parameters (see <<http://www.iana.org/assignments/sdp-parameters>>).

<sup>7</sup>RFC 3551: RTP Profile for Audio and Video Conferences with Minimal Control <<http://tools.ietf.org/html/rfc3551>>.

<sup>8</sup>RTP Payload Format for Theora Encoded Video <<http://tools.ietf.org/html/draft-barbato-avt-rtp-theora>>. Work in progress.

<sup>9</sup>See <<http://www.theora.org/>>.

<sup>10</sup>The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <<http://www.iana.org/>>.

## 5 Negotiating a Jingle Video Session

When the initiator sends a session-initiate stanza to the responder, the <description/> element includes all of the payload types that the initiator can send and/or receive for jingle video, each one encapsulated in a separate <payload-type/> element (the rules specified in RFC 3264<sup>11</sup> SHOULD be followed regarding inclusion of payload types).

Listing 2: Initiation

```
<iq from='romeo@montague.net/orchard'
  to='juliet@capulet.com/balcony'
  id='jinglevideo1'
  type='set'>
  <jingle xmlns='urn:xmpp:tmp:jingle'>
    action='session-initiate'
    initiator='romeo@montague.net/orchard'
    sid='v1d30k11l3dth3r4d10st4r'>
    <content content='initiator' name='this-is-the-video-content'>
      <description xmlns='urn:xmpp:tmp:jingle:apps:video-rtp' profile=
        'RTP/AVP'>
        <payload-type id='96' name='theora' clockrate='90000'>
          <parameter name='height' value='720'/>
          <parameter name='width' value='1280'/>
          <parameter name='delivery-method' value='inline'/>
          <parameter name='configuration' value='somebase16string'/>
          <parameter name='sampling' value='YCbCr-4:2:2'/>
        </payload-type>
        <payload-type id='28' name='nv' clockrate='90000'/>
        <payload-type id='25' name='CelB' clockrate='90000'/>
        <payload-type id='32' name='MPV' clockrate='90000'/>
      </description>
      <transport xmlns='urn:xmpp:tmp:jingle:transports:ice-tcp'/>
    </content>
  </jingle>
</iq>
```

Upon receiving the session-initiate stanza, the responder determines whether it can proceed with the negotiation. The general Jingle error cases are specified in XEP-0166 and illustrated [Jingle RTP Sessions \(XEP-0167\)](#)<sup>12</sup>. In addition, the responder must determine if it supports any of the payload types advertised by the initiator; if it supports none of the offered payload types, it must reject the session by returning a <not-acceptable/> error with a Jingle-Video-specific condition of <unsupported-codecs/>:

Listing 3: Responder does not support any of the codecs

<sup>11</sup>RFC 3264: An Offer/Answer Model with the Session Description Protocol (SDP) <<http://tools.ietf.org/html/rfc3264>>.

<sup>12</sup>XEP-0167: Jingle RTP Sessions <<https://xmpp.org/extensions/xep-0167.html>>.

```

<iq from='juliet@capulet.com/balcony'
  id='jingleaudio1'
  to='romeo@montague.net/orchard'
  type='error'>
  <error type='cancel'>
    <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
    <unsupported-codecs xmlns='urn:xmpp:tmp:jingle:apps:video:errors' />
  </error>
</iq>

```

If there is no error, the responder acknowledges the session initiation request:

Listing 4: Responder acknowledges session-initiate request

```

<iq from='juliet@capulet.com/balcony'
  id='jinglevideo1'
  to='romeo@montague.net/orchard'
  type='result' />

```

If the responder wishes to accept the content definition, it MUST send a content-accept action to the initiator, which SHOULD include a list of the payload types that it can send and/or receive. The list that the responder sends MAY include any payload types (not a subset of the payload types sent by the initiator) but SHOULD retain the ID numbers specified by the initiator. The order of the <payload-type/> elements indicates the responder's preferences, with the most-preferred types first.

Listing 5: Responder accepts content type

```

<iq from='juliet@capulet.com/balcony'
  to='romeo@montague.net/orchard'
  id='jinglevideo2'
  type='set'>
  <jingle xmlns='urn:xmpp:tmp:jingle'>
    action='content-accept'
    initiator='romeo@montague.net/orchard'
    sid='v1d30k11l13dth3r4d10st4r'>
    <content content='initiator' name='this-is-the-video-content'>
      <description xmlns='urn:xmpp:tmp:jingle:apps:video-rtp' profile='RTP/AVP'>
        <payload-type id='96' name='theora' clockrate='90000'>
          <parameter name='height' value='720' />
          <parameter name='width' value='1280' />
          <parameter name='delivery-method' value='inline' />
          <parameter name='configuration' value='somebase16string' />
          <parameter name='sampling' value='YCbCr-4:2:2' />
        </payload-type>
        <payload-type id='32' name='MPV' clockrate='90000' />
      </content>
    </jingle>
  </set>

```



```

    <payload-type id='33' name='MP2T' clockrate='90000' />
  </description>
  <transport xmlns='urn:xmpp:tmp:jingle:transports:ice-tcp' />
</content>
</jingle>
</iq>

```

The initiator acknowledges the 'content-accept' with an empty IQ result:

Listing 6: Initiator acknowledges modified application type

```

<iq from='romeo@montague.net/orchard'
  to='juliet@capulet.com/balcony'
  id='jinglevideo2'
  type='result' />

```

After successful transport negotiation (for the ICE-UDP method, see XEP-0176), the responder then accepts the session:

Listing 7: Responder definitively accepts the session

```

<iq from='juliet@capulet.com/balcony'
  id='accept1'
  to='romeo@montague.net/orchard'
  type='set'>
  <jingle xmlns='urn:xmpp:tmp:jingle'
    action='session-accept'
    initiator='romeo@montague.net/orchard'
    responder='juliet@capulet.com/balcony'
    sid='v1d30k11l3dth3r4d10st4r'>
    <content content='initiator' name='this-is-the-video-content'>
      <description xmlns='urn:xmpp:tmp:jingle:apps:video-rtp' profile='RTP/AVP'>
        <payload-type id='96' name='theora' clockrate='90000'>
          <parameter name='height' value='720' />
          <parameter name='width' value='1280' />
          <parameter name='delivery-method' value='inline' />
          <parameter name='configuration' value='somebase16string' />
          <parameter name='sampling' value='YCbCr-4:2:2' />
        </payload-type>
        <payload-type id='32' name='MPV' clockrate='90000' />
        <payload-type id='33' name='MP2T' clockrate='90000' />
      </description>
      <transport xmlns='urn:xmpp:tmp:jingle:transports:ice-tcp'>
        <candidate component='1'
          foundation='1'
          generation='0'
          ip='192.0.2.3'
          network='1'

```

```

        port='45664'
        priority='1694498815'
        protocol='udp'
        pwd='asd88fgpdd777uzjYhagZg'
        rel-addr='10.0.1.1'
        rel-port='8998'
        rem-addr='192.0.2.1'
        rem-port='3478'
        type='srflx'
        ufrag='8hhy' />
    </transport>
</content>
</jingle>
</iq>

```

And the initiator acknowledges session acceptance:

Listing 8: Initiator acknowledges session acceptance

```

<iq from='romeo@montague.net/orchard'
    to='juliet@capulet.com/balcony'
    id='accept1'
    type='result' />

```

Note: For more examples, see XEP-0167.

## 6 Mapping to Session Description Protocol

The SDP media type for Jingle Video via RTP is "video" (see Section 8.2.1 of RFC 4566). If the payload type is static (payload-type IDs 0 through 95 inclusive), it MUST be mapped to a media field defined in RFC 4566. The generic format for the media field is as follows:

```
m=<media> <port> <transport> <fmt list>
```

In the context of Jingle video sessions, the <media> is "video", the <port> is the preferred port for such communications (which may be determined dynamically), the <transport> is whatever profile is negotiated via the 'profile' attribute of the <content/> element in the Jingle negotiation (e.g., "RTP/AVT"), and the <fmt list> is the payload-type ID.

For example, consider the following static payload-type:

Listing 9: Jingle format for static payload-type

```
<payload-type id="28" name="nv" />
```

That Jingle-formatted information would be mapped to SDP as follows:

Listing 10: SDP mapping of static payload-type

```
m=video 9000 RTP/AVP 28
```

If the payload type is dynamic (payload-type IDs 96 through 127 inclusive), it SHOULD be mapped to an SDP media field plus an SDP attribute field named "rtptime". For example, consider a VC-1 payload such as that described in RFC 4425<sup>13</sup>:

Listing 11: Jingle format for dynamic payload-type

```
<payload-type id='98' name='vc1' />
```

That Jingle-formatted information would be mapped to SDP as follows:

Listing 12: SDP mapping of dynamic payload-type

```
m=video 49170 RTP/AVP 98
a=rtptime:98 vc1/90000
```

As noted, if additional parameters are to be specified, they shall be represented as attributes of the <payload-type/> element or its child <parameter/> element, as in the following example.

Listing 13: Jingle format for dynamic payload-type with parameters

```
<payload-type id='96' name='theora' clockrate='90000'>
  <parameter name='height' value='720' />
  <parameter name='width' value='1280' />
  <parameter name='delivery-method' value='inline' />
  <parameter name='configuration' value='somebase16string' />
  <parameter name='sampling' value='YCbCr-4:2:2' />
</payload-type>
```

That Jingle-formatted information would be mapped to SDP as follows:

Listing 14: SDP mapping of dynamic payload-type with parameters

```
m=video 49170 RTP/AVP 98
a=rtptime:96 theora/90000
a=fmtp:96 sampling=YCbCr-4:2:2; width=1280; height=720;
delivery-method=inline; configuration=somebase16string;
```

## 7 Error Handling

The Jingle-Video-specific error conditions are as follows:

<sup>13</sup>RFC 4425: RTP Payload Format for Video Codec 1 (VC-1) <<http://tools.ietf.org/html/rfc4425>>.

Jingle Video Condition	XMPP Condition	Description
<unsupported-codecs/>	<not-acceptable/>	The recipient does not support any of the offered video encodings.

## 8 Determining Support

If an entity supports Jingle video exchanges via RTP, it MUST advertise that fact by returning a feature of "urn:xmpp:tmp:jingle:apps:video" in response to [Service Discovery \(XEP-0030\)](#)<sup>14</sup> information requests (see [Protocol Namespaces](#) regarding issuance of one or more permanent namespaces).

Listing 15: Service discovery information request

```
<iq from='romeo@montague.net/orchard'
  id='disco1'
  to='juliet@capulet.com/balcony'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 16: Service discovery information response

```
<iq from='juliet@capulet.com/balcony'
  id='disco1'
  to='romeo@montague.net/orchard'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:tmp:jingle' />
    <feature var='urn:xmpp:tmp:jingle:apps:video-rtp' />
    ...
  </query>
</iq>
```

Naturally, support may also be discovered via the dynamic, presence-based profile of service discovery defined in [Entity Capabilities \(XEP-0115\)](#)<sup>15</sup>.

<sup>14</sup>XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

<sup>15</sup>XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

## 9 Informational Messages

Informational messages may be sent by either party within the context of Jingle to communicate the status of a Jingle video session, device, or principal. The informational message MUST be an IQ-set containing a <jingle/> element of type "session-info". No informational message payload elements have yet been defined for Jingle Video via RTP, but they may be specified in a future version of this document.

## 10 Implementation Notes

### 10.1 Codecs

Support for the Theora codec is RECOMMENDED.

## 11 Security Considerations

In order to secure the data stream, implementations SHOULD use encryption methods appropriate to the transport method and media being exchanged; for example, in the case of UDP, that would include Datagram Transport Layer Security (DTLS) as specified in [RFC 4347](#)<sup>16</sup>. [RTP Over DTLS](#)<sup>17</sup> defines such methods for the Session Description Protocol; the relevant RTP profile (e.g., "UDP/TLS/RTP/SAVP" for transporting the RTP stream over DTLS with UDP) shall be specified as the value of the <content/> element's 'profile' attribute.

## 12 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>18</sup>.

## 13 XMPP Registrar Considerations

### 13.1 Protocol Namespaces

Until this specification advances to a status of Draft, its associated namespaces shall be:

---

<sup>16</sup>RFC 4347: Datagram Transport Layer Security <<http://tools.ietf.org/html/rfc4347>>.

<sup>17</sup>Real-Time Transport Protocol (RTP) over Datagram Transport Layer Security (DTLS) <<http://tools.ietf.org/html/draft-fischl-mmusic-sdp-dtls>>. Work in progress.

<sup>18</sup>The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <<http://www.iana.org/>>.

- urn:xmpp:tmp:jingle:apps:video
- urn:xmpp:tmp:jingle:apps:video:errors

Upon advancement of this specification, the [XMPP Registrar](#)<sup>19</sup> shall issue permanent namespaces in accordance with the process defined in Section 4 of [XMPP Registrar Function \(XEP-0053\)](#)<sup>20</sup>.

The following namespaces are requested, and are thought to be unique per the XMPP Registrar's requirements:

- urn:xmpp:jingle:app:video-rtp
- urn:xmpp:jingle:app:video-rtp:errors

## 13.2 Jingle Application Formats

The XMPP Registrar shall include "video-rtp" in its registry of Jingle application formats. The registry submission is as follows:

```
<application>
  <name>video-rtp</name>
  <desc>Jingle sessions that support video exchange via the Real-time
    Transport Protocol</desc>
  <transport>lossy</transport>
  <doc>XEP-0180</doc>
</application>
```

## 14 XML Schemas

### 14.1 Application Format

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:tmp:jingle:apps:video-rtp'
  xmlns='urn:xmpp:tmp:jingle:apps:video-rtp'
  elementFormDefault='qualified'>
  <xs:element name='description'>
    <xs:complexType>
```

<sup>19</sup>The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

<sup>20</sup>XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

```

    <xs:sequence>
      <xs:element ref='payload-type' minOccurs='0' maxOccurs='
        unbounded' />
    </xs:sequence>
    <xs:attribute name='profile' use='required' type='xs:string'
      default='RTP/AVP' />
  </xs:complexType>
</xs:element>

<xs:element name='payload-type'>
  <xs:complexType>
    <xs:sequence minOccurs='0' maxOccurs='unbounded'>
      <xs:element ref='parameter' />
    </xs:sequence>
    <xs:attribute name='channels' type='xs:integer' use='optional'
      default='1' />
    <xs:attribute name='clockrate' type='xs:short' use='optional' />
    <xs:attribute name='id' type='xs:unsignedByte' use='required' />
    <xs:attribute name='name' type='xs:string' use='optional' />
  </xs:complexType>
</xs:element>

<xs:element name='parameter'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='name' type='xs:string' use='required' />
        <xs:attribute name='value' type='xs:string' use='required' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:simpleType name='empty'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='' />
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

## 14.2 Errors

```

<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:tmp:jingle:apps:video:errors'

```

```
xmlns='urn:xmpp:tmp:jingle:apps:video:errors'  
elementFormDefault='qualified'>  
  
<xs:element name='unsupported-codecs' type='empty' />  
  
<xs:simpleType name='empty'>  
  <xs:restriction base='xs:string'>  
    <xs:enumeration value='' />  
  </xs:restriction>  
</xs:simpleType>  
  
</xs:schema>
```