

XCON WG  
Internet-Draft  
Expires: January 17, 2006

C. Jennings  
Cisco Systems  
B. Rosen  
Emergicon  
July 16, 2005

Media Conference Server Control for XCON  
draft-jennings-xcon-media-control-03

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 17, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

Conference servers have many controls that change how the media is combined for the various conference participants. It is necessary to describe these controls to the clients connected to a centralized conference, so that the clients can render a user interface and allow the user to manipulate them.

This work is being discussed on the xcon@ietf.org mailing list. This

Internet-Draft

Media Mixer Control

July 2005

draft has not changed since the 02 version.

## Table of Contents

<a href="#">1.</a>	Conventions . . . . .	<a href="#">3</a>
<a href="#">2.</a>	TODO Items . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Non Problems . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Terminology . . . . .	<a href="#">6</a>
<a href="#">5.1</a>	Templates . . . . .	<a href="#">6</a>
<a href="#">5.2</a>	Controls . . . . .	<a href="#">6</a>
<a href="#">5.3</a>	Parameters . . . . .	<a href="#">7</a>
<a href="#">5.4</a>	Roles . . . . .	<a href="#">8</a>
<a href="#">5.5</a>	Streams . . . . .	<a href="#">9</a>
<a href="#">6.</a>	Examples . . . . .	<a href="#">10</a>
<a href="#">6.1</a>	Simple Audio Example . . . . .	<a href="#">10</a>
<a href="#">6.2</a>	Simple Audio Video Example . . . . .	<a href="#">11</a>
<a href="#">7.</a>	Types of Controls . . . . .	<a href="#">14</a>
<a href="#">7.1</a>	Strings . . . . .	<a href="#">14</a>
<a href="#">7.2</a>	Integer . . . . .	<a href="#">15</a>
<a href="#">7.3</a>	Boolean . . . . .	<a href="#">15</a>
<a href="#">7.4</a>	Selection . . . . .	<a href="#">15</a>
<a href="#">7.5</a>	Multiple Selection . . . . .	<a href="#">16</a>
<a href="#">7.6</a>	Control Array . . . . .	<a href="#">16</a>
<a href="#">7.7</a>	Layout . . . . .	<a href="#">16</a>
<a href="#">7.8</a>	Panel . . . . .	<a href="#">17</a>
<a href="#">8.</a>	Template Registry . . . . .	<a href="#">17</a>
<a href="#">9.</a>	IANA . . . . .	<a href="#">18</a>
<a href="#">10.</a>	Security . . . . .	<a href="#">18</a>
<a href="#">11.</a>	Acknowledgments . . . . .	<a href="#">18</a>
<a href="#">12.</a>	References . . . . .	<a href="#">18</a>
<a href="#">12.1</a>	Normative References . . . . .	<a href="#">18</a>
<a href="#">12.2</a>	Informative References . . . . .	<a href="#">18</a>
	Authors' Addresses . . . . .	<a href="#">19</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">20</a>

Internet-Draft

Media Mixer Control

July 2005

## 1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [4].

## 2. TODO Items

Note - the issue of switching from presenter mode to Q and A mode (etc.) is essentially one of floor control? Need much more on how MPCP and floor control work.

Note - using panel for now - may later replace with media neutral term such as placement

## 3. Introduction

This work tries to solve the problem of how a conference participant should manipulate the media flow in a conference server. It defines a protocol between the centralized conference server and the end user's software that manipulates the conference. This protocol needs to be rich enough for a conference server to express what information it wants, yet simple enough to allow the client to render a useful user interface. This work takes into account that real conference servers have constraints on what media flows are possible and that UIs have buttons, knobs, etc. that users manipulate. The goal is for a conferencing end point made by one vendor to work with conference servers or conference systems made by other vendors.

Someone wishing to create a conference uses CPCP (or some other means) to create a conference and obtain a Conference URI. The conference creator can query the server to find out its media capabilities, information such as the set of templates that a server supports. A template defines a type of conferencing service that a conference server can provide. It includes what media streams can flow in and out of the conference, the roles that are possible in the

conference, and most importantly, what controls a client can manipulate on the conference to affect the media mix. A set of standardized templates that a server may support is defined, and in addition, conference servers that support the flow graphs work in [TODO REF](#) can dynamically define new templates. Note that templates contain media specific information, so to know which templates are supported is also to know what media types are supported. Each template lists a number of parameters that must be set to initialize the conference and can have limits imposed by the conference server. Parameters are typically maximum values that are hardware or software (or policy) hard limits that constrain what is possible in a conference. Parameters can only be set when the conference is

instantiated and can not be changed after that. For things that are changed as the conference progresses, controls are used. The point of the parameters in the templates is simply to reduce the number of templates needed.

The conference creator can then choose a template, populate the parameter values and upload using CPCP to the server. If the chosen parameter values are acceptable to the server, the update is accepted and the media policy created. If not, an error message indicating the failure is returned.

The simplest template will have just a single role: Participant. By default, each participant will join a conference as a Participant. More interesting templates will have multiple roles. For example, a template might have two roles: Lecturer and Participant. A template role definition will indicate if there can be more than one participant having that role. For this example, there can be only one Lecturer but many Participants.

The conference creator can assign roles to participants. This can be done in advance of the conference or dynamically during the conference. For example, the conference creator can assign the role of Lecturer in advance, if it is known. When this participant joins the conference, they will be automatically assigned the role of Lecturer. This is Conference Policy not Media Policy but it does relate to the templates. The conference package [TODO REF](#) includes the Role of each participant in the conference.

Once a conference starts, a participant can find out the media policy

template. They can also download the set of controls for each role they may assume during the conference. This template may also have controls that allow a participant to control their view of/input to the conference. These controls may be rendered to the participant, and any changes to the controls result in commands being sent to the conference server. A template may define different controls for different roles. For example, a Participant may have only a very small set of controls, a Lecturer a larger set, and the Floor Holder an even larger set. If a participant's role changes during a conference, their set of controls may change, and the user interface needs to be updated accordingly.

An advanced conference server may support the definition of custom templates using flow graphs. If so, the conference policy will indicate this capability. If it is supported, a conference creator may upload a flow graph using CPCP. This flow graph will contain enough information for the conference server to create a custom template: it will contain stream level media mixing information and information about parameters, roles, controls, and support for floor

control. If the server can process the flow graph and support the mixing defined by the template, the server returns a success response. If it is not able, it returns an error indicating how the flow graph might be fixed. A custom template created using flow graphs will be identical to the set of standardized templates - it will just have a different name, roles, parameters, controls, etc. The same methods that allow a participant to render an unknown standardized template will be used to render a custom template.

Once a conference begins, the template and parameters are fixed and MAY NOT be manipulated during the conference. As a result, flow graphs can only be uploaded prior to the start of the conference, although they could be downloaded by a participant during a conference using CPCP. In general, however, flow graphs will only be used by the creator of the conference prior to the start of the conference.

A conference client can request the conference object from the focus. This allows the client to discover what the current media policy is and what controls it can manipulate. The client can then send an update to the focus to change the controls to manipulate media policy for various participants.

The conference has a set of physical streams that get contributed to the conference and a set of streams that are sent to the client. The streams coming into the conference feed into an input stream group, and the streams coming out of a conference come from an output stream group.

A template may define various logical stream lists. For example, one video stream may contain video of the active presenter, and another video stream may have the presentation that the presenter is showing. Media from a participant is contributed to one of the input stream groups. Various controls, such as gain, may be attached to each physical input stream, to logical stream groups, or to a top-level conference or sidebar.

Each conference also has output stream groups that represent media being sent to the client. Output streams to a client are named and may have complex controls that affect which streams are selected to contribute to the result. Output streams may be formed using multiple input component streams. This is typically done for video when the output is some composited form of the input component streams, but it can also be done for audio, e.g. when selecting multiple mono audio streams and defining how they are composited into a stereo stream.

#### [4.](#) Non Problems

There are several topics that are completely internal to the conference systems and are out of scope of this work. These include:

- how the focus manipulates the conference server
- how one describes what a conference server is capable of doing;
- and
- managing resource allocation and how busy a given DSP is, and checking whether more work can be allocated to a media processor.

#### [5.](#) Terminology

##### [5.1](#) Templates

TODO - one template instantiated per conference. Changing a template

is close to stopping a conference and starting a new one.

A template defines a model for the reception, manipulation and transmission of streams. A template provides enough information that the client can intelligently render a useful GUI to the end user to manipulate the model. There is a registry of well known templates, but a conference server can define new ones. A convener can find all the templates a conference server supports and select one to use when creating the conference.

Templates contain a list of logical stream, input and output stream, roles for participants, and controls for the conference.

A template for a very basic audio conference, for example, may indicate that there is one audio stream for each participant, and one output stream group named "main". Each participant in the stream has a single binary control called "Mute". There is only one Role that can be used, called "participant".

## [5.2](#) Controls

Controls are variables in a conference object that participants may manipulate to control the media streams of the conference. The Control has information about what type of inputs it accepts that help the client render a user interface. Conferences can have controls, participants in a conference can have controls, and streams in a conference can have controls. A control has a name, a value, and constraints on its value. The controls that are available are defined in the template.

A control can be defined as being part of a role. In that case, all participants who assume that role have an instance of the control. A control may also be defined as part of an input stream group, in

which case all contributors of that stream will have an instance of the control; or an output stream, in which case each output stream will have an instance of the control. There can be global controls that change values for the whole conference.

A control can be inside the template, participant, or stream group. The control will apply to the appropriate context. By including stream definitions in multiple roles that have the same name,

different controls can be provided to different roles affecting streams contributed or sunk from multiple roles. For example, a moderator may be given a set of input volume controls controlling a mix, and every participant can be given an output master mix control for the output stream sent to him.

### [5.3](#) Parameters

TODO - need better name for Parameters. Perhaps Instantiation Values

Parameters are variables in the template that are set when the conference is created. The point of a Parameter is simply to reduce the number of templates required. For example, in the audio conference, whether or not sidebars are supported might be a parameter. The template can indicate the valid range for parameters. Parameters can also be used for an application instantiating a conference to limit what capabilities it will use.

Parameters are variables that modify the function of the template. They are fixed when the conference object is instantiated and can not be changed after that. Parameters allow a single template definition to describe a range of possible conference server capabilities.

Parameters have a name, a type, a value and, optionally, a min and max value.

The parameters in the templates customize a generic template for a specific conference. Parameters have name, type, value, and optionally min/max. Parameters are defined in the template description. Only conveners can set template parameters.

One typical template parameter is "max-sidebars". When the CS generates the template for the client, it can customize the min and max value of this parameter to match what it is capable of, which might range from zero or one to infinite. When the client instantiates the template and creates the conference, it can specify the value that has been requested. The value typically represents the limits the conference server is capable of. Resource availability may limit the actual value that can be achieved.

Parameter names are strings.



Parameter Types:

- Integer

- Real

- Enumeration

Values of course are constrained to the type. Min and Max, if defined, also constrain the the value.

TODO - need to be able to make the limits of controls be parameters.

#### [5.4](#) Roles

Participants in a conference can take on multiple different Roles that will change what controls they may manipulate and which media streams they have access to. The template defines what Roles are available for the client. Manipulation of Roles is not directly part of MPCP, but the various Roles that are possible are found in the template. Some common roles include:

- Participant

- Presenter

- Moderator

- Observer

OPEN ISSUE - decide if we want Role so a single participant can simultaneously have multiple roles

Roles are defined as part of Conference Policy but are used here so that the Media Policy can define separate streams and controls depending on role. Roles are defined by in the template. Some templates may allow a participant to take on more than one role at a time. Each template must define a role named "Participant", which is the default role. "Moderator" is a typical role, but templates do not intrinsically define or require such roles. A given user will only be able to access parts of the template that are not inside a Role or are inside a Role that the this user is a member of.

Templates define all the Roles that a participant can take and (optionally) the max number of participants of each role. Each role is defined in a role element. A Role element includes a name and optionally a "max-participants" value. Role elements may also contain stream elements, which define per-participant-in-role streams. The first stream list of a given media type inside a Role is the default location for that type of media.

## [5.5](#) Streams

Streams correspond to a given flow of media. They are named and can be selected by controls. The conference package is used to understand the relationships between users or participants, dialog or session, and streams.

The physical streams are the actual media streams sent and/or received by or on behalf of conference participants. Media streams are typically established when conference participants join a conference and are described by the SDP media lines in the offer/answer exchange between the participants and the focus, or the analogous exchange in other protocols (ex: H.245 logical channel establishment). Each stream is described by a media type, direction and at least one identifier. Initially media types considered include audio, video or text. (Other media types can also be considered in the future.) The direction "in" corresponds to streams originating from the conference participants to the conference, and "out" for streams originating from the conference and terminating at the conference participants. A stream-id is an integer assigned by the focus to each physical input and output stream. This integer is unique to all streams in a specific conference (and all its sub-conferences).

Logical streams are names that are defined in the template and can be used like other streams but correspond to some virtual stream that the conference is creating. Logical streams often change dynamically and potentially very quickly during the lifetime of a conference. For example, one logical set is the set of input video streams corresponding to the current speaker or speakers. Logical stream lists are discussed in more detail in the following section.

Streams have types. These correspond to the major MIME types of the media the stream carries.

Audio Streams originate as participant contributions (dir is "in") that are mixed using some kind of algorithm. Controls commonly available on audio streams include input or output faders (volume controls), stereo balance, and mute.

Video Streams originate as participant contributions (dir is "in"), which are combined with some kind of algorithm. Intermediate streams may be created, which are subsequently combined with other streams to yield streams that are sent to participants (dir is "out"). Controls commonly available on video streams might include selectors for choosing a tiling format, selectors that choose which input stream is rendered on an output tile, and video

freeze and blank.

Internet-Draft

Media Mixer Control

July 2005

Text Streams originate as participant contributions (dir is "in") (Instant Messages). Messages from all participants are combined using some algorithm. Intermediate streams may be created, which are subsequently combined with other text streams to yield streams that are sent to participants (dir is "out").

The stream id correlates the stream with a particular RTP session or media session for non RTP based media. The client can learn the correlation of stream ID to the particular media streams it is sending by TBD (TODO could be subscribing to the conference package).

A stream-id is an integer assigned by the focus to each physical input and output stream. For RTP media, this corresponds to a single RTP session. This integer is unique to all streams in a specific conference (and all its sub-conferences).

## [6.](#) Examples

### [6.1](#) Simple Audio Example

The examples in this section will all be moved to XML, but to help make them easier to understand and focus on the semantics instead of the syntax, they are currently just some text with indentation representing containment.

The client selects the basic audio template that looks like:

Template BasicAudio

```
PhysicalStream direction=input type=audio name=main-audio-in
  control type=bool name=mute label="Mute"
```

```
PhysicalStream direction=output type=audio name=main-audio-out
  control type=real name=gain label="Volume"
```

This templates defines that this conference has one input stream group called main-audio and one output stream group called main-audio. There is a single control, called mute, for each physical input stream, and a gain for each output stream.

After Alice and Bob have joined, the conference server informs Bob that the current state of the conference object is as shown in the xml below.

Conference BasicAudio

PhysicalStream name=main-audio-in stream-id=1  
control bool mute=false

PhysicalStream name=main-audio-out stream-id=3  
control bool mute=false

PhysicalStream name=main-audio-out stream-id=2  
control real gain=0.0

PhysicalStream name=main-audio-out stream-id=4  
control real gain=0.0

There are two participants, Alice and Bob, who both contribute input streams and receive output streams, and neither is muted.

A key part of this is that Bob's client may have known about this basic audio template and what the semantics of the "mute" control implied. The client may have connected this up with a button of the client's that was labeled mute. On the other hand, Bob's client may not have known anything about this template and simply rendered a button on the screen and labeled it "mute" with no idea what this would do. A third client may not have been able to deal with the control at all and may have just ignored it. Clearly the user interface can be better if the client understands the semantics of what the template means, but the user interface is still functional when the client does not.

## [6.2](#) Simple Audio Video Example

A more complex video example is given below.

Template basicAudioVideo

LogicalStream type=video name=activeSpeaker-video

LogicalStream type=video name=presenter-video

LogicalStream type=video name=presentation-video

Floor name=presenter-floor

Control type=bool name=eCan label="Echo Cancelation"

Role name=listener

PhysicalStream direction=output type=audio name=main-audio-out

PhysicalStream direction=output type=video name=main-video-out

Role name=participant

PhysicalStream direction=input type=audio name=main-audio-in

Control type=bool name=mute label="Mute"

PhysicalStream direction=input type=video name=main-video-in

Control type=choice name=video-mute

choices="normal,blank,freeze"

PhysicalStream direction=input type=video name=presentation

Control type=choice name=video-mute

choices=" normal, blank, freeze"

PhysicalStream direction=output type=audio name=main-audio-out

Control type=real name=gain label="Volume"

PhysicalStream direction=output type=video name=main-video-out

Control name=main-laoyout type=layout choices=1x1,2x2,1x2,

Sidebar

Control name=mainConfVolume type=real

PhysicalStream direction=input type=audio name=side-audio-in  
control type=bool name=mute label="Mute"

PhysicalStream direction=output type=audio name=side-audio-out  
control type=real name=gain label="Volume"

Role name=moderator

controllArray index=main-audio

control type=bool name=mute label="Mute"

This template has some logical streams that can be used for selecting in the layout. It defines a control called eCan that applies to the whole conference. It defines a Listener role that can only receive input and a Participant role. There is also a moderator role that has everything a participant has along with an additional set of controls to mute any of the contributors to the main-audio. The

participants share a single layout control that defines the video layout. The conference supports sidebars but they can only have audio media.

The instantiated value of the conference object for this might look like

ConferenceObject type=basicAudioVideo name=conf1234

Role Participant

PhysicalStream name=main-audio-in stream-id=22  
Control name=mute value=false

PhysicalStream name=main-video-in stream-id=23  
Control name=video-mute value=normal

PhysicalStream name=main-audio-out stream-id=24  
Control name=gain value=0

PhysicalStream name=main-video-out stream-id=25

```
PhysicalStream name=main-audio-in  stream-id=32
  Control name=mute value=false
PhysicalStream name=main-video-in  stream-id=33
  Control name=video-mute value=normal
PhysicalStream name=main-audio-out stream-id=34
  Control name=gain value=0
PhysicalStream name=main-video-out stream-id=35
```

```
PhysicalStream name=main-audio-in  stream-id=42
  Control name=mute value=false
PhysicalStream name=main-video-in  stream-id=43
  Control name=video-mute value=normal
PhysicalStream name=main-audio-out stream-id=44
  Control name=gain value=0
PhysicalStream name=main-video-out stream-id=45
```

```
Control name=main-layout value="3x1+16"
  Pannel input=presenter-video postion=1 selectionQ=0.8 exGrp=1
  Pannel input=presentation   postion=2 selectionQ=1.0 exGrp=1
  Pannel input=active-speaker postion=3 selectionQ=0.9 exGrp=1
  Pannel input=main-video-in  postion=4 selectionQ=0.5 exGrp=2
```

In the example above three participants have joined. All the video participants are watching the same video composition, which has the presenter in position 1, the presentation in position 2, the active speaker in position 3; followed by all the video from participants contributed to the main input group. The streams that are in the positions after the first three are not in the same exclusivity group as the first three, so streams could repeat even if they were being shown in the first three positions. The active speaker position 3 is

in the same exclusion group as the presenter in position 1, so if the primary active speaker was the presenter, this would not be shown in position 3 and instead the previous active speaker would be shown.

## [7.](#) Types of Controls

Controls need to collect information of several different types. It should be possible to provide default values, a name for the control and text it displays, help text, control if a value is required, and control of whether or not the value is editable. It should be possible to express constraints on the form an input can take by

specifying a minimum or maximum for types where that makes sense, or specifying a regular expression that must be satisfied. For numeric values in a constrained range, it should be possible to provide an increment value used by the control. For strings it should be possible to indicate that they should not be displayed when they are entered for things like passwords. These controls are necessary to make it possible to internationalize any text that is displayed to the user.

There are control types for:

- Strings
- Multi-line Strings
- Integer
- Real
- Boolean
- Date
- Time
- Date Time
- URI
- Select Single
- Select Multiple
- Select Stream - TODO ADD THIS
- Layout - video layout object
- Panel - a portion of a Layout

If an unknown control is encountered, it should be treated as a string type. The `<label>` element controls what is displayed to the user and the `<value>` element contains the current setting of the control. If set in the template definition, it represents the default value. An optional `<description>` element provides some text that can be used as help text for the control.

## [7.1](#) Strings

This is typically rendered as a text input field.

```
<control type="string" name="Host" private="true" >  
  <label> Meeting Host </label>  
  <value>Richard</value>  
  <description>Host for this weeks meeting</description>
```



```
<regex>.*[rR].*</regex>
</control>
```

The "private" attribute indicates that the string should not be displayed as it is entered.

## [7.2](#) Integer

This can be rendered as a slider or volume knob if it has a constrained range; otherwise it is a text field. The text field may have increment or decrement buttons.

```
<control type="integer" name="gain">
  <label> Volume </label>
  <value>0</value>
  <range min="-18" max="6" increment="3"/>
</control>
```

## [7.3](#) Boolean

This is typically rendered as a toggle button.

```
<control type="boolean" name="mute">
  <label> Mute </label>
  <value>True</value>
</control>
```

## [7.4](#) Selection

This is typically rendered as a pull down menu or as a radio button box.

```
<control type="select1" name="foo">
  <label> the thing </label>
  <value>2</value>
  <item>
    <label>one</label>
    <value>1</value>
  </item>
  <item>
    <label>two</label>
    <value>2</value>
  </item>
</control>
```

The list of items that can be selected is contained in `<item>` elements. Each item has a label that is displayed and a value that is returned when it is selected.

### [7.5](#) Multiple Selection

This is typically rendered as a combo box or list.

This is the same as a selection, except that the type is selected and the initial value is a space-separated list of values.

### [7.6](#) Control Array

The control array is a way of defining a logical group with a variable number of related controls. When the control array is defined in the template, it is associated with some group of streams. Then in the actual conference object, there will be one instance of the control for each of the actual streams in the stream group. Inside the control array in the template is a single control. A typical use of this would be to form a separate mute control for each of the audio stream correctly contributing to a conference.

### [7.7](#) Layout

Layouts provide a control for compositing multiple video streams. A layout uses several windows that each contain one video stream. Each of these individual windows is referred to as a panel. Generally the windows are rectangles arranged in some grid, but they do not have to be this way if the conference service has some other way of laying them out, such as warping them so they look like talking heads arranged in a circle.

A layout control allows the selection of an enumerated type of layout. This might include things like "2x1", "2x2", "5+1", "auto",

"vendor X special layout 10". Some of the layouts are well

Internet-Draft

Media Mixer Control

July 2005

understood, some may be customized. There are also a number of panels that are being selected into this layout. The layout will support some compositing of a number of streams, like 4 for a 2x2 layout. However, there may be more than 4 streams competing to get shown. The first thing the layout group does is select which 4 streams will get shown. It then figures out where to display these streams in the layout. Each panel inside the layout has a selection Q value. This is used to order the streams to figure out if they will be selected or not. Q values range from 0.0 to 1.0 and higher values are selected over lower values. In addition, panels can be grouped into exclusion groups. If an exclusion group has the same stream more than once, the type of the exclusion group will determine whether the stream is displayed once or twice. If the exclusion group is set at no repeat, the repeat stream with the highest Q value is selected and the rest are removed from the selection set.

Once the set of selected streams is formed, they need to be put in the appropriate place in the layout. Each "window" in the layout has a position number. Generally these increment left to right and then top to bottom. If a panel sets its position attribute, then it gets that location in the layout. For panels where the position is not set, the selected streams are sorted, with a the primary key of the position Q value and a secondary key of the selection Q value then placed into the remaining positions in this order. The streams may be further ordered to minimize streams moving their location in the rendered output as the conference server sees fit.

TODO - need to work back in way of dealing with exclusivity groups. Suggest using a exclusivity set value for each panel so that if two things have the same set value, they will not both display the same image.

## [7.8](#) Panel

Panels are a control to select some streams to display in a layout. They are a control that can select any stream or list of streams in the conference, be they logical or physical. They can select a specific physical stream using the stream id. The client can correlate this with participants using the conference package. They can select a stream out of a logical group such as "active-speaker".

## 8. Template Registry

An IANA registry will be created for commonly encountered template definitions. This document will include some starter templates

[Still need TODO this].

Jennings & Rosen

Expires January 17, 2006

[Page 17]

---

Internet-Draft

Media Mixer Control

July 2005

## 9. IANA

TODO - will need template registry

## 10. Security

TODO

## 11. Acknowledgments

Many thanks to Rohan Mahy, Nermeen Ishmail, Alan Johnson, Orit Levin, Roni Evin, and Lyndsay Campbell for many helpful comments. Rohan in particular has spent a huge amount of time improving this document. Large chunks of text were contributed by Alan Johnson.

## 12. References

### 12.1 Normative References

- [1] Even, R., "Conferencing media policy requirements", [draft-even-xcon-media-policy-requirements-00](#) (work in progress), June 2003.
- [2] Boulton, C. and U. Chandra, "Media Policy Templates for XCON", [draft-boulton-xcon-media-template-00](#) (work in progress), October 2004.
- [3] Barnes, M. and C. Boulton, "A Framework and Data Model for Centralized Conferencing", [draft-barnes-xcon-framework-01](#) (work in progress), December 2004.
- [4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

## 12.2 Informative References

- [5] Mahy, R. and N. Ismail, "Media Policy Manipulation in the Conference Policy Control Protocol",  
[draft-mahy-xcon-media-policy-control-00](#) (work in progress),  
June 2003.
- [6] Even, R., "Conferencing Scenarios",  
[draft-even-xcon-conference-scenarios-00](#) (work in progress),  
June 2003.
- [7] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol",  
[draft-ietf-sipping-conferencing-framework-00](#) (work in progress),

Jennings & Rosen

Expires January 17, 2006

[Page 18]

---

Internet-Draft

Media Mixer Control

July 2005

May 2003.

### Authors' Addresses

Cullen Jennings  
Cisco Systems  
170 West Tasman Drive  
Mailstop SJC-21/2  
San Jose, CA 95134  
USA

Phone: +1 408 421 9990  
Email: fluffy@cisco.com

Brian Rosen  
Emergicon  
470 Conrad Dr.  
Mars, PA 16046  
USA

Phone: +1 724 382 1051  
Email: br@brianrosen.net

#### Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any

copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.