BROADCASTER SETUP AND TROUBLESHOOTING GUIDE

VERSION 1.0

© 2010 RayV Technologies Ltd. All Rights Reserved.

RayV™ is a trademark of RayV Inc.

TABLE OF CONTENTS

1

INTRODUCTION

RayV Broadcaster is a software application responsible for acquiring, encoding, and streaming video/audio content from a customer's broadcast site into RayV's grid.

This guide explains the terminology used in RayV Broadcaster and guides you through the process of configuring the settings and solving troubleshooting problems.

The following topics are covered in this guide:

- Audio/Video Terminology
- Verifying CPU Usage and Buffer Rates
- **Error! Reference source not found.**
- Configuring Timeshifting
- Configuring VOD Recordings
- Making the Broadcaster Less CPU Intensive
- Configuring Uplinks
- Troubleshooting

2

AUDIO/VIDEO TERMINOLOGY

The following terminology is used in the audio/video industry. Review these terms before proceeding.

BITMAPS

A bitmap (.bmp) is a table of pixel values, representing the amount of color to apply at each location of an image. A bitmap representation of a picture is substantially larger than a JPEG representation of the same picture. For instance, the 1024x768 JPEG of the picture shown below is 269KB. The same picture represented as a bitmap with the same resolution is 2.25MB, an increase in size of 834%.

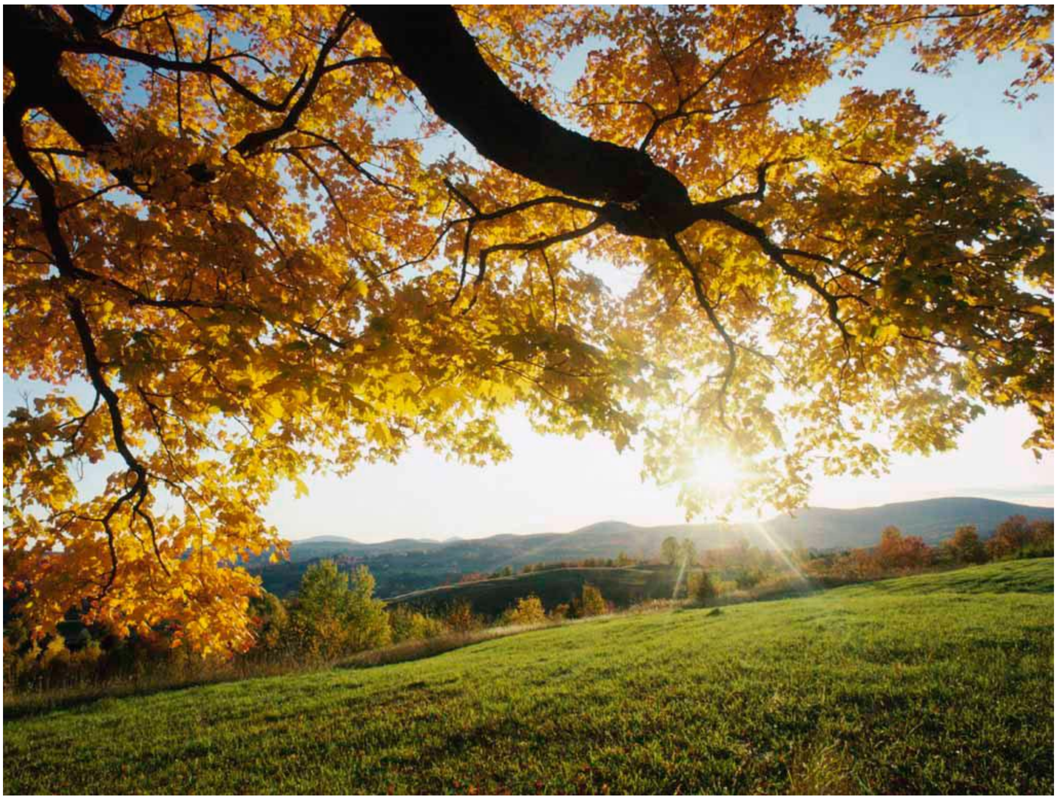

Figure 1: Example Picture

Raw (Digital) Video

A raw (digital) video is a series of raw pictures where each picture in the series has a given duration, which is usually the same for every picture.

In motion pictures, there are 24 pictures per second, or 41.6 milliseconds per frame.

Raw (Digital) Audio

Raw (digital) audio is a series of volume values, also referred to as *samples*.

Digital audio is captured by measuring the audio at a given *sampling rate,* usually 44100 times per second, or 44.1 kHz. In other words, each sample determines the audio at a duration of $\frac{1000}{44100} = 0.0226ms$ per sample.

A sampling rate of 48 kHz is often used in CDs.

Encoding and Compressing

Encoding and compressing is the process of converting raw audio or video, into a (considerably) more compact representation, using a *codec*.

This process is similar to zipping a document in order to reduce its size with the following reservations:

- **Lossless vs. Lossy Compression**

  By zipping and then unzipping a document, the integrity of the original document is maintained without any changes. This process is known as **lossless** compression.

  On the other hand, by encoding and decoding video, the quality of the original video is compromised. The video is visually similar to the original video, but there is a loss of data. This process is known as **lossy** compression.

- **File Size**

  Encoding a video file produces a copy of the file that is larger than the original video.

- **CPU Intensity**

  The process of encoding a video can be very CPU intensive. In the RayV Broadcaster, the video encoding accounts for ~90% of the CPU used by the Broadcaster.

Buffers or Packets

Buffers, or packets, refer to a single atomic unit of information.

In videos, this is often a single picture, which may also be referred to as a frame.

In audio, this may refer to any chunk of consecutive audio data, which may be as small as 10ms or as large as 1000ms.

Buffers, or packets, may refer to either raw or encoded media.

A buffer of encoded audio, usually contains 1024 samples. At 44100 samples per second, that amounts to $\frac{44100}{1024} = 43.06$ buffers per second (refer to Figure 2),

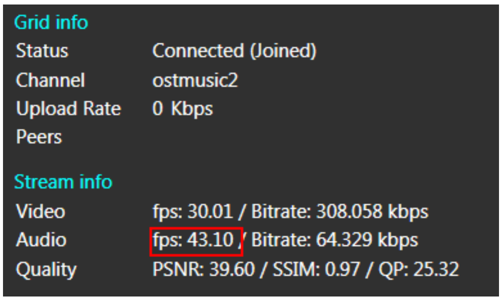

Figure 2: Buffer of Encoded Audio

Frames Per Second (FPS) or Buffer Rate

Frames per second or a buffer rate is a metric that counts the number of *frames*, *buffers*, or *packets*, per second.

For audio:

- The *sampling rate* (44.1 kHz) may be thought of as an FPS, where *sample=frame*.
- Alternatively, you may want to consider the *buffer rate*, or the rate at which a whole packet of audio data is processed.

For video

- FPS refers to whole *frames*, or pictures per second.
- The frames may be raw or encoded.

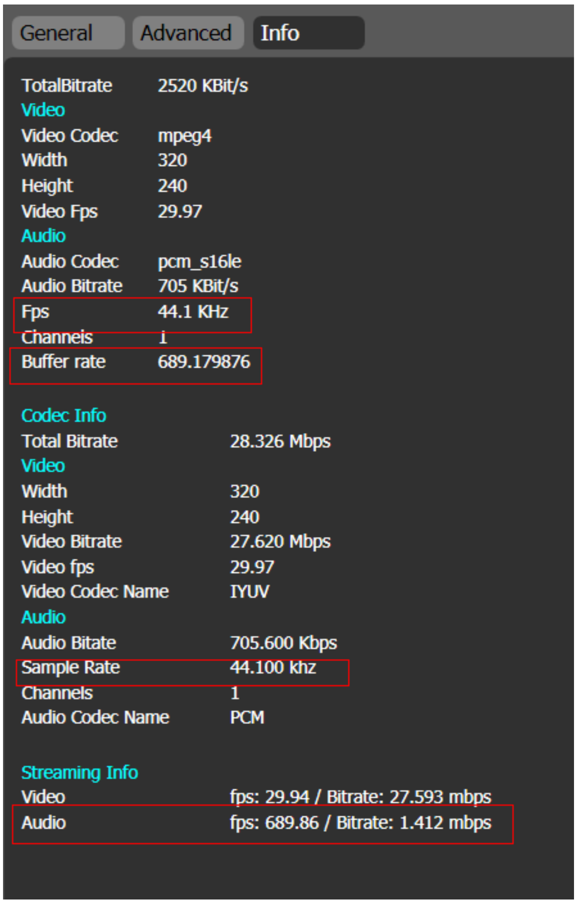

Figure 3: FPS or Buffer Rates

BITRATE

Bitrate is a metric that measures the number of bits processed every second.

Bitrate may refer to the number of bits processed in a raw or encoded stream, or to the number of bits uploaded or downloaded.

For audio:

- Conventional bitrates for music audio are 128kbps, 160kbps, and 192kbps.
- Uncompressed audio stored on an audio-CD has a bit rate of 1,411.2 kbit/s so the bitrates 128, 160, and 192 kbit/s represent compression ratios of approximately 11:1, 9:1, and 7:1 respectively.

For video:

- An uncompressed PAL bitrate (analog cables) is
  8 bits per color @ 720 x 576 @ 25fps = 248,832kbps.

- Conventional bitrates with the H.264 codec for PAL video is between 1000kbps and 3000kbps. These rates represent compression ratios of approximately 248:1 and 83:1 respectively.

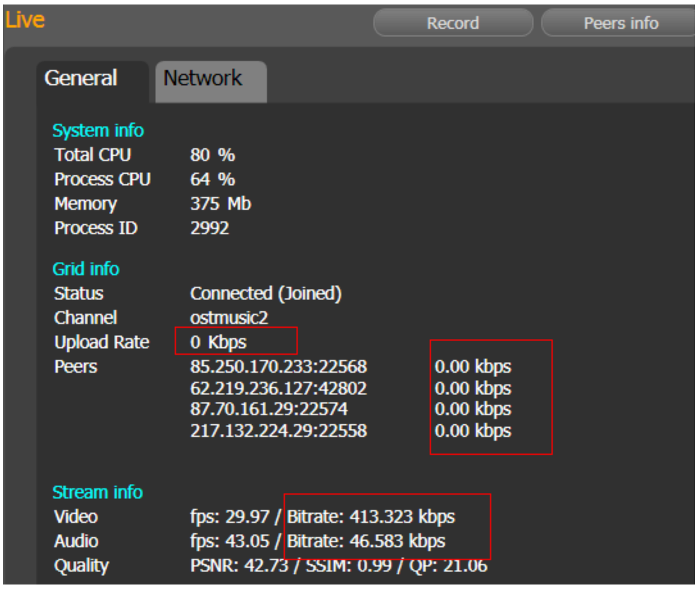

Figure 4: Bitrates

CODEC

A *codec* (short for ***Co**der/**Dec**oder)* is a defined way to express media (audio/video) in an encoded form.

Usually the *decoder* is completely specified, so that there cannot be any innovation in the decoder part.

The encoder, on the other hand, is more complicated, and encoder design is as much an art as a science.

Encoding requires considerably more computing resources than decoding, because encoding is only done once, while decoding is necessary every time the video is watched.

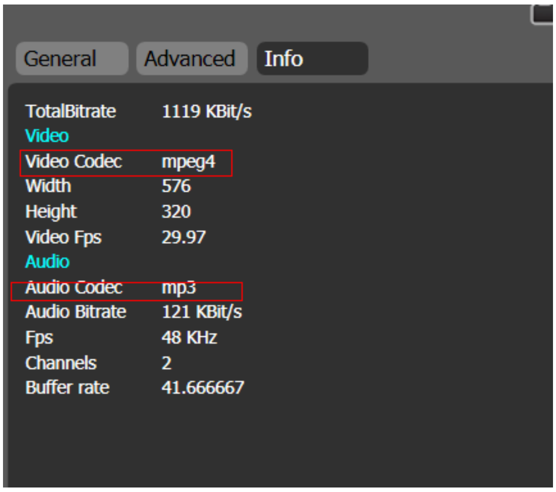

Figure 5: Codec

QUALITY

When copying an audio cassette in a double-cassette deck, quality either remains the same or decreases. This is known as generation loss.

Digital media solved this problem, because digital data can be copied and reproduced *losslessly*, while analog data transfer is always *lossy*.

While (digital) media compression is a *lossy* process, the loss occurs only during encoding. The same video is reproduced every time the video is decoded.

AUDIO

Maximum quality is called *transparency*, which means that it is unable to discern between the original audio and the reproduced audio.

Usually an MP3 encoding of a CD at 320kbps is considered to be *transparent*, even though it requires ~4.5 times less space.

L*ossless* audio codecs (such as APE, FLAC, etc.) is usually compressed to half the bitrate of *raw* audio.

RayV uses the AAC codec at either 64kbps or 96kbps. RayV can use these low bitrates because:

- AAC is a better codec than MP3, offering better quality at the same bitrate.
- RayV's audio channels are voice channels, not music. Voice can be *transparently* encoded at lower bitrates than music.

VIDEO

Changes in compressed video, compared to the original, are called artifacts

These artifacts include blockiness or macroblocking, loss of detail and texture, blurriness, and other effects. Figure 6 is the original image. Figure 7 shows the same image after it has been compress. This figure displays a loss of edge clarity and tone. The fuzziness is a result of a low bitrate compression.

Figure 6: Original Image

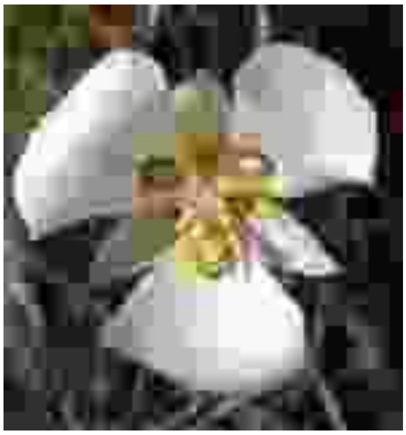

Figure 7: Compressed Image

Figure 8 is an example of an image with macroblocking due to a transmission error.

Figure 8: Example of Macroblocking

Figure 9 is an example of macroblocking in a JPEG. The error appears as a low resolution problem. The highlighted portions clearly highlight the distorted visible edges rather than showing all the errors on their true scale.

Figure 9: Example of Microblocking in a JPEG

With higher compression bitrates, the artifacts become less and less accentuated until quality reaches a point known as visually lossless, where it is impossible to discern between the original and compressed video with the naked eye.

Video quality can be objectively measured with metrics such as PSNR and SSIM. However, these metrics should only serve as a tool. You should also personally inspect the video

Encoding/Media Quality vs Network/P2P Quality vs Viewer Experience

Encoding or media quality refers to loss due to media compression. Only the broadcaster is aware of this quality.

Network or P2P quality refers to the amount of data received in comparison with the amount of data expected.

Viewer experience refers to what a viewer experiences when using the RayV Player to watch a channel. It combines the effect of loss from encoding and artifacts with the effects of not receiving all the data over the network.

When viewer experience deteriorates, it is difficult to know if the problem is in the broadcaster (encoding quality) or the grid (network quality).

Tools such as RayV Probe, RayV RVCM, the Viewer's P2P Info page, and the Broadcaster's interface can help identify the source of the problem. Figure 10 displays an example of objective measurements of video encoding quality in the RayV Broadcaster.

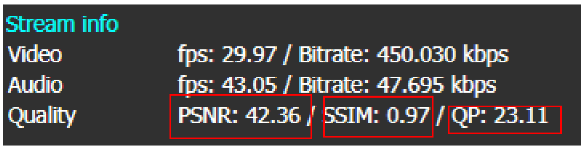

Figure 10: Video Encoding Quality

Aspect Ratio

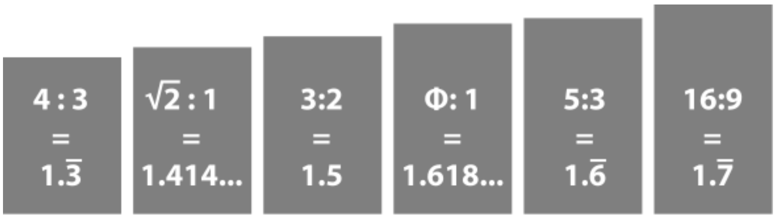

Figure 11: Examples of Aspect Ratio

Aspect ratio describes the ratio between an image’s width and height.

The difference between the aspect ratio that is implied by the resolution and the desirable aspect ratio is called anamorphic video. For example, a wide 16:9 movie may be captured with an SD capture card with a resolution of 4:3 640x480.
The Broadcaster assumes that the content is in 4:3 aspect ratio, resulting in video that will appear squeezed (i.e., the figures appear too thin and long).

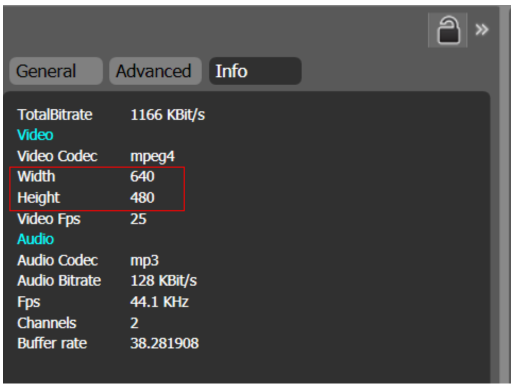

Figure 12: Width and Height Resolution Displayed in the Broadcaster

We can force the Broadcaster to treat the aspect ratio of the source as 16:9, stretching the video back to its natural proportions.

If we know the original content should have a wide 16:9 aspect ratio, we can correct the source's aspect ratio to 16:9.

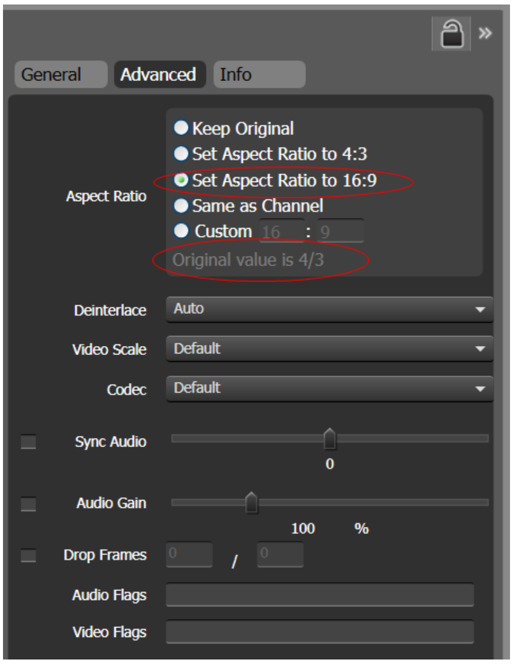

Figure 13: 640x480 Resolution (4:3 Aspect Ratio)

If shooting in widescreen picture format on 4-perf film, without an anamorphic lens, the available film area is not used completely, and some of the film surface is wasted on the frame lines.

Figure 14: No Anamorphic Lens

With an anamorphic lens, the picture is optically squeezed in the horizontal dimension to cover the entire film frame, resulting in better picture quality. When projecting the film, the projector must use a complementary lens of the same anamorphic power to stretch the image horizontally back to its original proportions.

Figure 15: Using an Anamorphic Lens

3

VERIFYING CPU USAGE AND BUFFER RATES

- Verify that the Broadcaster does not use more than 80% of the CPU.
- The Broadcaster should always output audio at ~43 FPS. If not, restart the Broadcaster.
- The Broadcaster should always output video at a **fixed rate**:
  - ~29.97 FPS for American TV
  - ~25 FPS for European TV
  - ~60 FPS for 720P HD

  If the FPS changes all the time, restart the Broadcaster or reduce the CPU usage.

4

CONFIGURING TIMESHIFTING

In Broadcaster 1.0 and 1.1, the recording feature supports timeshifting with the following featurs:

- The recording automatically changes files every 15 minutes.
- Click the Recording button to start recording. The recording continues until you stop it.

When working with timeshifting, you need to open two instances of the Broadcaster. One instance is the *live* Broadcaster and the other instance is the *timeshifting* Broadcaster. The following provides information about each type of Broadcaster:

- **The *live* Broadcaster:**
  - In the *live* Broadcaster, live content (usually through the capture card) is acquired and recorded.
  - The channel does not have to be available for viewing by RayV customers.
  - The *live* Broadcaster must be on air for the recording to operate, and a connection to the satellites and monitoring is highly recommended.
  - The *live* Broadcaster should operate in *raw mode*. (The *live* Broadcaster should never operate in *compressed mode*.)
  - Note the location where recorded files are located. You will need this information when configuring timeshifting
- **The *timeshifting* Broadcaster:**
  - Open a second instance of the Broadcaster, known as the *timeshifting* Broadcaster.
  - The *timeshifting* Broadcaster should operate in *compressed mode*, in order to conserve CPU, and prevent generation loss.
  - The *timeshifting* Broadcaster should have exactly the same encoding preset as the *live* Broadcaster, otherwise *compressed mode* is not effective.
  - The *timeshifting* Broadcastershould be on air and broadcasting to customers who wanted their shows timeshifted.
  - The *timeshifting* Broadcaster should have a different channel ID than the *live* broadcaster.

- The *timeshifting* Broadcaster is designed to consume only 1% -3% of CPU resources.
- Do not re-record using the *timeshift* Broadcaster.

To configure timeshifting:

1. Configure the *live* Broadcaster as usual. (For more information, refer to the *Broadcaster Getting Started User's Guide*.) Take note of the following information:
   - The video and audio encoding presets.
   - The location where the recorded files are located.
   - The name of the channel.
2. Using the *live* Broadcaster, record a short file and copy it to a new location, where it will not be erased by the Broadcaster's *delete history* feature.
3. Configure the *timeshift* Broadcaster as follows:

   a. In the *Channel Encoding Settings* page, select **Use Compressed mode**.

   b. Configure the encoding presets with the values used for the *live* Broadcaster (see step 1).

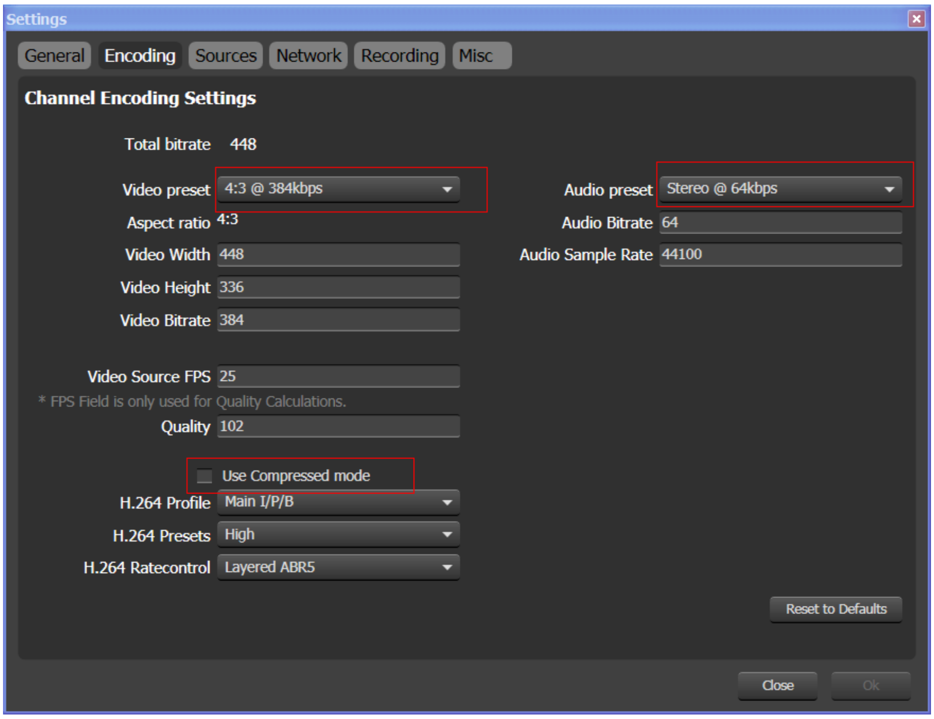

Figure 16: Channel Encoding Settings

   c. In the *Sources* panel, add the file you created in step 2 as a source.

   d. Create a new timeshift source.

   e. In the *Timeshift Folder* field of the *Edit* panel, select the folder where the *live* Broadcaster's recorded files are located (see step 1).

   f. In the *Filler Source* field, select the source you created in step 3c.

g. In the *Shift Value* field, select the appropriate timeshift offset.

h. In the *Timeshift Prefix* field, select the name of the *live* channel (as noted in step 1).

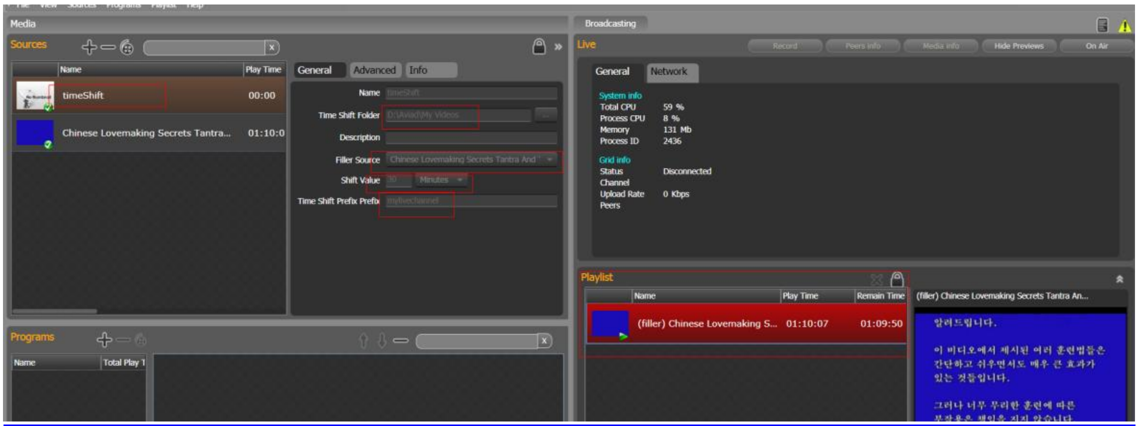

Figure 17: Configuring the *Timeshift* Broadcaster

i. Take note of the encoding settings in the *live* Broadcaster (refer to Figure 16).

4. Copy the encoding settings values from the *live* Broadcaster to the timeshift source. Select a filler source that you created specifically for the timeshift Broadcaster that has a matching encoding type.

5. In the *Recording Settings* page, take note of:

- The location where the recorded files are located.
- The name of the *live* channel.

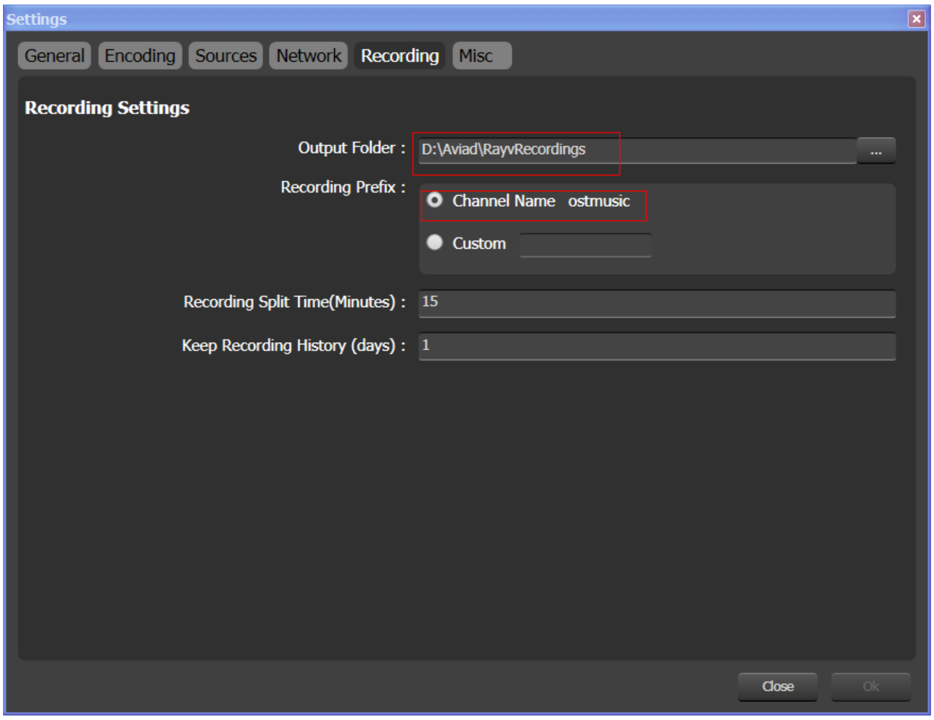

Figure 18: Recording Settings

5

CONFIGURING VOD RECORDINGS

The following are the VOD recording limitations:

- VOD viewing is based on flash technology. Therefore, all VOD files must be in flash format (.flv) with flash-compatible codecs.
- The H.264/AAC codecs RayV uses are only partially supported by flash. Flash supports a subset of these codecs called the *baseline* profile.
- In order for these files to be playable in VOD, the recording Broadcaster is limited to which video compression it can use, so B-Frames must be disabled. This has a slightly adverse affect on encoding quality therefore, we only use it for VOD encoding.

To configure VOD recordings:

1. From the Channel Encoding Settings page, disable B-Frames by selecting **Main I/P** in the H.264 Profile field.

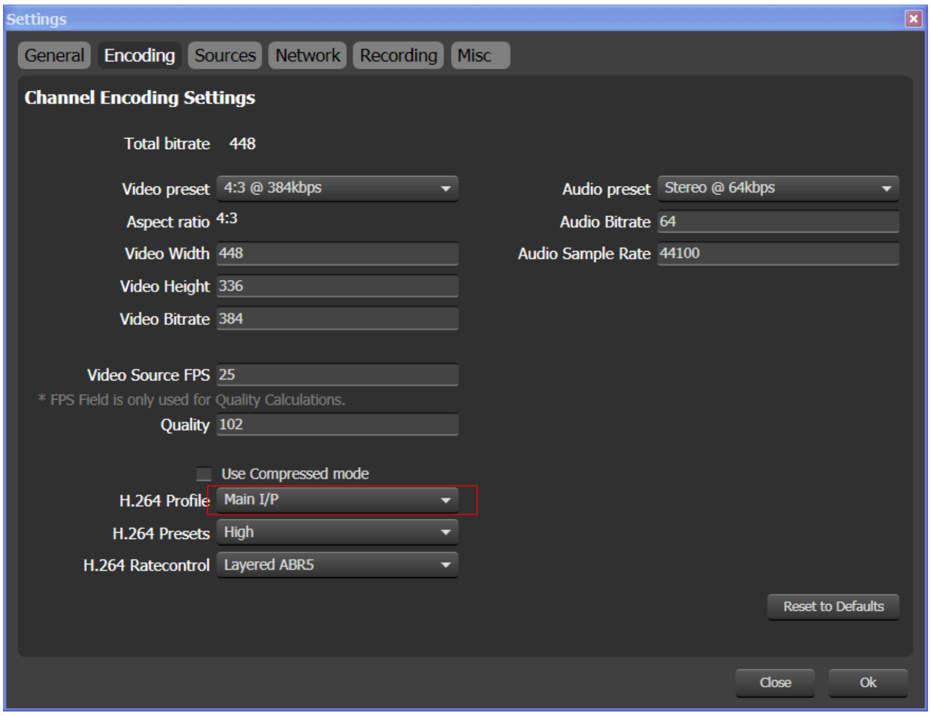

Figure 19: H.264 Profile

2. In the *Recording Split Time* field of the *Recording Settings* page, enter a time (in minutes) that is long enough to cover the entire event.

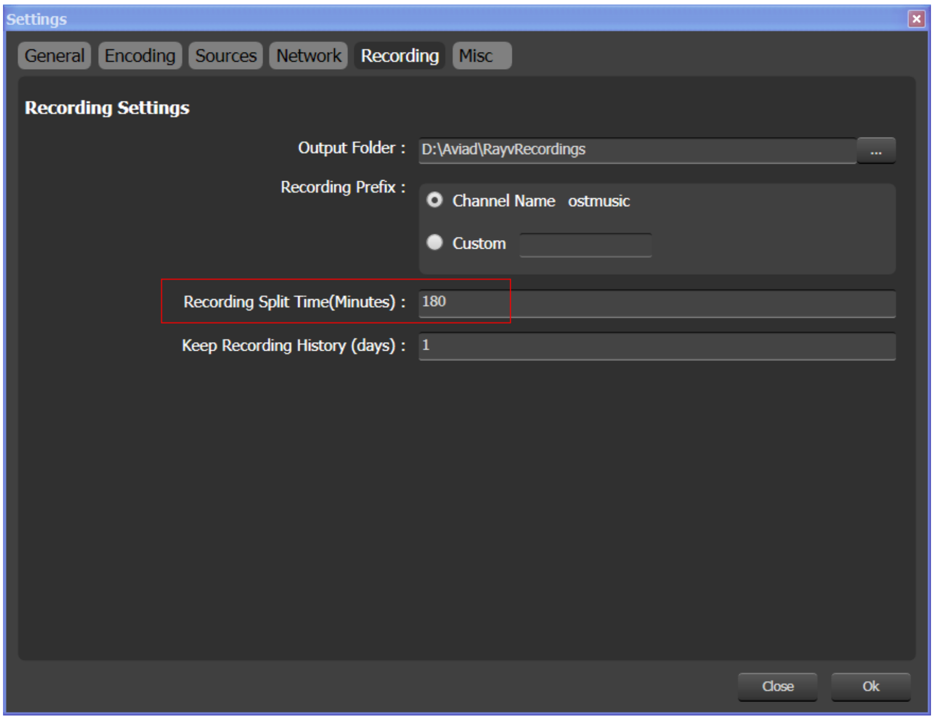

Figure 20: Recording Split Time

3. Start recording a few seconds before the event starts.

4. Stop recording a few seconds after the event stops.

6

Making the Broadcaster Less CPU Intensive

The broadcaster is a very CPU intensive server. The Broadcaster normally operates on an extremely strong HP G6, with 8 powerful cores. However, if enough broadcasters are on a G6, or if the broadcaster is operating on a dual-core machine, you might be operating beyond the 80% CPU guideline. If so, check the following before changing your Broadcaster configuration:

- Is one of the broadcasters a *timeshift* Broadcaster? Is it using more than 1% of the CPU? If so, check the timeshift configuration.
- Are you using too many Broadcasters on a single machine? If so, move one of the Broadcasters to a different machine and split the load.
- Is the Broadcaster configured with the correct encoding preset?

If you need to change the Broadcaster configuration:

- In the *H.264 Preset* field, change the value from **High** to **Normal**. This can reduce the CPU usage by 40%, without significantly affecting the quality.
- Use a lower Video preset (lower bitrate and resolution).

Note: *Changing the Video preset should not be done without approval from the project manager as it can lead to a very drastic change in quality.*

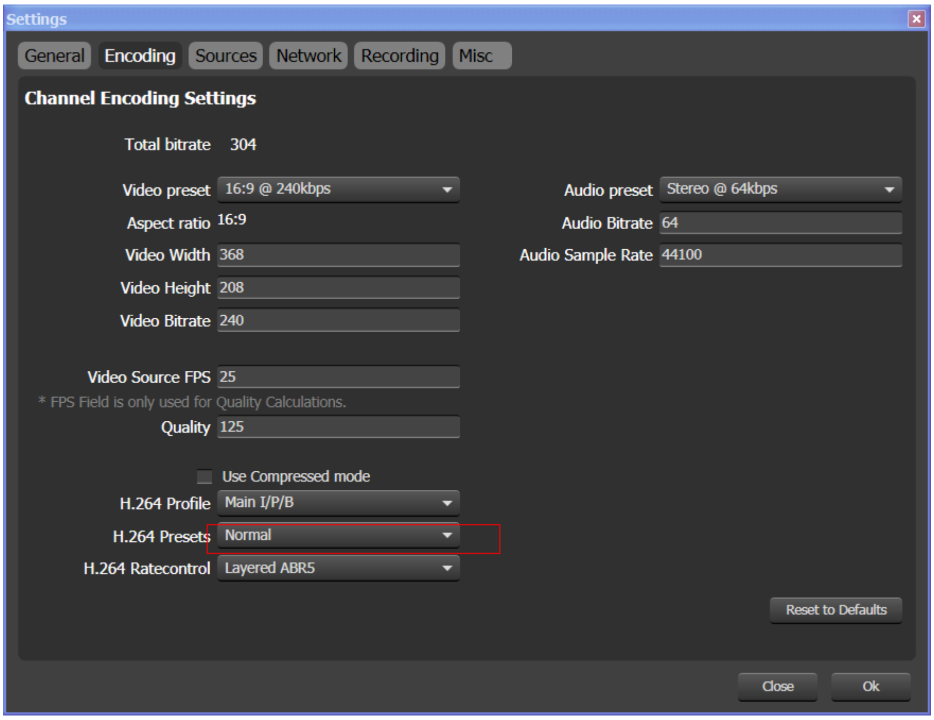

Figure 21: H.264 Presets

7

Configuring Uplinks

This section consists of the following topics:

- Broadcaster Mode
- Stream Output
- Configuring an Uplink
- Creating an MPEG-TS Source
- Recording to a File
- Configuring an MPEG-TS
- Configuring Multicasting

Broadcaster Mode

The Broadcaster can operate in one of the following modes:

- Tunneling Mode
- Raw Mode
- Compressed Mode

Tunneling Mode

Tunneling mode is a broadcaster setting, in which there is a single data stream and no audio and video streams.

This mode is useful in MPEG-TS scenarios, where the original stream passes from point-to-point with no modification.

This mode implies that there is no video encoding involved, so CPU usage is negligible.

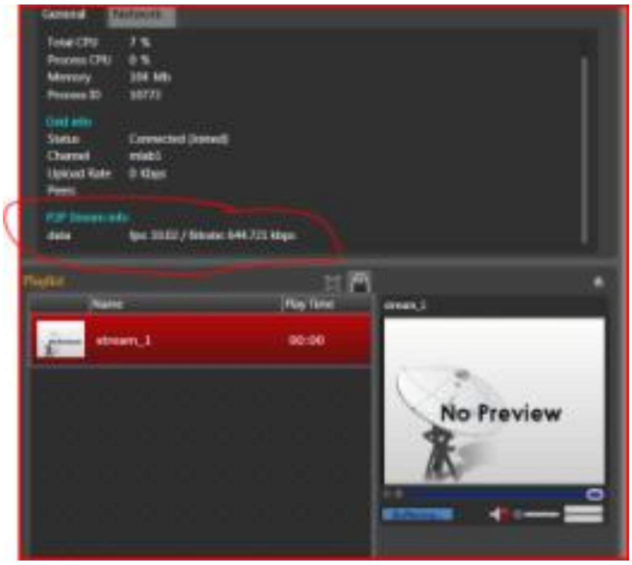

Figure 22: Tunneling Mode

Raw Mode

Raw mode is the default mode of the Broadcaster, in which audio and video is re-encoded when using P2P output.

This is the mode used when capturing from cards, using an MMS source, or broadcasting file playlists.

Compressed Mode

Compressed mode was introduced in version 0.9 of the RayV Broadcaster for the purpose of re-transmission of timeshift files without re-encoding the video. In this mode, compatible audio/video (in terms of bitrate, resolution and codec) is not re-encoded.

Compressed mode is used for timeshifting.

Stream Output

A stream output enables the Broadcaster to send content to third-party software, such as video players.

This is a crucial part of the uplink use-case, where it is used to send the MPEG-TS content to the client.

A stream output works in parallel with P2P output. Therefore, the Broadcaster can broadcast to P2P, record, and use the stream output at the same time or separately.

After configuration, stream output can be activated by clicking **Stream** on the main Broadcaster panel.

Note: *Stream output works only in compressed or tunneling mode. It does not work in raw mode.*

Note: *Currently, only one stream output can be set per Broadcaster.*

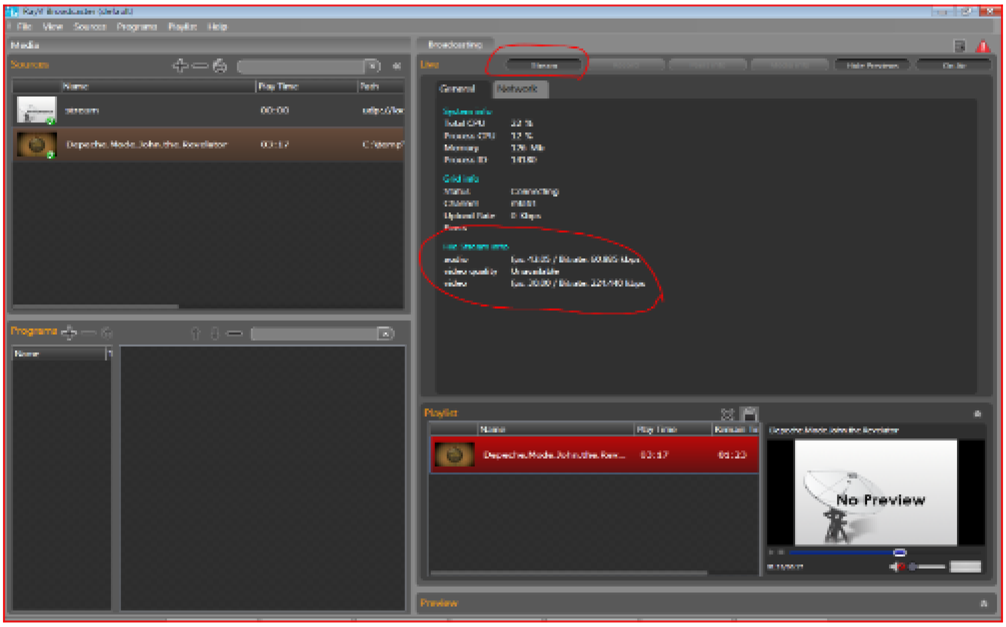

Figure 23: Activating Stream Output

CONFIGURING AN UPLINK

To configure an uplink:

1. In the *Bus Mode* field of the *Channel Encoding Settings* page, select **Tunneling**.

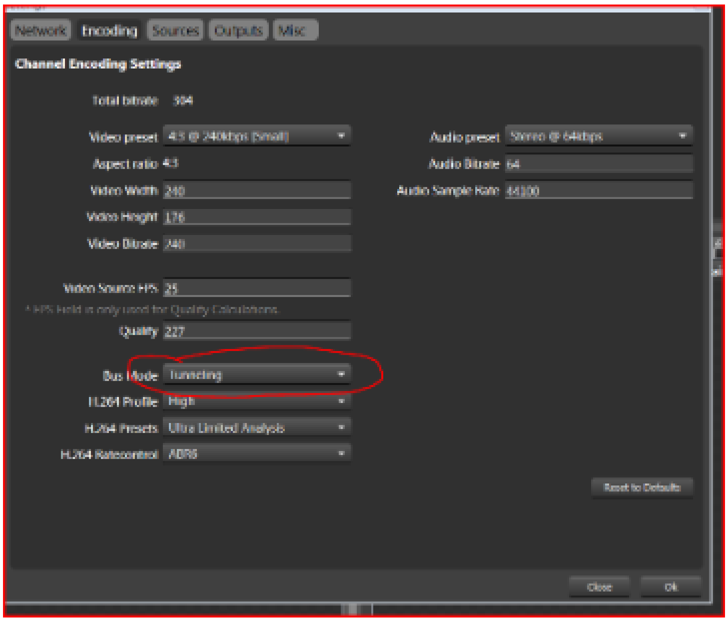

Figure 24: Select Tunneling Mode

2. In the *Stream Output* field of the *Channel Settings* page, enter the URL of the location of where the stream will arrive. The client will provide you with this URL. The URL is in the form `udp://<host>:<port>`. For internal testing using ffplay listening at udp://myserver?localport=6666, and use the URL `udp://myserver:6666`.

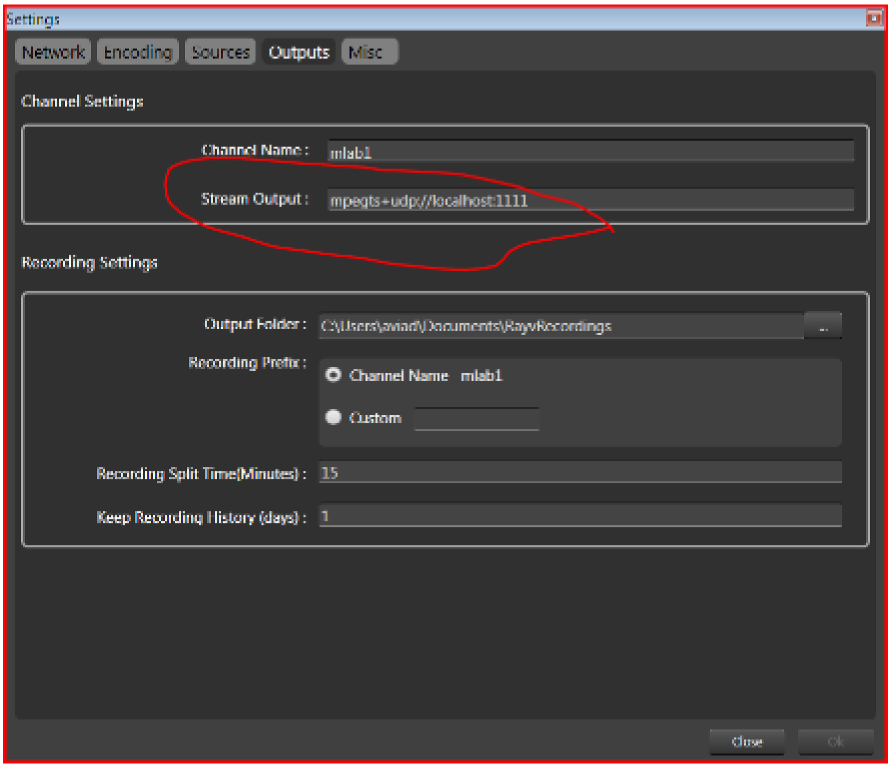

Figure 25: Stream Output

Creating an MPEG-TS Source

To create an MPEG-TS source:

1. In the *Bus Mode* field of the *Channel Encoding Settings* page, select **Compressed**.
2. In the *Stream Output* field of the *Channel Settings* page, enter a URL in the form `mpegts+udp://<host>:<port>`. For instance, if the URL is `mpegts+udp://localhost:2244`, you can run ffplay on the same machine with the command line `ffplay udp://localhost?localport=2244`.

Recording to a File

You can record to a file in either compressed or tunneling mode.

To record to a file:

1. In the *Bus Mode* field of the *Channel Encoding Settings* page, select **Compressed** or **Tunneling**.
2. In the *Stream Output* field of the *Channel Settings* page, enter the path of the file in the form: `file:///<fullpath>`, for example `file:///c:/temp/test.ts`.

Configuring an MPEG-TS

In an uplink use-case, the original MPEG-TS is generated by the client at point A, who wants to stream it to point B. In a production environment, this step is taken care of by the client.

In a test setting, it is necessary to generate an MPEG-TS stream, often without special hardware.

The following provides instructions for configuring a one time or permanent MPEG-TS generator.

Note: *It is also possible to do configure an MPEG-TS using other tools such as VLC or MEncoder. Instructions for using these tools is not in the scope of this document.*

Configuring a One Time MPEG-TS Generator using FFmpeg

FFmpeg is useful for setting up an MPEG-TS source. FFmpeg is installed in c:\program files\rayv\broadcaster when the Broadcaster is installed.

***Note:** FFmpeg only broadcasts the file once, and then stops. Therefore, it is recommended to use a long file. For a continuous MPEG-TS source, refer to Setting up a Stream Source on page 26 to setup the Broadcaster as a stream source.*

The syntax is:
```
$> ffmpeg -i <input-file> -re -s <resolution> -f mpegts -b <bitrate>
<url>
```

For instance, to transcode the file Bray-Cocoon_PC.avi, which is located in the directory `c:\temp`, at 3000kbps bitrate, 640x480 resolution, to URL localhost:1234, use the following command line

```
$> ffmpeg -i c:\temp\bray-cocoon_PC.avi -re -b 3000k -s 640x480 -
f mpegts udp://localhost:1234
```

To connect to this stream using FFplay, use the following command line:

```
$> ffplay udp://localhost?localport=1234
```

Notice how in the FFmpeg command line, the *destination* port is 1234, and in the ffplay command line, the *source* port is the same as the one ffmpeg sends data to.

***Note:** Only one application can connect to a stream at a time.*

To connect to the stream using the Broadcaster, create a stream source, and enter the same URL as you would in ffplay [i.e., `udp://localhost?localport=1234`]

Setting up a Permanent MPEG-TS Generator using the Broadcaster

1. In the *Bus Mode* field of the *Channel Encoding Settings* page, select **Compressed**.
2. Select the appropriate resolution/bitrate.
3. Set the stream output to a URL in the form `mpegts+udp://<host>:<port>`, where host and port are the address to which you want the MPEG-TS streamed.
4. Place the file you want to transmit in the playlist.
5. Click **Stream** to activate streaming.

Setting up a Stream Source

In Broadcaster 1.2, the stream source [MMS source] can be used for more than just MMS. Specifically, it can be used to receive MPEG-TS input from network.

In an uplink use-case, the client provides you with the URL of the MPEG-TS stream. In a test setting, if you set up the source with a URL such as `udp://myserver:3456`, then the URL used in the Broadcaster is `udp://myserver?localport=3456`

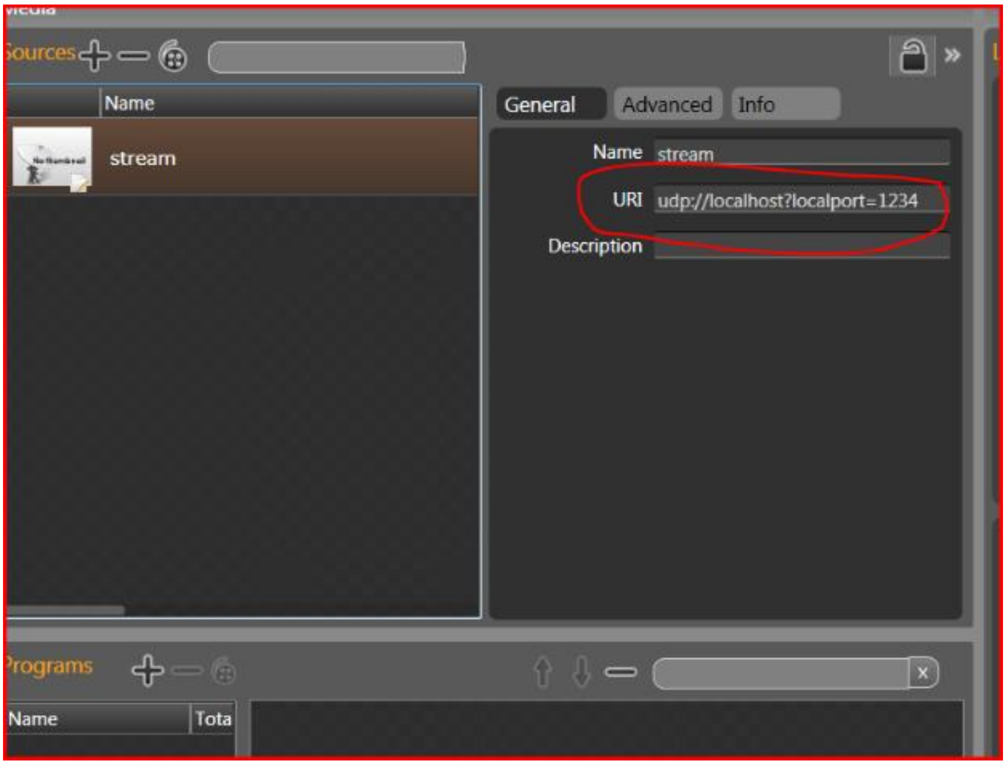

Figure 26: Setting up a Stream Source

This stream source can be used like any other source, with the addition that this is the only kind of source that can be used in Tunneling mode.

LEECHING

Leeching is the ability of the Broadcaster to behave like a gateway/viewer, by connecting to the P2P grid in a gateway role, and accepting inbound P2P traffic (or leeching) a channel.

The term leeching is borrowed from P2P file sharing terminology (such as torrent or emule) where it designates clients who are downloading content rather than uploading and contributing bandwidth.

This ability is useful in several gateway-like scenarios, such as down-transcoding video, recording, monitoring, etc,, but more importantly, it is used in uplink A→B scenarios, at node B to receive the MPEG-TS data from RayV grid.

How Leeching Work

In Broadcaster 1.2, the Broadcaster maintains two separate engine/server sessions. The first is the broadcasting [or seeding] engine/server session that the Broadcaster has always supported. The seeding session always connects to the server with a *broadcaster* role.

The second engine/server session, is built around the viewer SDK (RayV.dll) and connects to the grid independently, with its own set of ports, users, STUN, etc. The leeching session always connects to the server with a *gateway* role.

Configuring the Server Connection Settings

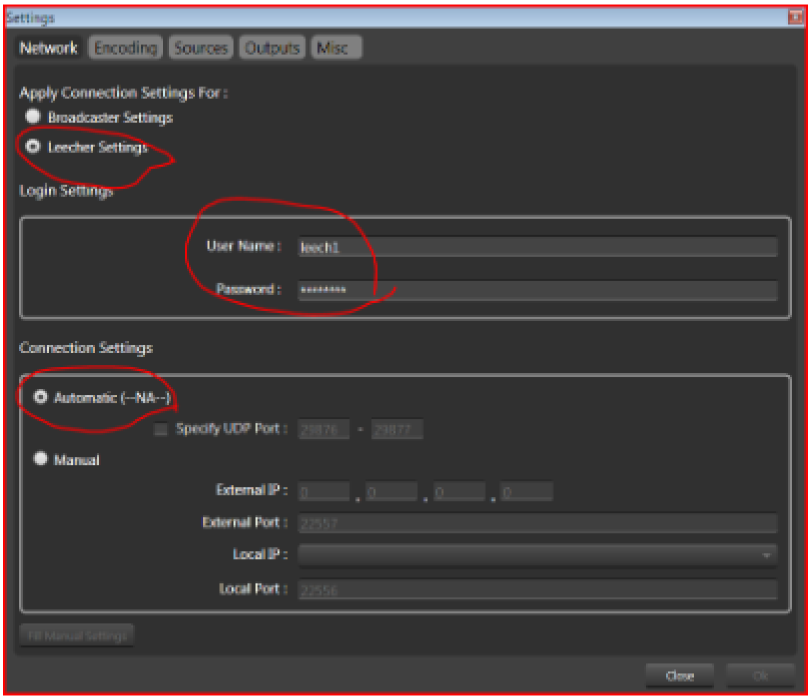

Figure 27: Network Settings, with Dual Entries for Broadcasting and Leeching

1. Click **View > Settings > Network**.
2. Select **Leeching Settings**.
3. In the User Name and Password fields, enter the username and password.

   Note the following:

   - Enter the actual password, not a hash code.
   - Leeching will not work without the proper credentials.
   - The user must exist in the server's database (http://admin.rayv-inc.com).
4. If there are special firewall considerations, in the *Connection Settings* section, select **Manual** and configure the IP/ports for leeching. Otherwise, select **Automatic**.

Creating the Source

1. In the *Sources* panel, click **+** and select **Channel** to create a channel source.

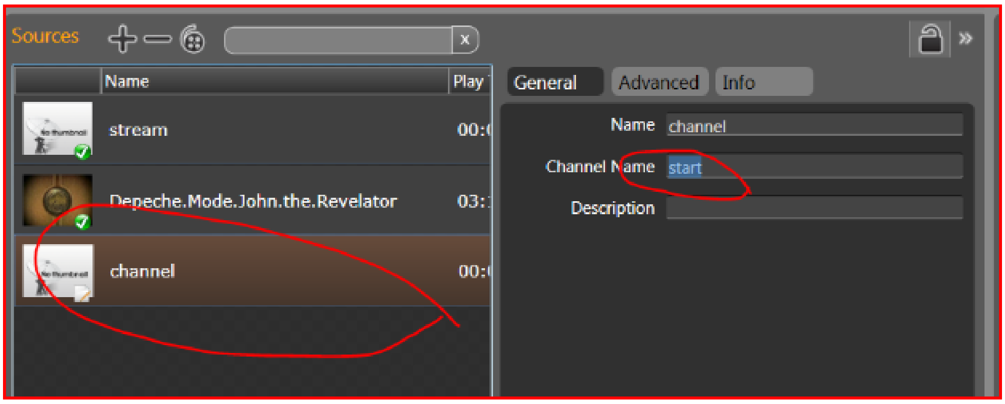

Figure 28: Setting up a Channel Source

2. Enter the channel name (such as france24fr for channel http://france24fr.rayv.com) in the *Channel Name* field
3. Press F3 to place the channel source in the playlist.

Note: *For leeching an uplink channel, the Broadcaster must be in Tunneling mode.*

How it all Fits Together

Figure 29 shows the workflow for configuring an MPEG-TS.

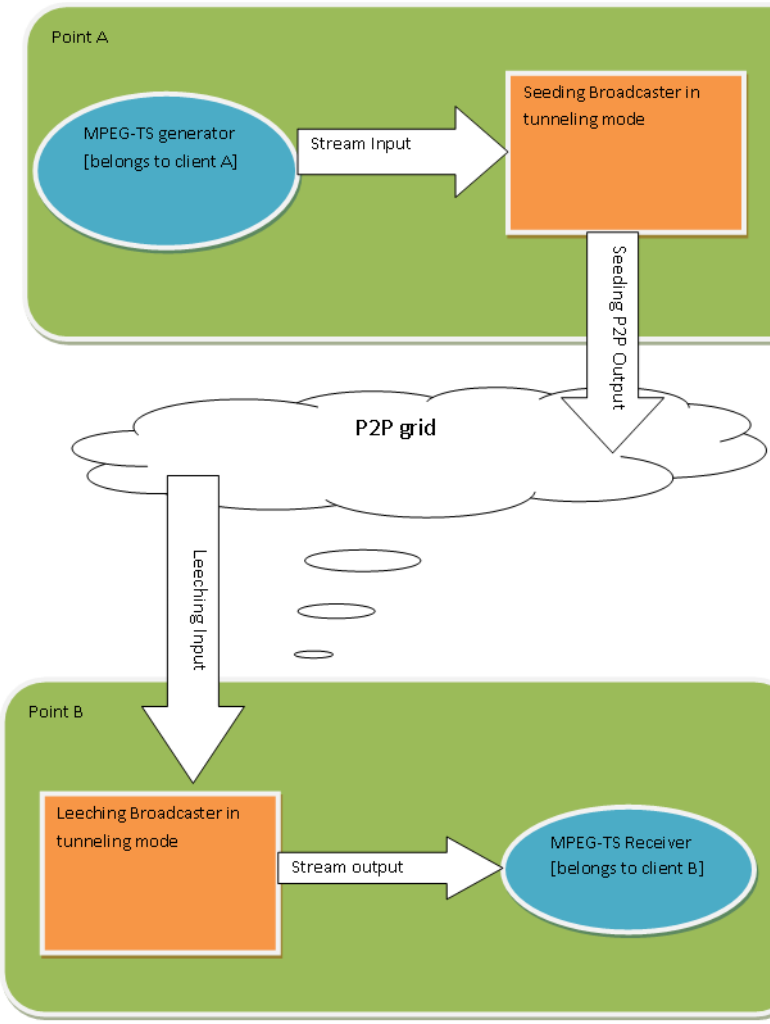

Figure 29: Configuring an MPEG-TS

Configuring Multicasting

Multicasting is a TCP/IP feature that is managed by your network administrator. It operates inside a local network or subnetwork. The address range 224.0.0.0 to 239.255.255.255 is known as the multicast range.

Packets sent by an application to a multicast address are received by all other applications in the network that joined this address.

With FFmpeg, VLC, or the Broadcaster, this means that using a multicast address as the output URL enables multiple clients to connect to this URL in parallel.

For example, running:

```
$> ffmpeg -i c:\temp\bray-cocoon_PC.avi -re -f mpegts udp://224.0.0.0:1234
```

enables any number of clients (FFplay, VLC, Broadcaster) to connect to this stream, at the same time using the same address as follows:

```
$> ffplay udp://224.0.0.0:1234
```

For more information, consult with your network administrator or IT group or visit the following website:
http://en.wikipedia.org/wiki/Multicast_address

8

TROUBLESHOOTING

The section contains the following troubleshooting information:

- Sending Logs
- Identifying if the Broadcaster is Not Connected to a Satellite
- Identifying if the Broadcaster is Not On Air

SENDING LOGS

If the broadcaster crashes, or if you need to restart due to problems, send the log files to RayV.

To send the log files to RayV:

1. With the Broadcaster active, press **CTRL+S**. The *Support* dialog box appears, with a message that the Broadcaster is generating support information. After a few seconds, the message changes to, *Support log was created successfully*.

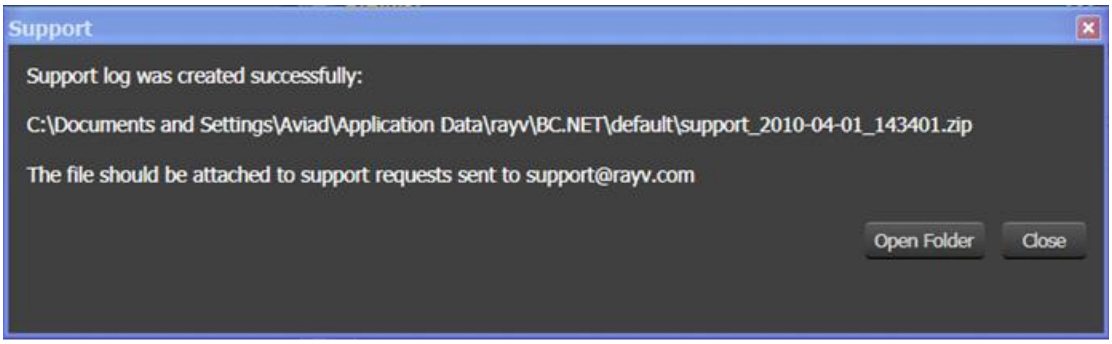

Figure 30: Support Dialog Box

2. In the *Support* dialog box, click **Open Folder**. A Windows Explorer folder opens containing a zip file.
3. Email this zip file to support@rayv.com.

Identifying if the Broadcaster is Not Connected to a Satellite

If the Broadcaster is not connected to a primary satellite, it has no direct peers, and therefore its upload speed is zero. This can occur even if the Broadcaster, feed, and encoding are all working.

This situation can be seen in the Network tab of Broadcaster's Live Info panel.

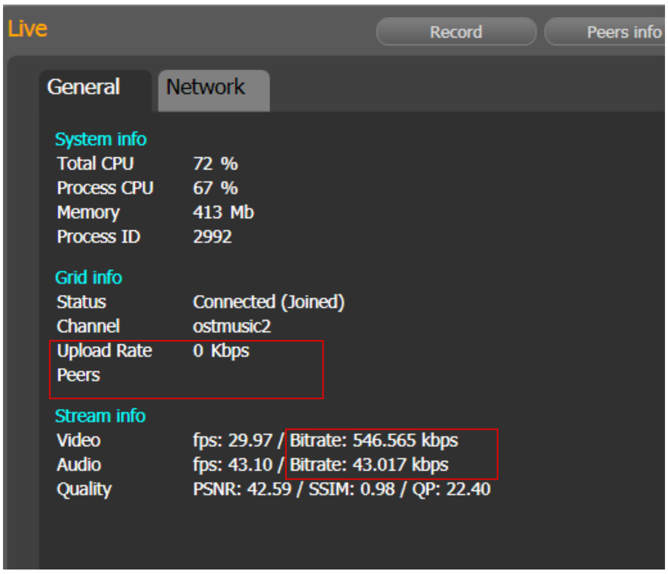

Figure 31: No Satellite Connection

In Figure 31, the Broadcaster and encoding are working properly, but the upload rate is 0.

Identifying if the Broadcaster is Not On Air

When the Broadcaster icon is grey , the Broadcaster is offline.

The Broadcaster is setup to be on air (the *On Air* button was clicked), but the Broadcaster is not actually on air because the upload is 0. This situation occurs when the Broadcaster is connected to the server, and joined with the channel, but there are no peers to which to upload (refer to Figure 32).

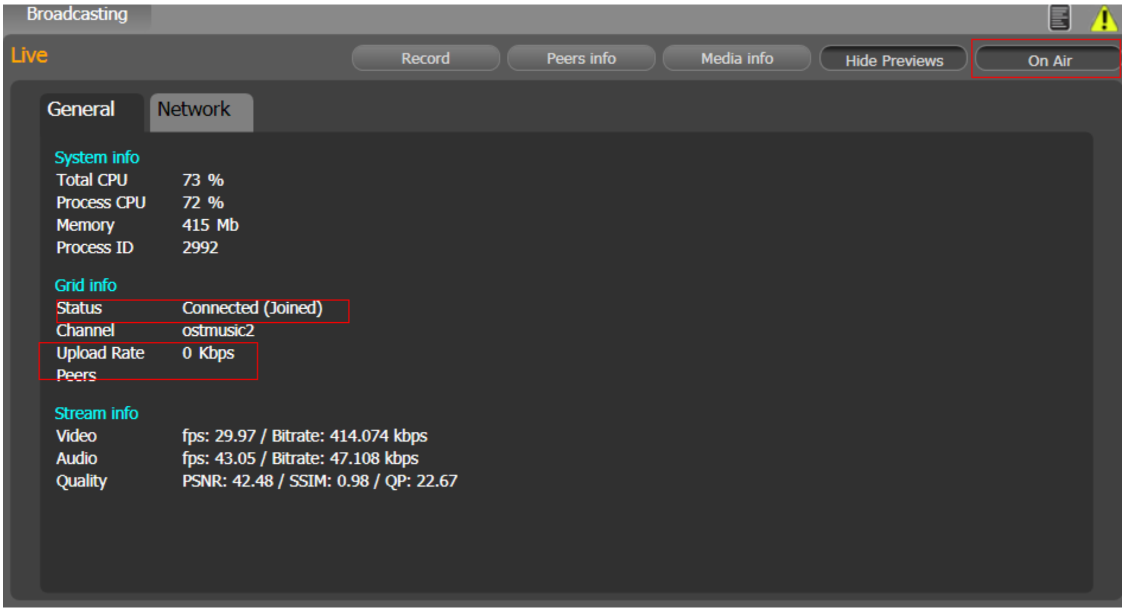

Figure 32: On Air Button Clicked, Connected to Server, No Peers

When the Broadcaster icon is green , the Broadcaster is online.

In this case, the Broadcaster is not setup to be on air (the *On Air* button was not clicked). The Broadcaster is connected to the server, but it is not joined with the channel (refer to Figure 33).

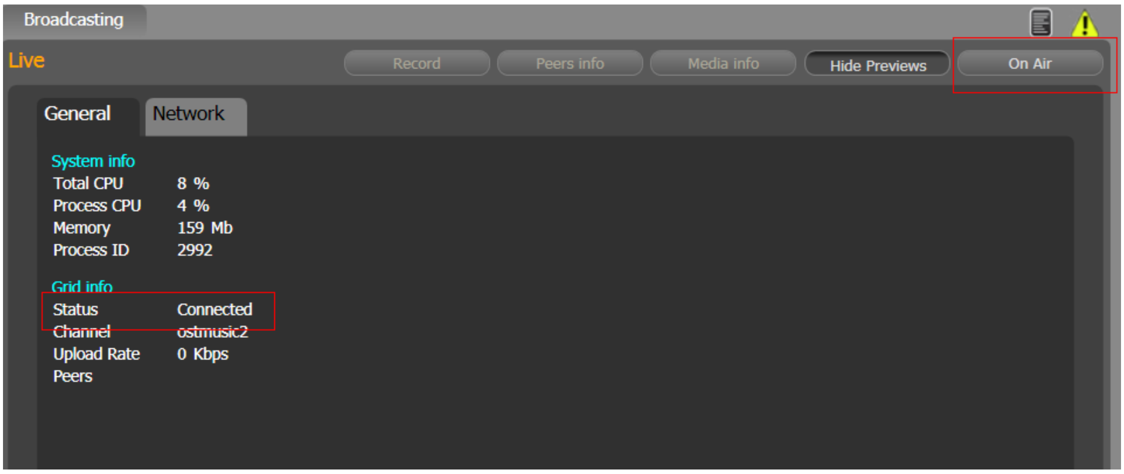

Figure 33: On Air Button not Clicked, Connected to Server

Figure 34 displays the message that appears when the Broadcaster cannot operate in offline mode because there is a problem with connecting or authenticating with the server.

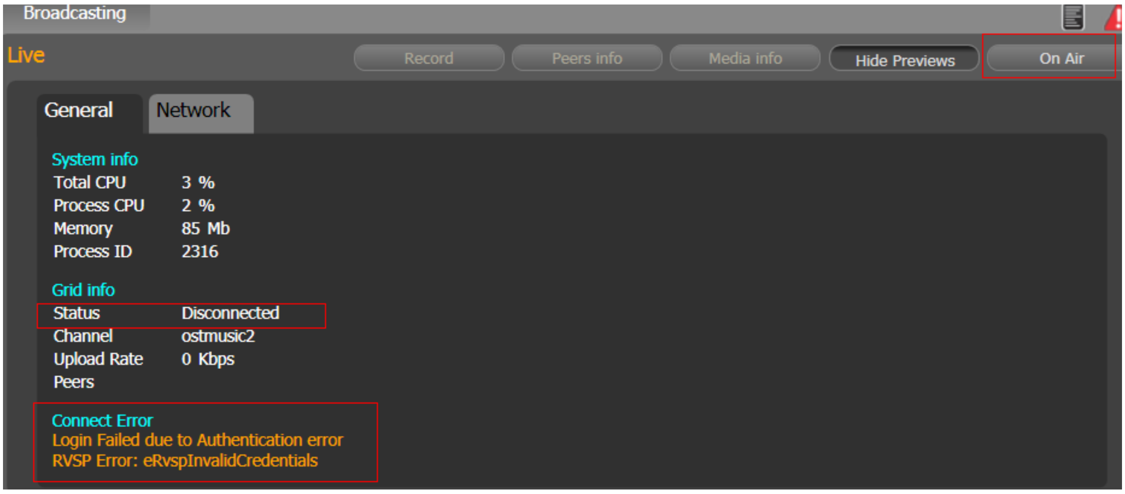

Figure 34: Problem with Connecting or Authenticating