envivio.®

4Sight

Administrator Guide

Version 5.0 rev.4

May 17, 2006

P/N 01000002

The leading provider of MPEG-4 broadcast and streaming solutions

Acknowledgments

Envivio Incorporated acknowledges use of software from the following sources and includes their copyright notices and required statements. here.

This software is based in part on the code of the Darwin Streaming Server (for which the source code is available under the terms of the Apple Public Source License Version 1.2 at http://www.opensource.apple.com/projects/streaming/).

The modifications applied to the original code are available at: http://www.envivio.tv/downloads/qtss.

Contents

CHAPTER 1

Introduction

Overview

The 4Sight Video Server from Envivio provides MPEG-4 and H.264 streaming capabilities for small, medium and large scale enterprise, broadband and broadcast deployments. It has been designed and optimized to meet the needs of professional video distribution applications.

Broadcasters and cable operators can use the 4Sight Video Server to manage and distribute interactive multimedia content through IP networks to EnvivioTV or any other MPEG-4 compliant player

VOD functions

The 4Sight Video Server can stream any MPEG-4 file that conforms to the ISO standard, whether the file is generated by the Envivio 4Coder or some other MPEG-4 encoding tool.
Both Advanced Video Coding (AVC/H.264) and Advanced Simple Profile (ASP) are supported.

Files need to be hinted before being served by the 4Sight Video Server.

The 4Sight Video Server supports trick play features (fast forward, fast backward) on all MPEG-4 content. The rendering of content played in fast forward or fast backward depends on the way the content is encoded (more specifically on the key frame frequency).

Networking

The 4Sight Video Server sends media to the network using the IETF defined suite of protocols based on IP for real-time streaming, and supports HTTP tunnelling to support firewall configurations. It also uses

the associated IETF recommendations dedicated to MPEG-4 transport over IP.

Monitoring and administration

Monitoring of the 4Sight Video Server can be done while it is online using either the 4Sight WebAdmin which is a web-based interface or using SNMP.

When the 4Sight Video Server is offline, monitoring can be done using the log files.

Performance

In streaming, 4Sight can support up to 400 simultaneous users, at 1.5 Mbps.

Error Resilience

4Sight offers error resilience so that EnvivioTV can receive content smoothly under severe network packet loss conditions (up to 2% packet loss rate).

Both 4Sight and EnvivioTV provide a rebuffering management policy that improves the user experience when traffic is congested.

Supported standards

The following protocols and standards are implemented in the 4Sight Video Server:

- RTSP (Real Time Streaming Protocol), IETF RFC 2326

- SDP (Session Description Protocol), IETF RFC 2327

- UDP (User Datagram Protocol), IETF RFC 769

- TCP (Transmission Control Protocol), IETF RFC 793

- IP (Internet Protocol), IETF RFC 791

- RTP (Transport Protocol for Real-Time Applications), IETF RFC 3550

- SAP (Session Announcement Protocol), IETF RFC 2974

- IGMP v2 (Internet Group Management Protocol), IETF RFC 2236

- SNMP (Simple Network Management Protocol), IETF RFC 1157

Envivio Professional Services and Support

You may contact us for specific projects requiring customized options or specific development, available through our service organization. For contact information, see the Envivio website at http://www.envivio.com.

If you have support questions, contact Envivio Customer Services.

Version 5.0

WebAdmin

Overview

4Sight WebAdmin is a web based administration tool that allows the administrator to monitor and configure the 4Sight Streaming Server. Using this tool, you can setup and monitor relays, switcher and playlists. You can also setup movie folders, authorization and streaming ports.

In order to connect to this administration tool, you can either:

Restricting access to the WebAdmin

Using the WebAdmin configuration file

To restrict access to the WebAdmin, you can use the two keywords, *allow* or *deny*, in your streamingadminserver.conf.

For example, at the end of the file, add the following:

```
allow=192.168.105.187 127.0.0.1
deny=192.168.105.113
```
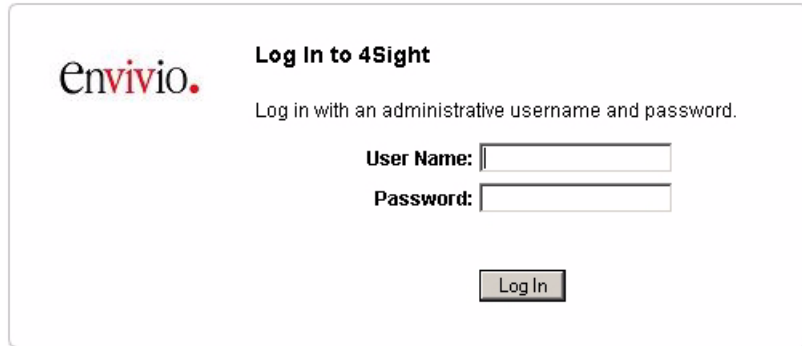
This allows access to the WebAdmin from local host and 192.168.105.187 and explicitly denies it from 192.168.105.113.

Using the WebAdmin user interface

You can restrict access to the WebAdmin by using a login screen. The user will have to provide a user name and password to connect to the WebAdmin.

1. Using a web browser, connect to the WebAdmin (see "WebAdmin" on page 17).

2. Click on the **Server Configuration>General Settings** menu item.

3. Select the **Authentication for the web admin** check box to restrict access to the WebAdmin, then click the **Save changes** button. You will be disconnected from theWebAdmin.

Version 5.0
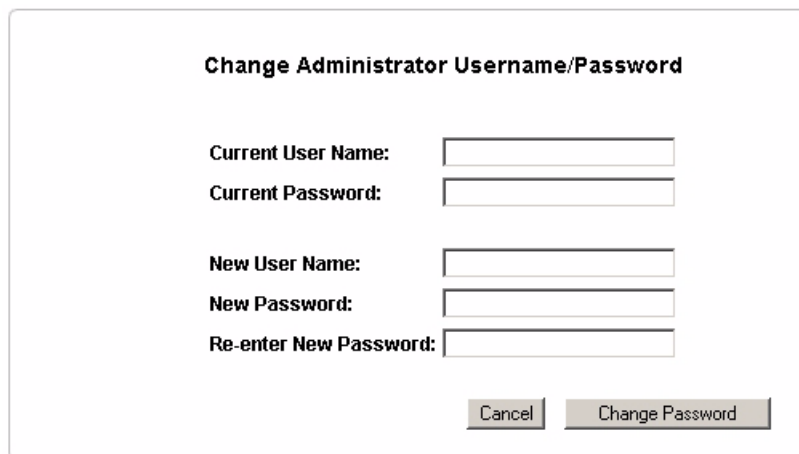
4. Reconnect to the WebAdmin using the default user name (*admin*) and password (*admin*).

Changing the WebAdmin default user name and password

5. Using a web browser, connect to the WebAdmin: (see "WebAdmin" on page 17).

6. Select the **Server Configuration>General Settings** menu item.

7. Click the **Change Admin Username/Password...** link. The following page should be displayed:

Version 5.0

8. Fill the different fields, then click the **Change password** button. You will be disconnected from the WebAdmin.

9. Reconnect to the WebAdmin using the new user name and password.

Video Server

Setting up Movie folders

To serve on-demand MPEG-4 files stored on your server, 4Sight needs to know where your files are located.

You can change the default movie folder or add new ones using the WebAdmin:

1. Open a web browser and connect to your WebAdmin (see "WebAdmin" on page 17 for further details).

2. Click the **Movie Folders** menu item. You will see a list of movie folders as shown in the figure below:

Movie Folders Settings
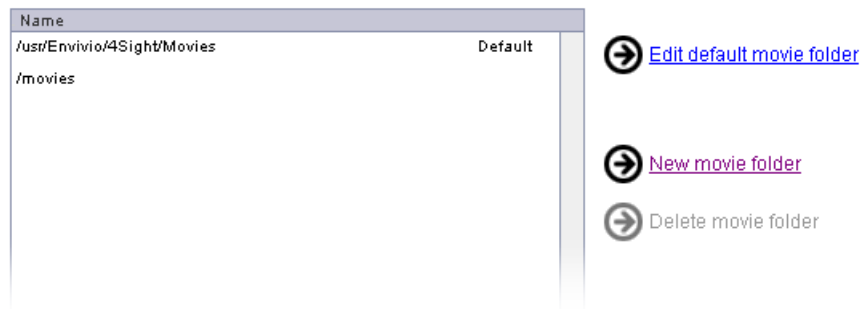
3. To change the default movie folder, click the **Edit default movie folder** button.

4. To add a movie folder to your list, click the **New movie Folder** button. Then enter the complete path of your folder.

NOTE The path to the **movie** folder must respect the path format of the operating system where 4Sight is installed.

---

5. Click the **Saves Changes** button to add this new folder, as shown in the figure below.:

Movie folder details

Movie folder name: `/usr/Envivio/4Sight/Movies`

Save Changes

Using 4Sight as a reflector

Overview

Most routers on the Internet cannot properly interpret multicast information as they are set up to transfer traditional Internet Protocol (IP) unicast packets that have a single, specific destination.

A reflector is a built-in mechanism of the server that converts one multicast stream to a series of individual unicast streams. The server's reflector module feeds each client with an individual stream.

Basic setup

In most cases, the reflector receives either a multicast or unicast stream from a broadcaster (an application which sends RTP packets) such as 4Caster. The reflector forwards the RTP streams upon request, in real time. The SDP file generated by the broadcaster must be placed in one of the movie folders of 4Sight.

Configuration examples

Once your broadcaster is setup and has started to broadcast to a multicast address, you can get the SDP associated with the multicast stream. Then, without changing anything in the SDP file, you can copy it into one of the server's movie folders. You are then ready for accessing the multicast stream using the 4Sight Video Server reflecting capabilities.

Figure 1. Example of reflecting
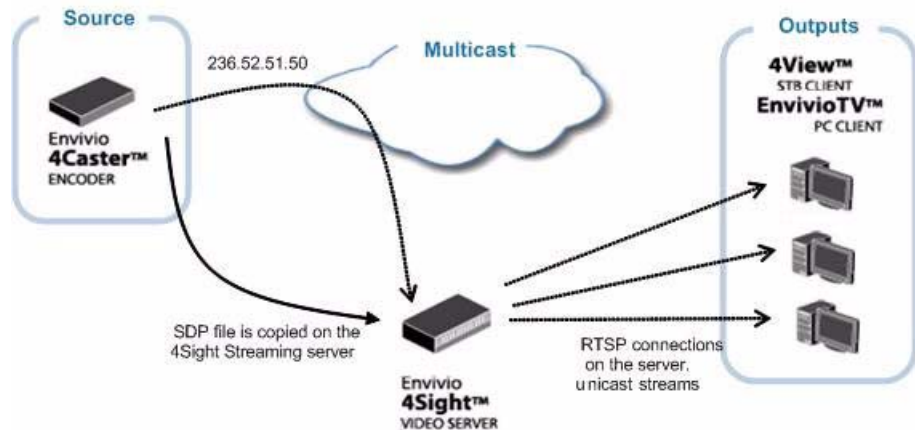
For example, if the SDP file referring to the multicast stream is called **live.m4e**, once it is copied on the server, you can access the multicast stream using the following URL:

```
e-rtsp://server_name:7070/live.m4e
```

A unicast session will then be created for your client that will be a reflected copy of the multicast stream.

WARNING   If you use reliable UDP (see "reliable_udp" on page 90), check that the M4E file does not contain the following line:
b=RR:0
This may prevent the player from sending the RTCP packets that are mandatory for using reliable UDP.

Using 4Sight as a relay

Overview

A relay is a built-in mechanism of the server that accepts a stream from another video server or a broadcaster and sends it or "relays" it to one or more unicast or multicast destinations.

Basic setup

To set up the relay using the WebAdmin, follow these steps:

1. Using a web browser, connect to the WebAdmin (see "WebAdmin" on page 17).

2. Select the **Relay Settings** menu item. You will see a list of your configured relays.

3. Click on **New Relay** to create a new relay. The following page is displayed:

4. Fill in the different fields, then click the **Save Changes** button.

5. Your relay will then be listed in the relay list, followed by its status that can be either *Started* or *Stopped*.

NOTE   You can click the ▶ or ■ buttons to respectively start or stop the relay.

Configuration examples

Relay a UDP stream

Consider the following graph where the user wants to relay a multicast stream to another multicast address and simultaneously to a unicast client.

Figure 2.   Relaying a UDP stream
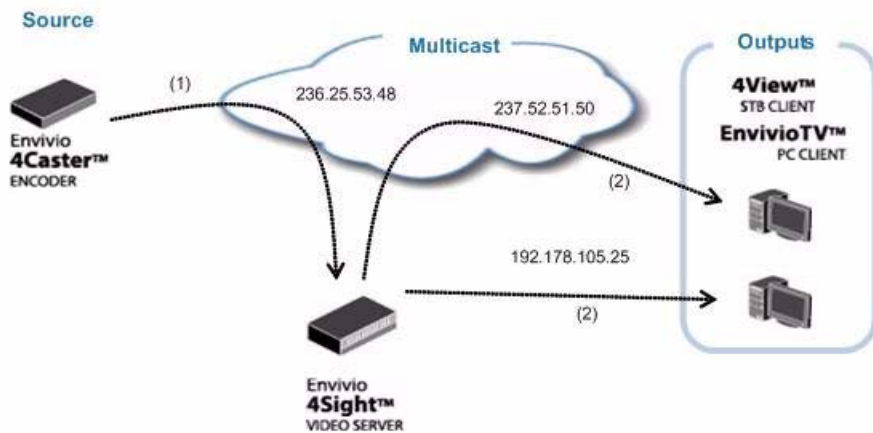
In this case, the relay module of the 4Sight Video Server Streaming Server will fetch the multicast stream coming from the broadcaster **(1)** and send it to both the multicast address and the unicast client **(2)**.

To relay a UDP stream, you need to follow these steps:

1. Specify the source address (this address can be either multicast or unicast to the server).

2. Select the **Relay a UDP Source** radio button.

3. Specify the ports you want to listen to (ports number can be separated by commas, semicolons or spaces).

4. Provide the information about the destination (you can add more destinations if needed).

5. Click the **Save Changes** button.

Source Settings

Description (optional): Relay Source
Source Hostname or IP Address: 236.25.53.48

⊙ **Relay UDP Source:**
UDP Port List: 5004, 5006

4Sight will then start relaying the source address to the configured destination addresses:

Destination Settings (1 of 2)

Description (optional): Relay Multicast Output
Hostname or IP Address: 237.52.51.50
UDP Ports: 6606, 6608
Multicast TTL: 5
Local Output Interface: 192.168.105.138
SAP Session Name (optional):

Remove Destination

Destination Settings (2 of 2)

Description (optional): Relay Unicast Output
Hostname or IP Address: 192.168.105.25
UDP Ports: 7070, 7072
Multicast TTL: 5
Local Output Interface: 192.168.105.138
SAP Session Name (optional):

Remove Destination

Version 5.0

You can define the network interface card that you want to use for sending packets by specifying the **Local Output Interface** parameter in the **Destination** section.

The SAP session name for outputs is only available when relaying an SAP source.

NOTE   The multicast time to live (TTL, see "Glossary" on page 109) is mandatory if you relay towards a multicast address, and it must be greater than 0. It can be omitted if relaying to a unicast address.

Relay an RTSP requested stream

This configuration is for the scenario where you have a server configured and running that you would like to relay, but the server is not broadcasting anything, it is just accepting and handling requests from clients. In order to relay requests, you will need to have an incoming stream.

Figure 3.   Relaying an RTSP requested stream

Version 5.0
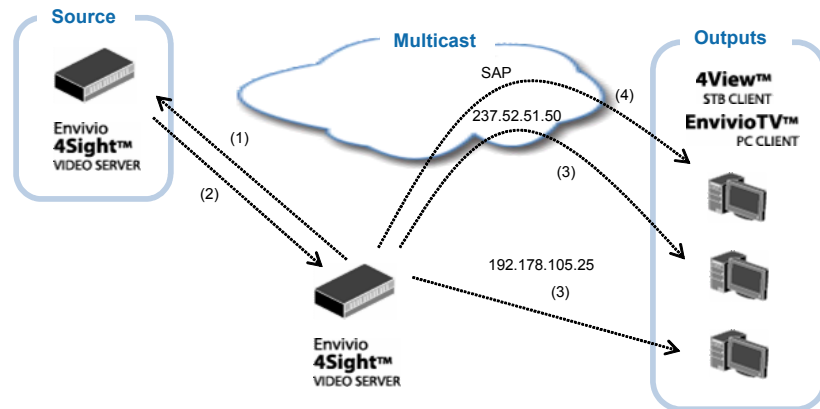
The 4Sight Video Server Video Server will request a stream from the RTSP server specified in the source settings **(1)**, get the video stream back **(2)** and relay it to the destinations **(3)**. If an output SAP session is specified, then the relay will also announce the new session using SAP **(4)**. A player will then be able to connect using SAP to the relayed stream.

By setting the IP address and the filename of the movie you want to relay, and selecting the **Request Incoming Stream** radio button, your server will get the stream as if it was a client and relay it to the destinations you configured.

If the origin server needs authorization, provide a user name and password in your relay configuration as shown in the figure below:

Source Settings

Description (optional):	Relay Source
Source Hostname or IP Address:	192.168.105.32

○ **Relay UDP Source:**
  UDP Port List: [   ]

◉ **Request incoming stream**
  MPEG4 or SDP File: movie.mp4
  User Name: user
  Password: ••••••••

○ **Wait for an SAP announced session**
  SAP session to listen: [   ]   [--- No session selected --- ▾]

Let's consider the same use case than in the previous paragraph, but instead of having a broadcaster as a source, we have a second video server that is able to handle RTSP requests and serve videos. In this case, we can configure our relay to request the video on the source server and relay it the different outputs we set up.

Version 5.0

Relay an SAP announced stream

In this last case, we suppose that we have a broadcaster that is able to announce the session to the SAP address. In this case, the 4Sight's relay module is able to connect to the announced stream and to send it to the different outputs we set up.

Figure 4.   Relaying an SAP announced stream
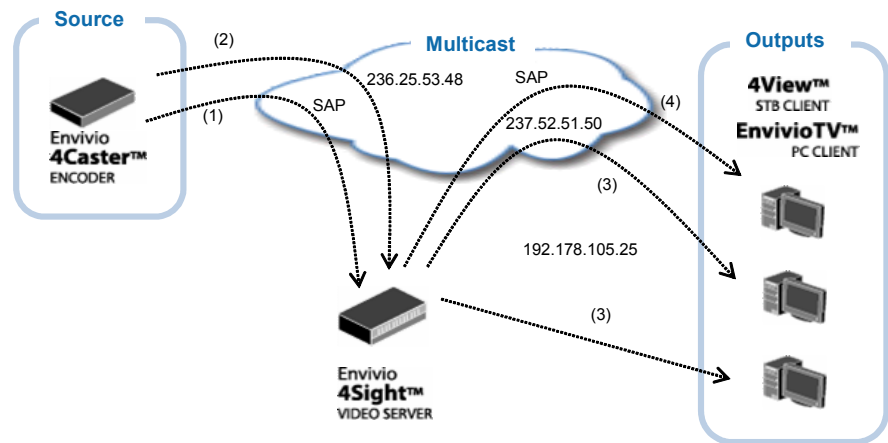
The 4Sight Video Server will wait till it gets the announcement specified in the source settings **(1)**, then, it will connect to the announced stream **(2)** and relay it to the destinations **(3)**. If an output SAP session is specified (as shown in the screen capture below), then the relay will also announce the new session using SAP **(4)**. A player will then be able to connect to the relayed stream using SAP.

Version 5.0

The following capture shows the source setup for an SAP relay:

Source Settings

Description (optional):	Relay Source
Source Hostname or IP Address:	

○ **Relay UDP Source:**

UDP Port List:

○ **Request incoming stream**

MPEG4 or SDP File:

User Name:

Password:

⦿ **Wait for an SAP announced session**

SAP session to listen: news | news ▼

When the 4Caster sends a unicast stream to the 4Sight Video Server Video Server, it also sends a unicast stream with the connection SDP. This stream is interpreted as an SAP stream by the 4Sight Video Server which acts exactly the same way than in the setup described above.

Figure 5.   Relaying a 4Caster unicast stream

Version 5.0
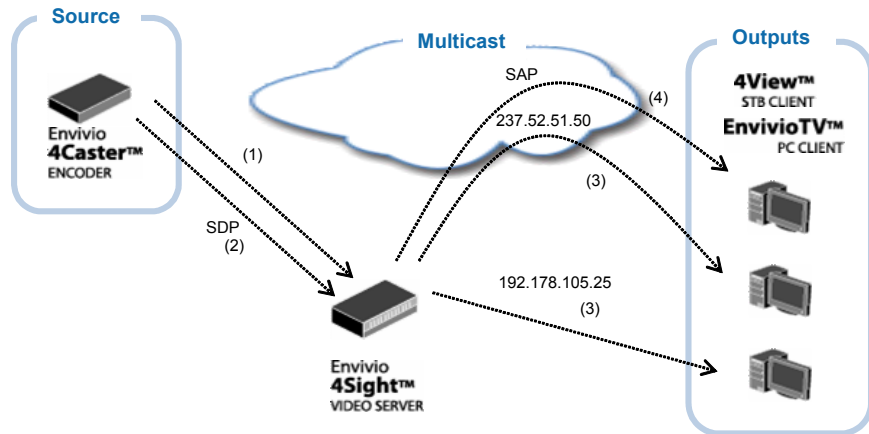
Connection to the output stream using an SDP file

To connect to an output stream of the relay, you will need to provide the description file of the stream (SDP file). The easiest way to do it is to get the SDP file from the source and modify it according to the output address and ports values that you set in the relay.

Once the SDP file is modified, you can feed it directly to the client or put it in the movie folder of the 4Sight Streaming Server and access it using an RTSP request.

For further details on how to modify the SDP file, see "How to update an SDP file" on page 80".

Connection to the output stream using SAP

In the case of an SAP relay, there is a simpler way to connect to the output stream. In the destination settings of your relay, you can specify an SAP session name. The relay will then forward the SAP messages containing the new connection settings for the relayed stream.

Then, if your player supports SAP URLs, you can feed it with an URL such as the following:

```
sap://session_name
```

Status of the configured relays

Once everything is configured, you can check that the source stream is effectively relayed by going to the **Relay status** page.

The following window should be displayed:

Version 5.0

Stream Switching

Overview

The Stream Switcher is a tool for relaying one stream out of a number of sources towards one output with the ability to select the source to be relayed using a control tool such as the WebAdmin. The switch operation is seamless for the clients connected to the output of the switcher.

Basic setup

To set up the Stream Switcher using the WebAdmin, follow these steps:

1. Using a web browser, connect to the WebAdmin (see "WebAdmin" on page 17 for more details).

2. Select the **Switcher Settings** menu item. You will see a list of your configured switchers.

3. Click on **New Switcher** to create a new switcher. The following page is displayed:

4. Fill in the different fields you need to set up your Stream Switcher and then click the **Save Changes** button.

5. Your switcher will then be listed in the switcher list, followed by its status that can be either *Enabled* or *Disabled*.

Switcher Settings
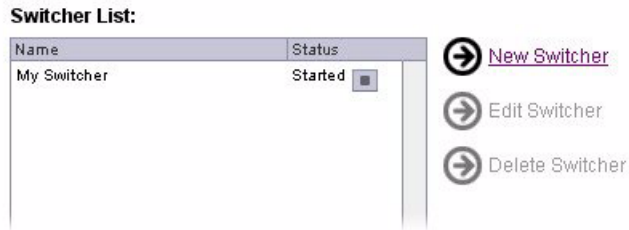
Switcher List:

Name	Status	
My Switcher	Started ■	→ New Switcher
		→ Edit Switcher
		→ Delete Switcher

NOTE   You can click the ▶ or ■ buttons to respectively start or stop the relay.

Configuration example

We will use the following network topology to demonstrate the configuration of a switcher.
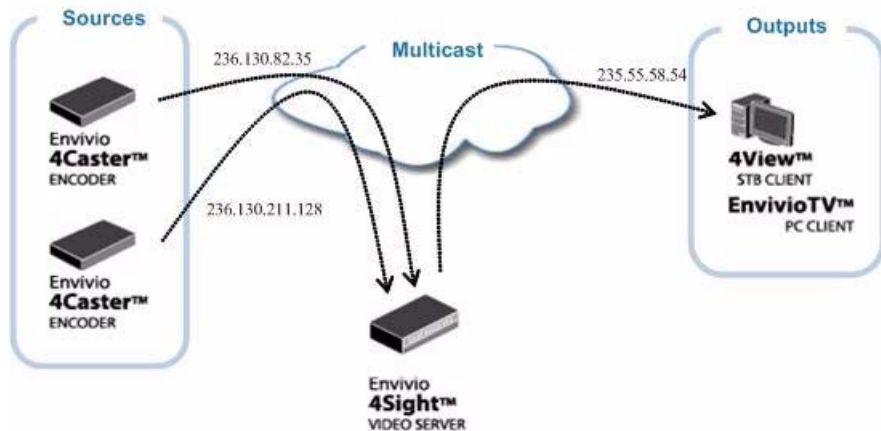
To setup this switcher, you need to follow these steps:

Version 5.0

1. Specify the output address (this address can be either multicast or unicast), the description, the local output interface, the UDP port list and the multicast TTL.

Output Settings

Description (optional):	Switcher Output
Output Hostname or IP Address:	235.55.58.54
UDP Port List:	6000, 6002
Multicast TTL:	15
Local Output Interface:	--- Use default interface --- ▾

You can define the network interface card that you want to use for sending packets by specifying the **Local Output Interface** parameter in the **Destination** section.

NOTE

The multicast time to live (TTL, see "Glossary" on page 109) is mandatory if you relay towards a multicast address, and it must be greater than 0. It can be omitted if relaying to a unicast address.

2. Provide the information about the input. Enter the hostname or IP address, then specify the ports you want to listen to (ports number can be separated by commas, semicolons or spaces).

Source Settings (1 of 1)

Description (optional):	My first source
Hostname or IP Address:	236.130.82.35
UDP Port List:	6850, 6852

⊕ Remove Source

Version 5.0

3. You can add more sources by clicking the **Add source** link.

Source Settings (1 of 2)

Description (optional): Switcher Source 1

Hostname or IP Address: 236.130.82.35

UDP Port List: 6850, 6852

→ Remove Source

Source Settings (2 of 2)

Description (optional): Switcher Source 2

Hostname or IP Address: 236.130.211.128

UDP Port List: 5048, 5050

→ Remove Source

4. Then click the **Save Changes** button.

WARNING   If you change an existing active switcher, when saving the modifications, the active source will switch to the first source declared in the switcher.

Connection to the output stream

To connect to the output stream of the switcher, you will need to provide the description file of the stream (SDP file). The easiest way to do it is to get the SDP file of one of the sources and modify it according to the address and port values that you set in the switcher.

Once the SDP file is modified, you can feed it directly to the client or put it in the movie folder of the 4Sight Video Server Streaming Server and access it using an RTSP request.

For further details on how to modify the SDP file, see "How to update an SDP file" on page 80.

Version 5.0

Status of the configured switchers

Once everything is configured, you can check that the source stream is effectively relayed by going to the **Switcher status** page.

The following window should be displayed:

You can switch to the other source by clicking the link **Switcher source 2**.

Connection to the output stream using an SDP file

To connect to the output stream of the switcher, you will need to provide the description file of the stream (SDP file). The easiest way to do it is to get the SDP file of one of the sources and modify it according to the output address and ports values that you set in the switcher.

Once the SDP file is modified, you can feed it directly to the client or put it in the movie folder of the 4Sight Streaming Server and access it using an RTSP request.

CHAPTER 4

Playlist Broadcaster

Overview

The Playlist Broadcaster is a tool for broadcasting a set of movies.

This section describes how to configure and start a playlist. It also describes simple use cases and applications of the Playlist Broadcaster.

A detailed description of the config file parameters is provided along with troubleshooting hints.

Basic setup

There are two ways to set up a playlist and broadcast it using the Playlist Broadcaster:

- using the WebAdmin,

- or by editing the playlist configuration files and running the Playlist Broadcaster in command line mode.

These two methods are not mutually exclusive and can be combined for advanced use of the Playlist Broadcaster.

Setting up a playlist using the WebAdmin

Using a web browser, connect to the WebAdmin (see "WebAdmin" on page 17 for more details). Click the **Playlists** menu item. You will see a list of your configured playlists. Click **New MPEG4 Movie Playlist** to create a new playlist. The following page is displayed:

Movie Playlist Details

Name:	
SDP File:	

Play Mode: Sequential ▾

Click and drag files from the left to the right to add to your playlist.

Available Movie Folders:
f:\Content\Plugfest ▾

Movie Folder Content	Order	Playlist Content
📁 Apple		
📁 Coding_Technologies		
📁 Envivio		
📁 FhG		

➔ Open Folder   ➔ Refresh Folder   ➔ Remove Item

Drag and drop the content you want to broadcast from the left panel to the right list.

Second part of the page lists the broadcast parameters

Fill in the different fields you need to set up your playlist and then click on the **Save Changes** button.

Example:

To broadcast the mp4profile.mp4 file in continuous loop mode, you should follow these steps:

1. Name the playlist: **myPlaylist**

2. Give a path to the SDP file: **myPlaylist.m4e**

3. In the **Play Mode** combo box, select **Sequential Looped**

4. Click and drag the mp4profile.mp4 file from the list of files to the Playlist items

5. Then click the **Save Changes** button

Your playlist will then appear in the list of available playlists.

You can then start and stop it directly by clicking on the play/stop icon on the right of the playlist.

If an error occurs, a red stop sign appears near the play icon. You can click on it to look at the log file.

You can test that your playlist is working by opening the SDP file in your player.

Restrictions

All the MPEG-4 files that you hint and include in your playlist must be ISMA-compliant and have exactly the same characteristics:

- contain one audio plus one video that have the same duration

- same video properties (video size, frames per second)

- same audio encoding parameters (frequency, stereo/mono).

NOTE You can put files encoded at different bit rate in your playlist only if the **sdp_reference_movie** or the first movie in the list has the highest bit rate.

If they do not have the same characteristics, EnvivioTV™ client will play the first stream fine and will probably stop whenever the video size, the number of streams, or any parameter is different in the next MPEG-4 file. This is due to the fact that the SDP file is generated only with regard to one MPEG-4 file (either the **sdp_reference_movie** or the first movie in the list).

CHAPTER 5

Load Balancer

Overview

The Load Balancer distributes traffic efficiently among video servers so that no individual server is overburdened. The master server redirects the player towards a randomly chosen slave server.
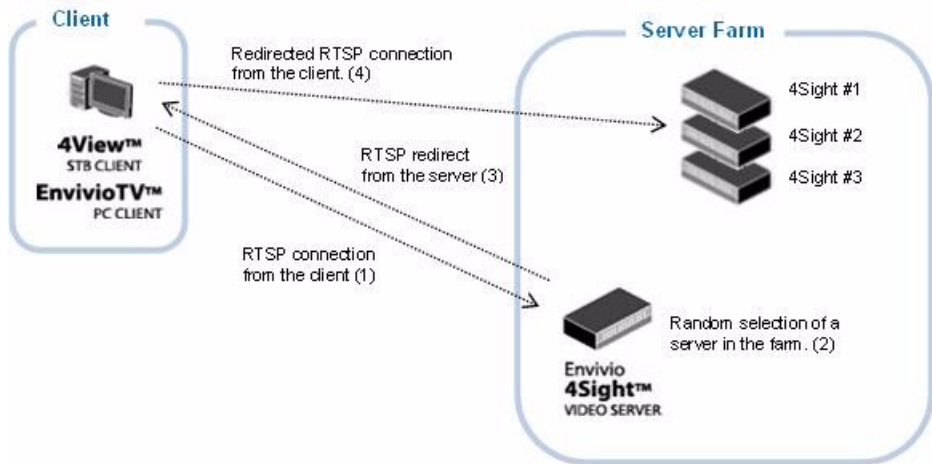
1. The player connects to the server.

2. The master server makes a random selection, and redirects the player to another server (4Sight #2).

NOTE The master server can also accept connections acting as a slave server.

Basic setup

Adding a slave server

To set up the load balancing using the WebAdmin, follow these steps:

1. Click the **Load Balancing** menu item. You will see a list of your configured servers.

2. Check the **Enable Load Balancing** check box to activate the load balancing mechanism on the server.

3. Click **New Slave server** to add a slave server to the list. The following page is displayed:

Slave server details

Slave server IP: 192.178.152.17:7080

Save Changes

4. Enter the IP address, followed by the port number of your slave server, then click the **Save Changes** button.

NOTE   If you do not specify the port number, the server will use the default streaming port.

Version 5.0

5. The slave server will then be listed, followed by its status (either
    Enabled or *Disabled)*.

Load Balancing Settings

Satus: ☑ Enable Load Balancing

Server Address	
192.167.156.23:7072	➡ New slave server
192.167.156.144:7070	➡ Delete slave server
192.178.152.17:7080	
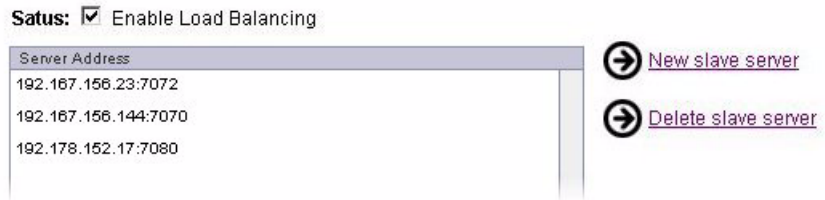
6. Click **Save Changes** to confirm your modifications.

Deleting a slave server

To remove a server from the server list, select the IP address of the server
you want to delete and click the **Delete Slave server** link. The server is
removed from the list after your confirmation.

NOTE   If you remove all the servers from the server list, the default server IP
address will be automatically added to the server list.
To remove the default server IP address, add a new address, then delete the
default server IP address.

Authentication and Authorization

Overview

4Sight provides authentication and authorization services. This is configured using three types of configuration files:

- Access file (for example, `qtaccess`)
- User file (for example, `qtusers`)
- Group file (for example, `qtgroups`)

and the `qtpasswd` executable.

Basic Setup

If you want the end user to be prompted for a user name and password when he/she tries to stream a content from the server, follow the steps described below:

1. Create a user file and/or a group file to define the users and groups allowed on your video server.

2. Create passwords for the new users using the qtpasswd utility.

3. Create an access file.

The content of each file and the use of the qtpasswd utility is described in the following sections.

Before going further, you should check that the two properties **modAccess_groupsfilepath** and **modAccess_userfilepath** are set correctly.

The **modAccess_qtaccessfilename** property specifies the name given to the access files. By default, access files are called qtaccess.

Here is the QTSSAccessModule section of the streamingserver.xml file:

```
<MODULE NAME="QTSSAccessModule" >

  <PREF NAME="modAccess_qtaccessfilename" >qtaccess </PREF>

  <PREF NAME="modAccess_groupsfilepath" >/usr/Envivio/4Sight/
  Config/qtgroups</PREF>

  <PREF NAME="modAccess_usersfilepath" > /usr/Envivio/4Sight/Config/
  qtusers</PREF>

  </MODULE>
```

The streamingserver.xml file also contains two parameters that are used for authentication and authorization:

- default_authorization_realm
  The text to display by default in the password dialog boxes

- authentication_scheme
  The default authentication scheme to be used by the server (basic, digest, or none)

User, Group, and Access files

By default, three files are provided as examples in **/usr/Envivio/4Sight/ Config**: qtusers, qtgroups and qtaccess. These files create a user with user name *admin*, password *admin* and group *admin*.

User file

The user file contains the list of users and their encrypted passwords. The first line of this file is the realm to be used. This file should not be modified manually (unless you want to modify the realm); use the qtpasswd utility to edit it.

This is an example of a user file with only one user called admin:

realm Envivio Streaming Server

admin:$dufr$.S5.....$DCUFFaEjQuyqcFbq9XXEk0:e7b400d4a5bb4e1020621ab
84d7c5ed1

NOTE   User names cannot be more than 255 characters long and must not include a colon.
Passwords cannot be more than 80 characters long.

Group file

The group file is a list of groups, each of which lists the users in that group.

To add, remove or modify a group, use a simple text editor and modify the group file.

The syntax of a line in the group file is:

groupname : <username> <username> ... <username>

Access file

An access file is a text file containing parameters to control access to a directory for a specific realm, a specific user or a specific group.

There is one access file per protected directory. If an access file is placed in a directory, all the subdirectories are protected in the same way, unless another access file is present in the subdirectory.

The following is a list of parameters accepted in the access file and their possible values.

Parameter name	Value
AuthName	This is the realm name. Realm name is displayed in the password dialog boxes.
AuthScheme	Can be either "basic" or "digest". • If basic is selected, a simple authentication is done with realm, user and password (base64 encoded). • If digest is selected, a MD5 based authentication system is used.
AuthUserFile	The user file to use in this directory.
AuthGroupFile	The group file to use in this directory.

Parameter name	Value
require user **require group**	A space separated list of user names or group names. This parameter specifies which users and/or groups are allowed to access the media in this folder. Two specific cases can be used: • require any-user any user is allowed • require valid-user any authenticated user is allowed
<Limit [READ|WRITE]>…</Limit>	Specifies access control for read and/or write privileges. If this parameter is not present or set, all parameters apply for read privileges. The parameters AuthName, AuthUserFile, AuthGroupFile and require can be defined between the two "Limit" tags.

NOTE

Parameter names and values are case sensitive. Respect the case for the following words:
`AuthName, AuthScheme, AuthUserFile, AuthGroupFile, require, Limit, basic, digest`

The qtpasswd utility

The qtpasswd utility lets you create a user file, and to add and delete users. To create a user, the name of the user is mandatory.

The syntax is the following:

```
qtpasswd -f filename [-F] [-c] [-r realm] [-p password] [-d] username
```

The following are optional flags that can be used:

[-f]	Password file to modify (defaults to the value set in streamingserver.xml)
[-F]	Does not confirm when deleting users or overwriting existing files
[-c]	Creates a new user file
[-r]	Realm to use when creating a new file with "-c" Default: "Envivio Streaming Server"
[-p]	Allows entry of the password at command line rather than prompting for it
[-P]	File to read the password from rather than prompting for it
[-d]	Deletes the specified user
[-h]	Displays usage

Configuration example

You can have multiple access files, user files, and group files. One way to configure your server is to have one movie folder associated to each realm.

For example, if you want to serve both Europe and US from your video server, follow the steps below:

1. Create two movie folders, one called `Movie_US`, and one called `Movie_Europe`.

2. Create a `qtaccess` file in each of those movie folders with the following parameters:

Parameter	Value for US	Value for Europe
AuthName	4Sight for US	4Sight for Europe
AuthUserFile	path_to_qtusers_US	path_to_qtusers_EU
require user	user_US1 user_US2 ... user_USN	user_EU1 user_EU2 ... user_EUN

3. Create the password file with a specific realm by running:
   qtpasswd -f _to_qtusers_US -c -r "4Sight US" user_US1
   qtpasswd -f _to_qtusers_EU -c -r "4Sight EU" user_EU1

4. Then create a password entry for all the other users by running:
   qtpasswd -f qtusers_US user_USx
   qtpasswd -f qtusers_EU user_EUx

Version 5.0

External authorization module

Overview

The Authorization module allows you to develop your own access control for the movies stored on the server.

By adding specific information into the URL, you will be able to apply your own dedicated authorization mechanism.

4Sight calls a function in the Authorization module for every incoming RTSP request:

bool ESSRequestAuthorization(char* inURL, char* inIPAddress, char* inQueryString);

where:

- inURL is the path to the movie – for example, if the request is **rtsp://192.178.155.10/tutorial1/file.mp4**, the URL is **tutorial1/ file.mp4**.

- inIPAddress is the IP address of the player requesting the movie,

- inQueryString is extra information from the player that is included in the URL after '**?**'. This information can include a user name and any specific information that you need. The field is NULL if nothing has been specified in the URL.

- The return value is true if you want to accept the RTSP request or false to refuse the RTSP request.

URL example:

  rtsp://192.178.155.10/tutorial1/file.mp4?user=julien&auth=123

See "Basic example of external authorization module" on page 68.

Installing the authorization module

This document explains how to install the external authorization module.

1. Stop 4Sight.

2. Create a new directory called plugins in your server directory.

3. Copy your authorization library into the plugins directory.

4. Open the server configuration file, **streamingserver.xml**.

5. Update the following lines in the **ESSExternalAuthModule** section:

```
<MODULE NAME="ESSExternalAuthModule">
<PREF NAME="enabled" TYPE="Bool16" >true</PREF>
<PREF NAME="auth_lib_file">path_to_your_library</PREF>
</MODULE>
```

6. Restart the server.

Version 5.0

Basic example of external authorization module

AuthModuleTestLib.h

```
extern "C"
{
  bool ESSRequestAuthorization(char* inURL, char* inIPAddress, char* inQueryString);
}
```

AuthModuleTestLib.cpp

```
#include "AuthModuleTestLib.h"
#include <stdio.h>
bool ESSRequestAuthorization(char* inURL, char* inIPAddress, char* inQueryString)
{
   char buf[256];
   bool authorize = false;
if(inQueryString != NULL) {
     sprintf(buf, "IP Address : %s\nRequested URL: %s\nQuery String: %s\nAuthorize ?",
                 inIPAddress, inURL, inQueryString);
   } else {
     sprintf(buf, "IP Address : %s\nRequested URL: %s\nAuthorize ?", inIPAddress, inURL);
}
   char c;
   int result;
   printf( "%s (y)es (n)o ?", buf);
   do {
     c =  getchar();
} while ( c!='y' && c!='n' );
   if ( c == 'y') {
     authorize =  true;
}
   else {
     authorize =  false;
}
   return authorize;
}
```

SNMP monitoring

Using SNMP monitoring

The 4Sight Video Server supports monitoring using the SNMP protocol. This makes 4Sight Video Server plugable into any of the network monitoring software like HP OpenView for example.

You can also use an SNMP browser like the one from MG-SOFT [1] to monitor one or more 4Sight Video Server at the same time. Open source solution like NET-SNMP [2] can also be used.

During the installation, all the necessary MIBs are copied to the **c:\Envivio\snmp\mibs** directory (or **/usr/Envivio/snmp/mibs** for Linux and MacOSX). You can find them there.

Two MIBs are necessary for monitoring a 4Sight Video Server using SNMP:

* ENVIVIO-COMMON-MIB.txt that is needed for all Envivio products

* ENVIVIO-4SIGHT-MIB.txt that is specific for 4Sight

---

1. MG-SOFT MIB Browser, http://www.mg-soft.com
2. NET SNMP, http://net-snmp.sourceforge.net

# Monitoring 4Sight using MG-SOFT MIB Browser

NOTE

This example is for windows only.

Once you have downloaded and installed the MIB Browser, follow these steps to compile the two MIBs mentioned above:

1. From the **Start>Program Files>MG-Soft MIB Browser** menu, select **MIB Compiler**.

2. Compile the MIB: go to the **Tools>Batch Compile menu**, select the **C:\Envivio\snmp\mibs** folder and click **OK**.

3. Click the **Save All button**, then click **OK**. This will save all the compiled MIBs in the correct folder for MIB Browser.

4. Exit the MIB compiler.

Your MIBs are now ready to be used in the MIB Browser.
Follow these steps to tell the MIB Browser which MIBs to use:

5. From the **Start>Program Files>MG-Soft MIB Browser** menu, select **MIB Browser**.

6. From the **MIB** tab, select the MIBs that you want to load from the bottom panel, then click the up arrow to load them.
You need to select at least **ENVIVIO-COMMON-MIB.txt** and **ENVIVIO-4SIGHT-MIB.txt**.
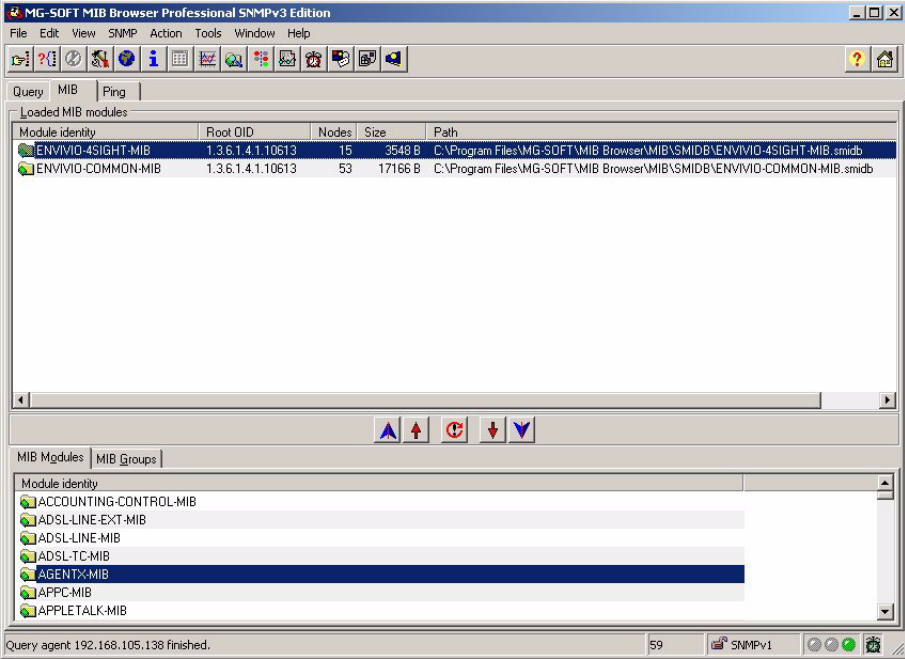
Your MIB Browser is now ready to be used with the 4Sight Video Server:

7. In the MIB Browser, return to the **Query** tab and enter the 4Sight IP address in the **Remote SNMP agent** field and select in the menu. Select **SNMP>Contact**, then expand the MIB tree to see the 4Sight Video Server.
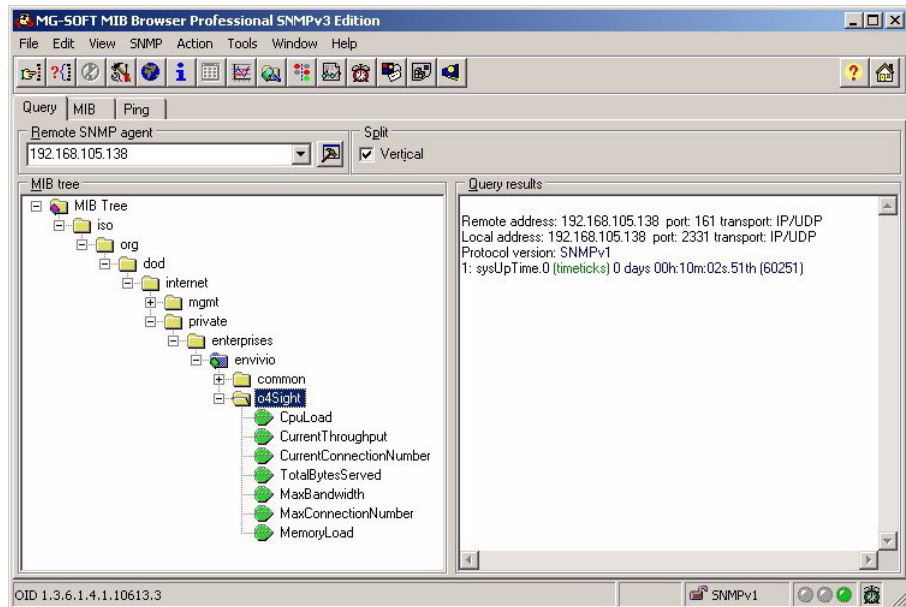
You can now query all the fields available in the 4Sight Video Server MIB. The following graphic gives the output of a "walk" query on the envivio node.
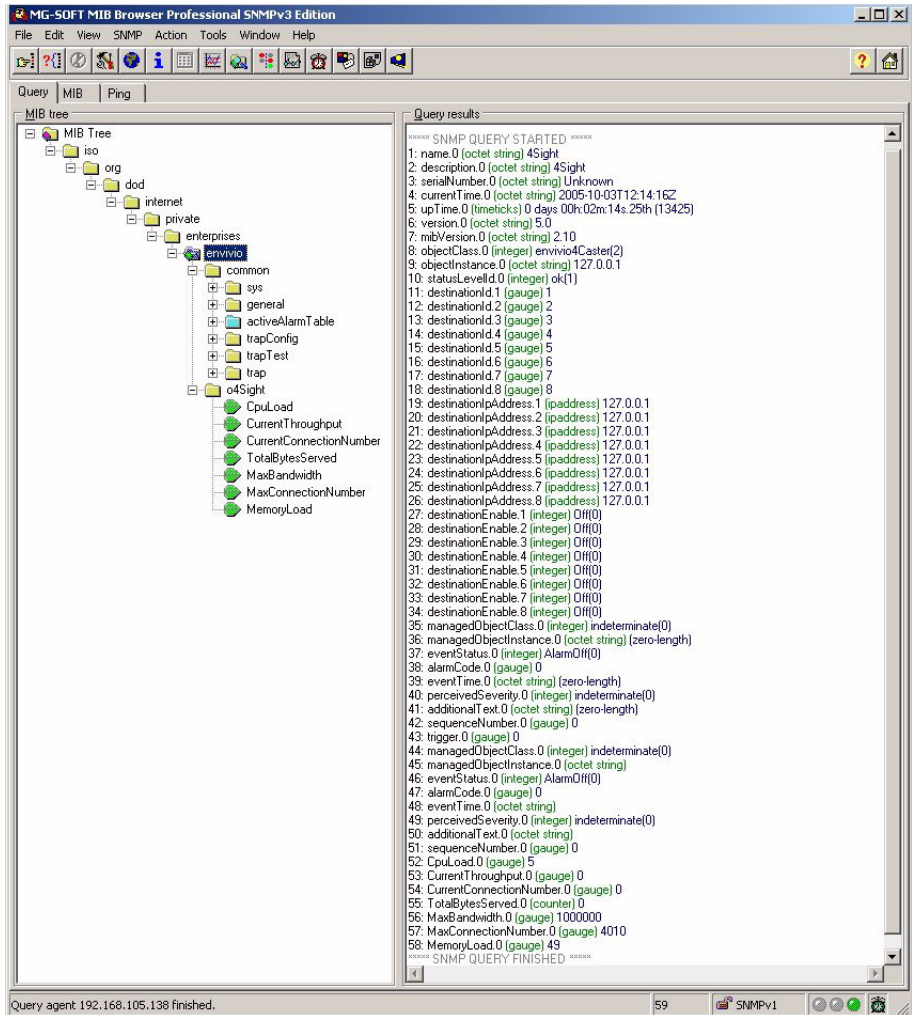
Enabling SNMP trap reception

The 4Sight Video Server uses traps to notify the SNMP monitor of events and alarms. To receive those traps, the monitor needs to be registered as a trap listener for 4sight. Each 4Sigth Video Server can send traps to up to ten trap listeners

Follow the steps below to register the MIB Browser (IP Address 192.168.105.32) as a trap receiver for 4Sight Video Server (IP Address 192.168.105.138):

1. In the MIB Browser look for the `TrapConfig` node in the common MIB.
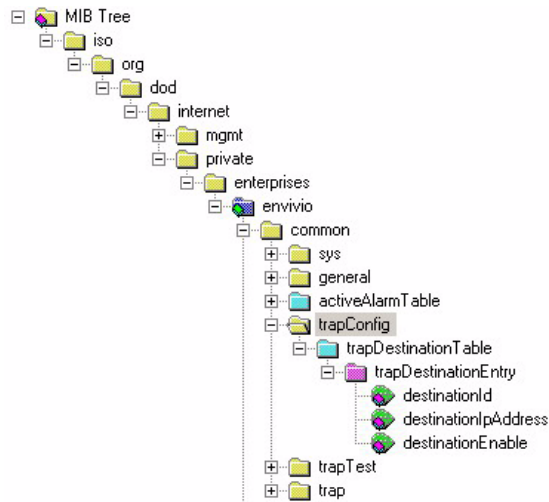
2. Right click on the **destinationIpAddress** node and select **Set**.
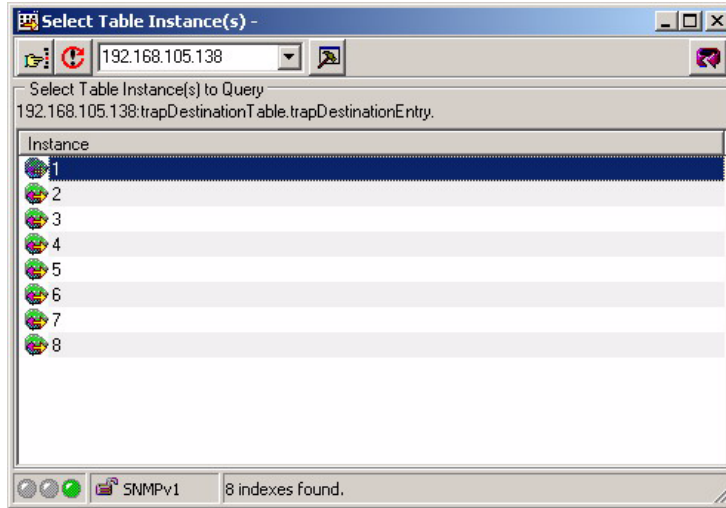
3. Double click on the first instance and specify the IP address of the MIB Browser
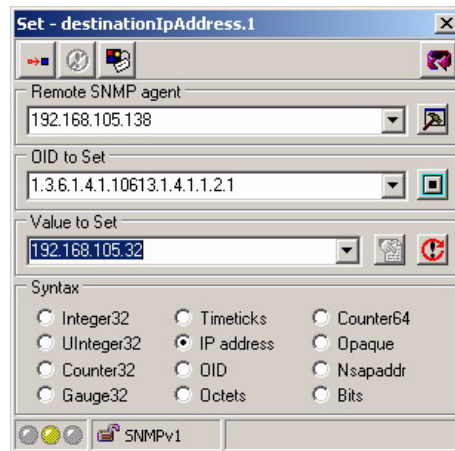
4. Repeat the operation for the first instance of the **destinationEnable** node and set it to *On*.
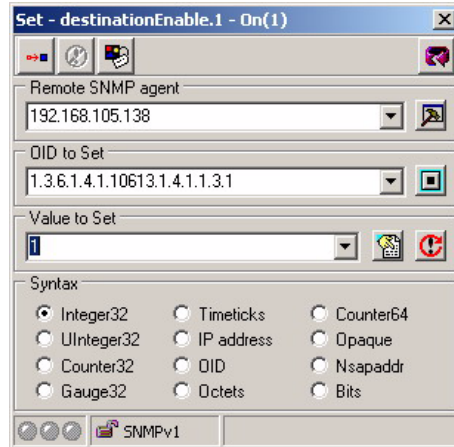
Your MIB Browser is now configured to receive the traps generated by the 4Sight Video Server.

The following graphic is an example of a "*Disk Full*" alarm.
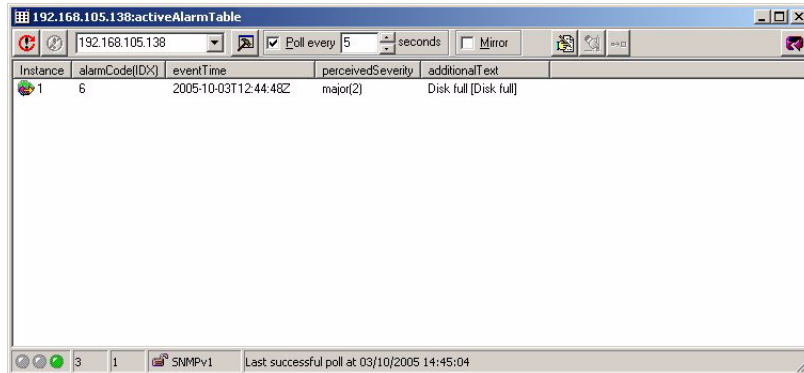
Appendix A

How-To

How to update an SDP file

In order to successfully open a multicast stream, you simply need to open the associated SDP file in a player. In the case of a relay or a switcher, you will need to update the information contained in the SDP file before opening it.

For example, let's say we are relaying multicast address 236.130.82.35, ports 6850 and 6852 towards address 235.55.58.54 ports 6002 and 6004. You will need to follow these steps to visualize the final stream:

1. Retrieve the SDP file created by one of the sources, for example, Envivio 4Caster.

2. Change the IP address (**c=**) and port (**m=**) to those defined in the output settings.

In our example, change:

*   **c=IN IP4 236.130.82.35/1** to **c=IN IP4 235.55.58.54** for the IP address,

*   **m=video 6850 RTP/AVP 96** to **m=video 6002 RTP/AVP 96**,

*   **m=audio 6852 RTP/AVP 97** to **m=audio 6004 RTP/AVP 97** for the port numbers and so on for each m= line.

How to link to a streamable content in a web page

You can link to an MPEG-4 streamed file directly into an HTML page. Opening the link will start an external player that will be used to read the file.

The two ways of doing that are listed below:

1. Use an anchor element pointing to an M4E file:

    `<a href="movie.m4e">Movie</a>`

    where **`movie.m4e`** contains: **`url=rtsp://server/movie.mp4`**

2. Or use an anchor with an RTSP URL:

    `<a href="e-rtsp://server/movie.mp4">Movie</a>`

How to embed a streamable content in a web page

When you embed a content in a web page, no external player is used and the content is played directly in the web page.

This can be done either with the Windows Media Player or with EnvivioTV ActiveX plugin:

- With the Windows Media Player

```
<object id="vidplayer"
    standby="Loading file in Windows Media Player..."
    classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95"
    codebase="http://activex.microsoft.com/activex/controls/mplayer/en/
    nsmp2inf.cab#Version=6,0,02,902"
    type="application/x-mplayer2">
    <param name="src" value="e-rtsp://server/movie.mp4">
    <param name="autostart" value="true">
</object>
```

- With the EnvivioTV ActiveX plugin

```
<object ID="ActiveX" CLASSID="CLSID:9272C6BB-41DD-4111-9B9E-
    91D48AE1F4D1" >
    <param name="url" value="e-rtsp://server/movie.mp4">
    <param name="refresh_rate" value="40">
</object>
```

You can display an MPEG-4 streamed file directly into an HTML page.

What is a hint track

To be streamed, an MPEG-4 file has to be hinted. The server uses hint tracks in an MPEG-4 file to packetize the media into RTP streams.

During the preparation of the MPEG-4 file for streaming, one hint track is added to the MPEG-4 file for each track whose media will be streamed over RTP. The hint tracks within the MPEG-4 file provide the information needed to turn the media into RTP packets.

These hint tracks are part of the MPEG-4 atom structure, and are similar to Apple QuickTime file format.

Can I use multicast over Internet?

Multicast will work on a LAN if multicast forwarding is enabled on the routers between the network segments (unless your router does not support multicast mode). However, this will not usually work across the Internet since most of the routers are not configured to support multicast, and you cannot be sure which routers will be used by your streams. Clients that are located behind routers, that do not implement multicasting can still receive a multicast by requesting the stream from a reflector.

The Video Server configuration file parameters

General parameters

The following are general parameters, valid across all modules of the server:

Keyword	Value
rtsp_port	Port(s) for accepting client connections NOTE: Add port 80 to this list if you are streaming across the Internet and want clients behind firewalls to be able to connect to the Video Server.
default_authorization_realm	The text to display as the login entity "realm" in the client is defined by default_authorization_realm. If the default realm is not defined here, then the default realm is set to "Streaming Server". The realm used in a response can also be dynamically set by authorizing modules.
authentication_scheme	Set this to be the authentication scheme you want the server to use. "basic", "digest", and "none" are the currently supported values.
do_report_http_connection_ip_address	When behind a DNS round robin, the client needs to be told the specific IP address of the server handling its request. This preference tells the server to report its IP address in the reply to the HTTP GET request when tunneling RTSP through HTTP.
tcp_seconds_to_buffer	The "scaling factor" to appropriately scale the buffer size. Using the bitrate of the movie as a guide, the server will set the TCP send buffer to fit this number of seconds of data.

max_tcp_buffer_size	Maximum size the send buffer can be set to.
min_tcp_buffer_size	Minimum size the send buffer can be set to.
error_logging	Either "true" or "false". This toggles error logging on and off.
screen_logging	If screen_logging is true, all error messages will be written to the terminal window. Note that in order to see these messages on screen, the server must be launched from the command-line, and must be launched in foreground mode (triggered by the addition of the -d flag).
logs_folder	Path to the folder containing the log files.
module_folder	Path to the folder containing dynamic loadable server modules.
pid_file	Path to the pid file.
safe_play_duration	If the server discovers it is serving more than its allowed maximum bandwidth (using the average bandwidth computation) it will attempt to disconnect the most recently connected clients until the average bandwidth drops to acceptable levels. However, it will not disconnect clients if they have been connected for longer than this time, in seconds. If this value is set to 0, it will never disconnect clients.
average_bandwidth_update	This is the interval in seconds between computations of the server's average bandwidth.
total_bytes_update	The interval in seconds between updates of the server's total bytes and current bandwidth statistics.

bind_ip_addr	This tells the server what IP address(es) it should accept client connections on. A value of 0 means all IP addresses currently enabled for the machine.
rtp_timeout	Amount of time in seconds the server will wait before disconnecting idle RTP clients. This timer is reset each time the server receives an RTCP status packet from the client.
maximum_bandwidth	Maximum amount of bandwidth (in kbps) the server is allowed to serve. If this limit is exceeded, new client requests for additional streams will receive the RTSP error 453, "Not Enough Bandwidth"
maximum_connections	Maximum number of concurrent connections allowed by the server.
real_rtsp_timeout	rtsp_timeout is the time-out that gets reported to the client. This is the actual time-out value.
rtsp_timeout	Amount of time in seconds the server will wait before disconnecting idle RTSP clients. This timer is reset each time the server receives a new RTSP request from the client.

Thinning parameters

The following preferences configure the server thinning algorithm. They are all in milliseconds.

Keyword	Value
drop_all_packets_delay	If a packet is only sent this far ahead of when the client needs to play it, it is dropped
drop_all_video_delay	If a video packet passes this delay, it is dropped.

always_thin_delay	If a packet passes this delay, thin some more.
thin_all_the_way_delay	If a packet is only sent this far ahead of when the client needs to play it, the server thins the stream as much as possible.
start_thinning_delay	If a packet is this late, we might thin some more.
start_thicking_delay	If the server is sending packets this far ahead of when the client needs them, it will attempt to add a little bandwidth to the stream
thick_all_the_way_delay	If a packet is this late, restores full quality.
quality_check_interval	The server checks packet delay differences at this interval.

More general parameters

Keyword	Value
append_source_addr_in_transport	If this is true, the server will add its source address to its transport headers. This is necessary on certain networks where the source address is not necessarily known.
alt_transport_src_ipaddr	The server appends its own IP address into the transport header. If you want an alternate address placed there, use this preference to specify the address.
send_interval	Minimum time in msec the server will wait between sending packet data to a client.
max_send_ahead_time	The server tries to get ahead of clients. This is the farthest ahead in seconds that the server can go.
overbuffer_rate	Rate at which to overbuffer the max_send_ahead_time.

Reliable UDP parameters

Keyword	Value
max_retransmit_delay	Maximum interval between when a retransmit is supposed to be sent and when it actually gets sent. Lower values means smoother flow but slower server performance.
small_window_size	If the client does not explicitly say what its window is, the server will use this as a default value. Specified in kilobytes.
medium_window_size	If the client does not explicitly say what its window is, the server will use this as a default value. Specified in kilobytes.
large_window_size	If the client does not explicitly say what its window is, the server will use this as a default value. Specified in kilobytes.
window_size_threshold	Bit rate (in kbps) at which the server switches to medium window size.
window_size_max_threshold	Bit rate (in kbps) at which the server switches to large window size.
rtcp_poll_interval	Time in msec between server checks for incoming RTCP packets. Longer interval means better server performance but less accurate estimates of packet round-trip-times.
rtcp_rcv_buf_size	Size in kilobytes for the RTCP UDP socket receive buffers. This needs to be large to absorb bursts of RTCP ACKs. Lower value may mean ACKs will get dumped by the kernel.
reliable_udp	Enable/disable reliable UDP protocol.

reliable_udp_dirs	If reliable UDP is enabled, specify the directories in which it is enabled. Directories should be relative to the movies folder, with a leading slash but no trailing slash. For example: /reliable_udp_dir
reliable_udp_slow_start	Enables or disables reliable UDP slow start. Disabling this may lead to an initial burst of packet loss due to the misestimation of the client's available bandwidth. Enabling this may lead to early thinning.

RTSP debugging parameters

Keyword	Value
RTSP_error_message	Enable debugging error messages in RTSP error response.
RTSP_server_info	Enable the server to send informations about itself.

Specific parameters

These parameters are module-specific.

Reflector module parameters

Keyword	Value
enforce_static_sdp_port_range	Turns on or off checking static ports in SDP files against a valid range. If an SDP file is played and it contains a port outside the static SDP range, then an invalid media error is returned to the client. Announced broadcasts will also receive the invalid media error from the server for setups on an invalid static port.
maximum_static_sdp_port	The maximum value for a static port. See enforce_static_sdp_port_range.
minimum_static_sdp_port	The minimum value for a static port. See enforce_static_sdp_port_range.

Access Log module parameters

Keyword	Value
enabled	Either "true" or "false". This toggles access logging on and off.
filename	Name of the access log file (default is 4SightAccess)
rotation_size	Maximum allowed size in bytes for the access log. A value of 0 means unlimited

| rotation_interval | Amount of time in days between rolls of the   access log. A value of 0 means unlimited. |
| use_gmt_time | Either "true" or "false". Log date time field in GMT if true and in local time if false |

Error Log modules parameters

Keyword	Value
enabled	Either "true" or "false". This toggles access logging on and off.
filename	Name of the access log file (default is 4SightAccess)
rotation_size	Maximum allowed size in bytes for the access log. A value of 0 means unlimited
rotation_interval	Amount of time in days between rolls of the access log. A value of 0 means unlimited.
verbosity	Verbosity of the logger (between 0 and 10)

Access module parameters

Keyword	Value
modAccess_qtaccessfilename	The file name for the Access module access file as it appears in controlled directories.
modAccess_groupsfilepath	The default path and file name for the Access module group list.
modAccess_usersfilepath	The default path and file name for the Access module user list.

WebAdmin parameters

Keyword	Value
RequestTimeIntervalMilli	The minimum delay (in milliseconds) between processing each admin request.
LocalAccessOnly	Only requests from 127.0.0.1 are allowed if LocalAccessOnly is true If set to false, then IP addresses as defined by the IPAccessList may make requests.
Authenticate	Require admin requests to authenticate with the server.
enable_remote_admin	Turn on or off remote admin request processing.
IPAccessList	A list of IP addresses that are allowed to make admin requests. LocalAccessOnly must be set to false before network addresses will be accepted. A wildcard of * is allowed for designating ranges. *.*.*.* means all addresses.
AdministratorGroup	The user has to belong to this group in order to be authorized by the admin module for admin requests.

File module parameters

Keyword	Value
sdp_url	This URL will be included in the dynamically generated SDP returned to the client. Typically, this should be set to your default web URL.
admin_email	This email address will be included in the dynamically generated SDP returned to the client.

max_allowed_speed	File module accepts the RTSP Speed header, which controls the overall play rate. This is the maximum rate it will allow.
max_allowed_scale	Maximum scale for fast forward and fast backward.
flow_control_probe_interval	When sending packets to the client, the connection may sometimes get flow controlled. When that happens, the server periodically checks to see if the condition has been lifted. This is how often it checks.
enable_shared_file_buffers **enable_private_file_buffers** **num_shared_buffer_increase_per_session** **shared_buffer_unit_k_size** **private_buffer_unit_k_size** **num_shared_buffer_units_per_buffer** **num_private_buffer_units_per_buffer** **max_shared_buffer_units_per_buffer** **max_private_buffer_units_per_buffer** **add_seconds_to_client_buffer_delay** **record_movie_file_sdp** **enable_movie_file_sdp**	Parameters used for file management.

Flow control module parameters

Keyword	Value
loss_thin_tolerance **num_losses_to_thin**	If a client reports loss percentages greater than loss_thin_tolerance, over the course of num_losses_to_thin consecutive RTCP (status) packets, the server will drop the bitrate of stream.

loss_thick_tolerance num_losses_to_thick	If a client reports loss percentages less than loss_thick_tolerance, over the course of `num_losses_to_thick` consecutive RTCP (status) packets, the server will increase the bitrate of the stream.
num_worses_to_thin	After this number of RTCP packets where the client is reporting degrading quality, the server will drop the bitrate of the stream.

Multiple movie folders parameters

Keyword	Value
movies_folders	List of extra movie folders.

Relay parameters

Keyword	Value
relay_prefs_file	Path to the relay preferences file.

Switcher parameters

Keyword	Value
switcher_prefs_file	Path to the stream switcher preferences file.
time_to_wait_for_synchro	After this amount of time (in ms), the switcher will try to force the switch.
switch_timeout	If switch did not occur by this time (in ms), it will be cancelled.

Licensing module parameters

Keyword	Value
license_file_dir	Path to the license file.

Load balancing module parameters

Keyword	Value
enabled	Enable/disable load balancing.
slave_server	List of server IP addresses for load balancing.

External authentication parameters

Keyword	Value
enabled	Enable/disable an external authentication library.
auth_lib_file	Path to the authentication library.

Recorder module

Keyword	Value
record_folder	Location of the recorder movies

The Playlist Broadcaster configuration file

Figure 1.   Playlist configuration file parameters

Keyword	Value
playlist_file	Playlist filename and path. The path can be absolute or relative.
play_mode	Three modes are available • sequential: The media is broadcast in the order it appears in the playlist file. When the last media file is done playing, the broadcast stops. • sequential_looped: The media is broadcast in the order it appears in the playlist file. When the last media file is done playing, the playlist repeats in the same order.
sdp_reference_movie	Reference movie file path and name. The absolute or relative pathname of the reference movie, which can be one of the media files you are broadcasting.
broadcast_name	Name of the playlist
destination_ip_address	The default value is the loopback address 127.0.0.1.You can also specify a unicast or multicast IP address
destination_base_port	The base port number for the broadcast must be an even number. If you do not specify a port number, the default port (5004) is used.
sdp_file	Path and name of the SDP file. The path can be absolute or relative path. This parameter is mandatory.
destination_sdp_file	Needs to be set to the same value as sdp_file.
announce_type	Specify the announce type for this playlist. Value can be "SAP", "RTSP" or "None". If not specified, value will be "None"
session_name	Name of the SAP session if announce_type is SAP.
broadcast_SDP_is_dynamic	Can be either "enabled" or "disabled".

Keyword	Value
broadcaster_name	User name if one needs to be provided to the destination address.
broadcaster_password	Password to be provided to the destination address.
multicast_ttl	Time to live of the multicast packets in the case of a multicast playlist.
pid_file	PID file of the broadcaster. This file is mandatory if you want to use the Playlist Broadcaster in both command line and WebAdmin.
logging	Can be either "enabled" or "disabled". If you enable logging, the Playlist Broadcaster records information about the broadcast in the log file, including error messages.
log_file	Path and name of the log file. If you enable logging but do not specify a file, then the Playlist Broadcaster creates a file with the name of the playlist followed by "." in the same directory as the playlist configuration file.
sap_session_name	In case of an SAP announce, use this session name
output_interface	Use this local interface to broadcast the playlist

The Web Admin configuration file

Table 1. WebAdmin streamingadminserver.conf parameters

Keyword	Value
port	The port on which WebAdmin listens for requests.
root	The root directory for all the HTML and CGI files. This path should be absolute.
plroot	The root directory for playlists managed by 4Sight. The path should be absolute.
host	The host name of 4Sight.
qtssIPAddress	The IP address of 4Sight to administrate.
qtssPort	The port of 4Sight to administrate.
qtssName	The name of 4Sight executable.
realm	The authorization realm for 4Sight.
log	Enable logging (1) or disable logging (0) for 4Sight.
Logfile	The full path of the WebAdmin log file.
logtime	Number of hours before which logs are rolled.
allow	Allow the following IP to access the server (optional).
deny	Deny access to the following IP (optional).

Video Server access log file

Field	Name	W3C or Custom	Example value
c-ip	IP address of client	W3C	157.100.200.300
date	Date of the access	W3C	11/16/1998
time	Time of the access (HH:MM:SS)	W3C	15:30:30
c-dns	Resolved dns of the client	W3C	fredj.ford.com
cs-uri-stem	Requested file	W3C	rtsp://saturn/mp4profile.mp4
c-starttime	Start time	W3C	0 [in seconds, no fractions]
x-duration	Duration of the session (s)	W3C	31 [in seconds, no fractions]
c-rate	Rate file was played by client	Custom	1 [1= play, -5=rewind, +5=fforward]
c-status	HTTP return code	Custom	200 [mapped to HTTP/RTSP status codes; 200 is success, 404 file not found,…]
c-playerid	unique player ID	Custom	[a GUID value]
c-playerversion	player version	Custom	3.0.0.1212
c-playerlanguage	player language	Custom	EN [two letter country code]
cs(User-Agent)	user agent	W3C	Mozilla/ 2.0+(compatible;+MSIE+3.0;+Windows 95) - this is a sample user-agent string
cs(Referer)	referring URL	W3C	http://www.gte.com
c-hostexe	host program	Custom	iexplore.exe [iexplore.exe, netscape.exe, dshow.exe, nsplay.exe, vb.exe, etc.]
c-hostexever	version	Custom	4.70.1215

c-os	os	Custom	Windows [Windows, Windows NT, Unix-[flavor], Mac-[flavor]]
c-osversion	os version	Custom	4.0.0.1212
c-cpu	cpu type	Custom	Pentium [486, Pentium, Alpha %d, Mac?, Unix?]
filelength	file length (seconds)	Custom	60 [in seconds, no fractions]
filesize	file size (bytes)	Custom	86000 [that is 86kbytes]
avgbandwidth		Custom	24300 [that is 24.3kbps]
protocol		Custom	MMS [mms, http]
transport		Custom	UDP [udp, tcp, or mc]
audiocodec		Custom	MPEG-Layer-3
videocodec		Custom	MPEG4
channelURL		Custom	http://server/channel.nsc
sc-bytes	bytes sent by server	W3C	30000 [30k bytes sent from the server to the client]
cs-bytes	bytes received by client	W3C	28000 [bytes received]
s-pkts-sent	packets sent	Custom	55
c-pkts-received	packets received	Custom	50
c-pkts-lost-client	packets lost	Custom	5
c-pkts-lost-net		Custom	2 [renamed from "erasures"; refers to packets lost at the network layer]
c-pkts-lost-cont-net		Custom	2 [continuous packets lost at the network layer]
c-resendreqs	packets resent	Custom	5
c-pkts-recovered-ECC	packets resent successfully	Custom	1 [this refers to packets recovered in the client layer]

c-pkts-recovered-resent		Custom	5 [this refers to packets recovered via UDP retransmission]
c-buffercount		Custom	1
c-totalbuffertime	seconds buffered	Custom	20 [in seconds]
c-quality	quality measurement	Custom	89 [in percent]
s-ip	server ip	W3C	155.12.1.234 [entered by the unicast server]
s-dns	server dns	W3C	foo.company.com
s-totalclients	total connections at time of access	Custom	201 [total clients]
s-cpu-util	cpu utilization at time of access	Custom	40 [in percent]
cs-uri-query		W3C	language=EN&rate=1&CPU= 486&protocol=MMS&transport =UDP&quality=89&avgbandwidt h =24300

Glossary

ADSL

Asymmetric Digital Subscriber Line.

bandwidth

Bandwidth has many meanings, depending on context. It started as a radio term and has been expanded to include other kinds of communications. It is the numerical difference between the highest and lowest frequencies of a radio band or channel.

Bandwidth means the same thing in audio, although in most applications, the lowest frequency is sufficiently low enough to be considered equal to zero. Hence, audio bandwidth usually means the highest frequency of an audio signal or the highest frequency that can be carried by an audio system.

BIFS

Binary Format for Scene. A set of nodes, based on VRML, that make it possible to combine 2D and 3D graphics, natural and synthetic sound, audio and video information, and stored and streamed material in one environment.

bit rate

The rate for transmitting data over a network. Envivio products use several bit rates. The channel capacity is the limit of the network. The maximum bit rate is a limit, less than the channel capacity, that the fitter uses to fit data into the actual bandwidth for a streaming server. The audio encoder uses an average bit rate and a maximum bit rate for variable bit rate encoding.

broadband

A general term for DSL, cable modem, and T1 lines.

codec

The software that encodes and decodes a file. The word is a shortened form of <u>co</u>der-<u>dec</u>oder or <u>co</u>mpression-<u>dec</u>ompression. This product has an audio codec and a video codec.

FTP

File Transfer Protocol

hinted movies

Movies that can be streamed for broadcast over the Internet. A hint track has information about the media file so that it can be streamed properly.

HTTP

Hypertext Transfer Protocol

I-frames

An MPEG-4 term for key frames.

IANA

Internet Assigned Numbers Authority

Internet

An interconnected system of networks that connects computers around the world via the TCP/IP protocol.

IOD

Initial Object Descriptor.

ISMA

Internet Streaming Media Alliance.

ISO

International Standards Organization.

kbps

Kilobits per second.

key frame

A frame that fully refreshes its contents and that can be used as a reference point for other frames in the sequence. In MPEG terms, this is an I-frame.

M4E

Envivio extension for SDP files. Allows to open SDP files using directly Envivio plug-in.

MIB

Management Information Base.

MPEG

As defined on the MPEG website, mpeg (pronounced M-peg), stands for Moving Picture Experts Group, and is the name given to a family of International Standards used for coding audio-visual information in a digital compressed format. The MPEG family of standards includes MPEG-1, MPEG-2, and MPEG-4, which is formally known as ISO/IEC-11172, ISO/IEC-13818 and ISO/IEC-14496.

multicast

Process where a single stream is served from one server to multiple clients. It is a one-to-many communication.

multimedia

As the name suggests, one presentation with multiple forms of media—text, graphics, sound, video, and animation. MPEG-4 gives

broadcasters the tools to deliver multimedia presentations over the Web.

Real-time

Rapid transmission and processing of event-oriented data and transactions as they occur instead of being stored and retransmitted or processed as batches.

realm

Authentication context.

RTCP

Real Time Control Protocol.

RTP

Real-time Transport Protocol.

RTSP

Real Time Streaming Protocol.

SAP

Session Announcement Protocol. SAP is a protocol to announce multicast broadcast sessions. A broadcast session is announced by periodically multicasting an SDP announcement packet to a common multicast address and port.

scene description

A description of the media in a presentation and how they are to be reconstructed. The technical term is BIFS (Binary Information for Scene).

SDP

Session Description Protocol.

Smoother

A process that modifies the send times of media packets to fit the transmission channel and makes the transmission information available to other processes, such as the user interface.

SNMP

Simple Network Management Protocol.

SSRC

Synchronization SouRCe.

stream

The encoded media or scene description data in an MPEG-4 file.

streaming

Sending media streams from a streaming server to a player as a steady, continuous stream. The player reconstructs media scenes using information in the scene description.

TCP

Transmission Control Protocol.

TTL

TTL stands for Time To Live. A value in the range 0 through 255 defines the scope within which multicast packets should be sent over a network using Internet Protocol (IP). Each router decrements the TTL by one. When the value reaches a predefined lower limit, the router throws the packet away. By default the TTL is set to 1.

UDP

User Datagram Protocol.

Unicast

Unicast is communication between a single sender and a single receiver over a network.

URL

Uniform Resource Locator. A method of naming documents or places on the Internet.

Index