

envivio®

4Sight Administrator Guide

Software Edition

Version 3.1 rev.0

December 7, 2004

4SI310ADM-SE-041207 -00



The leading provider of MPEG-4 broadcast and streaming solutions

© 2004 Envivio Inc. All rights reserved.

4Sight Administrator Guide 3.1

This manual and the software described in it are furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is provided for information only. It is subject to change without notice and should not be construed as a commitment by Envivio Inc. Envivio Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Envivio.com Inc. Please remember that existing media that you may want to include in your project may be protected under copyright law. Incorporating such material into your new work could violate the rights of the copyright owner. Please be sure to obtain any required permissions from the copyright owner.

Envivio and the Envivio logo are registered trademarks of Envivio Inc. EnvivioTV, 4Caster, 4Coder, 4Sight, 4Manager, 4Sight and 4View are trademarks of Envivio Inc.

Microsoft, Windows, Windows 2000, Windows XP, and Windows NT are registered trademarks of Microsoft Corporation in the U.S and/or other countries. Apple, Macintosh, Power Macintosh, QuickTime, and TrueType are trademarks of Apple Computer, Inc. registered in the U.S. and other countries. QuickTime and the QuickTime logo are trademarks used under license. Helvetica and Times are trademarks of Linotype-Hell AG and/or its subsidiaries. Pentium is a registered trademark of Intel Corporation. Flash is a trademark of Macromedia, Inc. Sun is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Linux is a registered trademark of Linus Torvalds. IRIX is a registered trademark of Silicon Graphics, Inc. All other trademarks are the property of their respective owners.

Protected by U.S. Patents [patents pending]. Envivio Incorporated, 400 Oyster Point Blvd., Suite 111, South San Francisco, California 94080 USA.

Notice to U.S. government end users. The software and documentation are "commercial items," as that term is defined at 48 C.F.R. §2.101, consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the commercial computer software and commercial computer software documentation are being licensed to U.S. government end users (A) only as commercial items and (B) with only those rights as are granted to all other end users pursuant to the terms and conditions set forth in the Envivio Incorporated standard commercial agreement for this software. Unpublished rights are reserved under the copyright laws of the United States.

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	11
	User Guide structure	12
	Envivio Professional Services and Support	13
Chapter 2	Overview	15
	Specifications	16
	Supporting Standards and Platforms	16
	Supported file formats	17
	Monitoring and Administration	17
	Performance	17
	Error Resilience	18
	Concepts	19
	Streaming overview	19
	Streaming session	20
	Unicast streaming	20
	Multicast streaming	21
	Relay	23
	Broadcast	23
	Hint tracks	23
Chapter 3	Installation	25
	System Requirements	26
	Installation of 4Sight	27
	Installation on Windows	27
	Registration	27
	Installing 4Sight	28
	Configuration	29

	Uninstallation under Windows.....	29
	Installation on Linux.....	30
	Uninstallation under Linux.....	31
	Installing the Envivio Linux Kernel.....	32
	Installing the Linux kernel patch.....	32
	Uninstalling the Linux kernel patch.....	32
Chapter 4	Quick start.....	33
	Overview.....	34
	Starting 4Sight.....	34
	Streaming your movie.....	34
	Obtaining the status of 4Sight.....	34
	Stopping 4Sight.....	35
Chapter 5	Streaming Server.....	37
	Setting up Movie Folders.....	38
	Using 4Sight as a reflector.....	40
	Using 4Sight as a relay.....	42
	Basic setup.....	42
	Preliminary setup of the server.....	42
	Setting up a relay using the WebAdmin.....	43
	Common configurations.....	44
	Relay a UDP stream to one or more destinations.....	44
	Relay an announced stream.....	46
	Relay a requested stream.....	47
	Relay a UDP stream using a specific network interface card..	48
	Status of the configured relays.....	49
	Hints.....	49
	Passwords.....	49

	Relay verification.....	49
	Running 4Sight as a service	50
	Installing 4Sight as a service	50
	Starting, stopping and removing the 4Sight service	51
	Embedding streamable content in a web page	52
Chapter 6	Playlist Broadcaster	53
	Overview.....	54
	Basic setup	55
	Setting up a playlist using the WebAdmin.....	56
	Setting up a playlist using the command line interface	59
	Common scenarios: Broadcasting a playlist	61
	Single server configuration	61
	Multiple servers: Unicast configuration.....	62
	Multiple servers: Unicast and announce.....	63
	Broadcast to a multicast address.....	64
	Configuration file parameters	65
	Command line parameters.....	68
	Troubleshooting	69
	The Playlist Broadcaster and the WebAdmin.....	69
	Playlist configuration file	69
	Playlist files	70
	Command line mode.....	70
Chapter 7	WebAdmin	71
	Overview.....	72
	Starting the WebAdmin	72
	Stopping the WebAdmin	72
	Obtaining the status of the WebAdmin	72
	Configuration parameters.....	73

Chapter 8	Stream Switcher	75
	Overview.....	76
	Basic setup	77
	Using a specific network interface card as output	78
	Configuration example	79
	Accessing the stream generated by the switcher.....	81
	Viewing the active source	82
	Modifying an SDP file.....	82
	Status of the configured switchers	84
Chapter 9	Load Balancer	85
	Overview.....	86
	Basic setup	87
	Adding a slave server.....	87
	Deleting a slave server	88
Chapter 10	Authentication and Authorization	89
	Overview.....	90
	Basic Setup.....	91
	User, Group, and Access files	93
	User file.....	93
	Group file.....	93
	Access file.....	94
	The qtpasswd utility	96
	Common scenario	97
	Restrict access to the WebAdmin.....	98
	Using the WebAdmin configuration file	98
	Using the WebAdmin user interface.....	98
	Changing the WebAdmin default user name and password.....	99
	Troubleshooting.....	101

	Access file.....	101
	qtpasswd.....	101
	Remote access to the WebAdmin.....	101
Chapter 11	External authorization module.....	103
	Overview.....	104
	Installing the authorization module.....	105
	Basic example of external authorization module.....	106
Appendix A	Description of the configuration file parameters ...	107
	General parameters.....	108
	Thinning parameters.....	111
	More general parameters.....	112
	Reliable UDP parameters.....	113
	RTSP debugging parameters.....	114
	LCD panel.....	114
	Specific parameters.....	115
	Reflector parameters.....	115
	Access log file parameters.....	116
	Access module parameters.....	116
	WebAdmin parameters.....	117
	File module parameters.....	118
	Flow control module parameters.....	119
	Multiple movie folders parameters.....	119
	Relay parameters.....	119
	Switcher parameters.....	120
	Licensing module parameters.....	120
	Load balancing module parameters.....	120
	External authentication parameters.....	120

Appendix B	Configuration examples	121
	Overview	122
	Configuration examples	123
	Example #1	123
	Example #2	123
	Example #3	125
Appendix C	Streaming Server access log file	127
Glossary	131
Index	137

CHAPTER I

Introduction

4Sight streams MPEG-4 media over IP networks. MPEG-4 is a standard that defines how to encode and compose media elements. Envivio products are based on this technology, providing a comprehensive toolset for encoding media to the MPEG-4 format, and streaming it over IP networks.

User Guide structure

This document is divided as follows:

- Chapter 2 introduces the main features of 4Sight and some basic concepts of MPEG-4 streaming.
- Chapter 3 gives system requirements and describes how to install 4Sight (for Windows and Linux).
- Chapter 4 explains how to run 4Sight and stream movies.
- Chapter 5 describes specific concepts of the streaming server itself: Setting a movie folder, SNMP monitoring, reflecting, and relaying.
- Chapter 6 describes the Playlist Broadcaster, a tool for broadcasting a set of movies.
- Chapter 7 describes the WebAdmin, a web-based interface for monitoring 4Sight.
- Chapter 8 describes the Stream Switcher, a tool for relaying one stream and then switching to another stream.
- Chapter 9 describes the Load Balancer, a tool for distributing traffic efficiently among streaming servers.
- Chapter 10 gives details about authentication and authorization.
- Chapter 11 describes the external authorization module.
- Appendix A details all the configuration parameters.
- Appendix B shows configuration examples combining 4Sight and Envivio 4Caster.

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.

CHAPTER 2

Overview

Specifications

4Sight is an MPEG-4 compliant server that broadcasters and cable operators can use to manage and distribute interactive multimedia content through IP networks to EnvivioTV™ or any other MPEG-4 compliant player.

4Sight consists of three components:

- The Streaming Server for video on demand, relaying, and reflecting.
- The Playlist Broadcaster that can broadcast a list of MPEG-4 files, which can then be reflected by the streaming server.
- The WebAdmin for web-based administration and monitoring.

Supporting Standards and Platforms

4Sight 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.

4Sight sends media to the network using the IETF defined suite of protocols based on IP for real-time streaming, and supports HTTP tunneling to support firewall configurations. It also uses the associated IETF recommendations dedicated to MPEG-4 transport over IP.

4Sight supports unicast streaming for media on demand, and multicast streaming for media broadcasts. It also includes a playlist manager to broadcast a list of MPEG-4 files. Multiple streaming redirection configurations allow flexible network setup.

4Sight supports trick play (fast forward and fast backward) based on the RTSP RFC 2326 by using play command with a specific SCALE attribute. The range of speeds allowed is by default [-128, -1] for fast backward and [1, 128] for fast forward. You can change them via the streaming server configuration file (see “File module parameters” on page 118).

NOTE

The fast forward/fast backward rendering depends on the way the file is encoded: 4Sight sends only key frames when fast forwarding. The playback quality is related to the frequency of key frames. It is increased when there are more key frames.

4Sight is available on Windows and Linux.

Supported file formats

4Sight streams MPEG-4 files (provided that the files are hinted).

Monitoring and Administration

4Sight allows network operators to dynamically monitor streaming through a web-based interface, either online during operation or offline with a log file mechanism.

An SNMP interface is also provided for integration into existing supervision systems.

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.

Concepts

Streaming overview

The streaming process includes two parts: a **client** and a **server**. The server sends an MPEG-4 formatted movie to a client over a network. Unlike the progressive download or the simple file transfer (where the client waits for the whole movie to be downloaded before playing anything), the server does not send the movie as a whole, it divides it into small packets that can easily be sent over a network. Then, the role of the receiver (the player or the client) is to reassemble them and play the movie. The set of these streamed small packets is called a **stream**.

The streaming protocol used is RTP (Real Time Protocol). RTP is usually used to stream the packets in real time.

There are several ways to transmit RTP packets over the Internet:

- using TCP (Transmission Control Protocol)
- using UDP (User Datagram Protocol) or reliable UDP
- using HTTP tunneling (Hypertext Transfer Protocol).

The command protocol that allows the client to communicate with the server is RTSP (Real-time Streaming Protocol). This protocol lets the client decide what it wants to do (for example, begin the session, play the clip, pause, or close).

The streams can be sent in two different ways:

- unicast - one to one (see “Unicast streaming” on page 20)
- or multicast - one to many (see “Multicast streaming” on page 21)

Streaming session

A streaming session is a set of streams (such as video, audio or BIFS for MPEG-4) that is sent from a server to a client.

A stream can contain a live track from a broadcast MPEG-4 movie (for example, audio or video generated by the Playlist Broadcaster or Envivio 4Caster), or from a stored MPEG-4 file.

1. To begin a streaming session, the client sends a request for a movie.
2. To start the session, the server first gives a full description of the movie using SDP (Session Description Protocol), including:
 - the types of media,
 - the number of streams,
 - the codec to be used for each stream.
3. The server then sends each media in a separate stream of RTP packets.

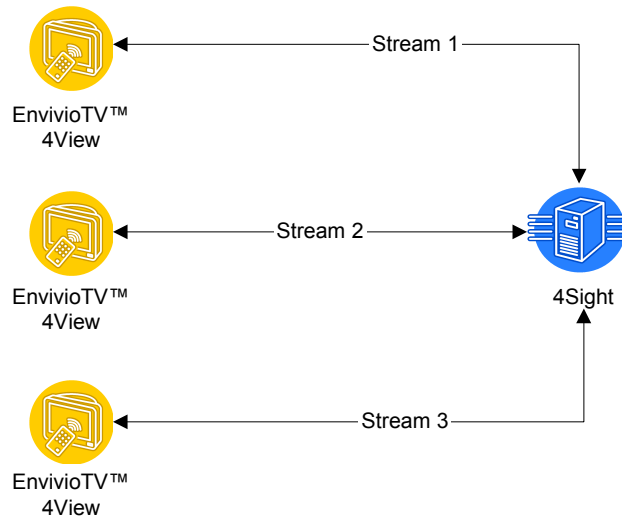
Unicast streaming

Unicast means communication between a single server (sender) and a single client (receiver) over a network. Unicast is contrasted with multicast (a single server, several clients).

1. For real time streaming in a unicast session, the client sends a request for a movie to the server using RTSP.
2. Using the same protocol, the server sends a description of the movie to the client.
3. Then the client sets up the session, and the server begins to send the streams to the client.

In the case of unicast streamed stored content, the user has the ability to jump anywhere in the movie (the server simply begins streaming the movie from this new point).

Figure 1. Unicast streaming



Multicast streaming

Multicast streaming means communication between one sender and multiple receivers. During such a session, one copy of each stream is sent over each branch of a network. The main advantage is that this reduces the network traffic necessary to send data to large numbers of clients. The main disadvantage is that the user has no control over the requested movies to navigate forward or backward.

The session process is the following:

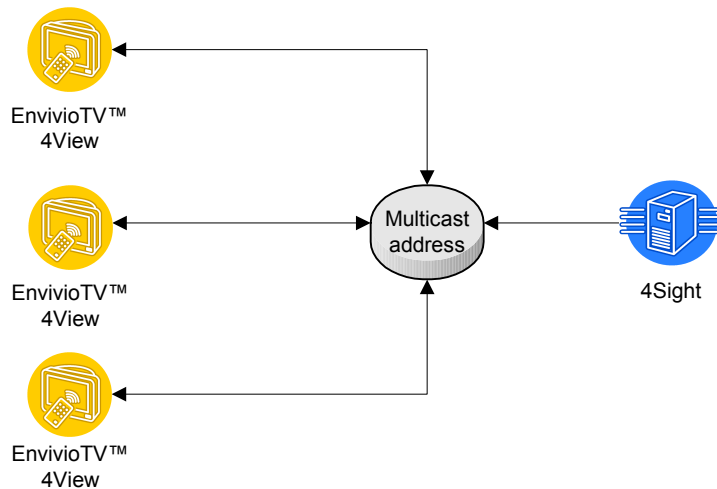
1. The server (or broadcaster) broadcasts a stored content or a live broadcast to a multicast address.
2. Clients join the multicast session by opening an SDP file, by accepting SDP broadcast, or by joining SAP session. The SDP describes the number of streams that need to be opened, the multicast address, the port numbers on which each stream is sent, and additional data such as transport information.

NOTE

Since multicast is one of the packet types in Internet Protocol Version 6 (IPv6), it 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. Clients that are located behind routers, that do not implement multicasting can still receive a multicast by requesting the stream from a reflector.

A **reflector** is a server that joins a multicast and converts it into several unicasts, one for each requesting client (see “Using 4Sight as a reflector” on page 40 for more information).

Figure 2. Multicast



NOTE

To multicast content, broadcast it to any address between the following values: 224.0.0.0 - 239.255.255.255
The main advantage of multicast mode is that less bandwidth is used compared to unicast mode.

Relay

4Sight acts as a relay server when it accepts a stream from a source (a broadcaster or another streaming server) and sends it or "relays" it to one or more other IP addresses. See "Using 4Sight as a relay" on page 42.

Broadcast

A **broadcast** is a stream of RTP packets generated by an application or process external to the streaming server (it often runs on a separate machine).

The Playlist Broadcaster reads the MPEG-4 files in a playlist and converts them into RTP packets. It sends the packets over UDP/IP, to a specific address bound to a defined port. This address can be either unicast or multicast.

The Playlist Broadcaster creates a file containing all the SDP information about the presentation needed by the client (and possibly reflector). The Playlist Broadcaster can broadcast the SDP to port 9875 of the destination address, or can send the SDP to the SAP address. The Playlist Broadcaster creates a standard SDP file named filename.m4e conforming to the SDP standard. See "M4E file" on page 130 for a sample of M4E.

Examples of broadcasters:

- Playlist Broadcaster
(included with 4Sight)
- Envivio 4Caster

Hint tracks

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's QuickTime file format.

CHAPTER 3

Installation

Version 3.1

System Requirements

The minimal machine configuration to run 4Sight is the following:

- Windows® 2000/2003 advanced server, Red Hat Linux® 8.0
- For Linux/Windows: Pentium® III 800 MHz with 256 MB RAM (512 recommended), SCSI architecture, Gigabit Ethernet Card.

NOTES

On a multiprocessor architecture, 4Sight only uses one processor.

4Sight is not optimized for hyperthreading. We recommend deactivating hyperthreading to fully benefit from the power of the CPU.

Installation of 4Sight

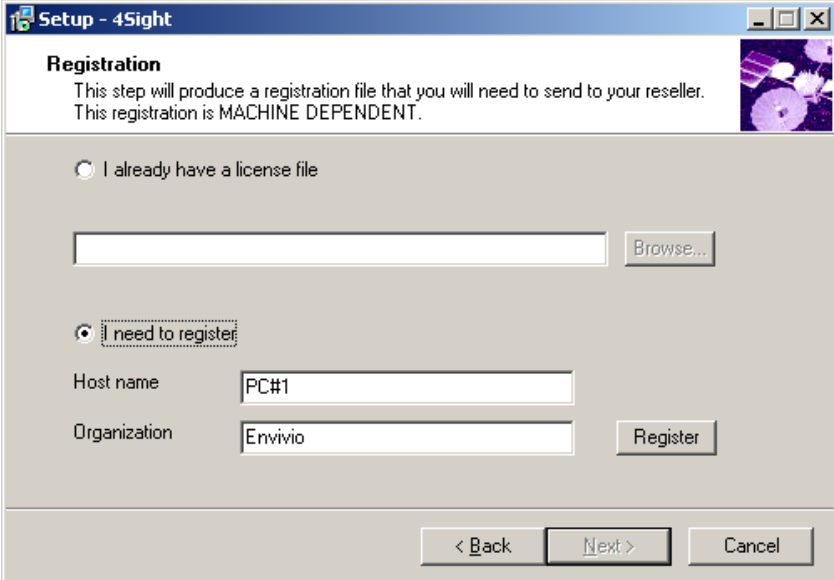
4Sight is available on Windows and Linux. Both versions are provided with an installer.

Installation requires 2 steps: create a registration file which will be used to generate a licence key and install with separately provided licensing file.

Installation on Windows

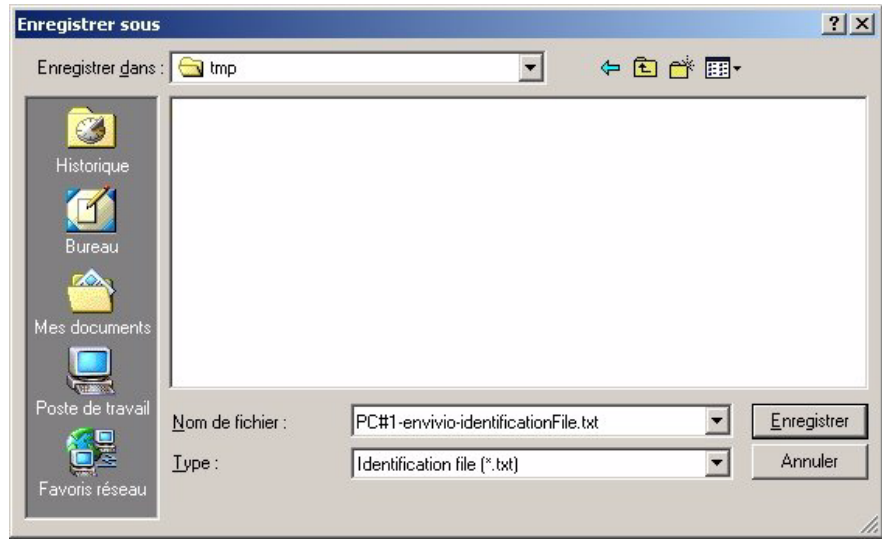
Registration

1. Double-click the installer file 4Sight-XXX.exe (where XXX is the version number), then read and follow the online instructions given during the installation process.
2. Select the I need to register box and complete the registration form with your host name and organization. Then click the Register button.



The screenshot shows the 'Setup - 4Sight' registration dialog box. The title bar reads 'Setup - 4Sight'. The main heading is 'Registration'. Below the heading, it states: 'This step will produce a registration file that you will need to send to your reseller. This registration is MACHINE DEPENDENT.' There are two radio button options: 'I already have a license file' (unselected) and 'I need to register' (selected). Below the first option is a text input field and a 'Browse...' button. Below the second option are two text input fields: 'Host name' with the value 'PC#1' and 'Organization' with the value 'Envivio'. A 'Register' button is located to the right of the 'Organization' field. At the bottom of the dialog are three buttons: '< Back', 'Next >', and 'Cancel'.

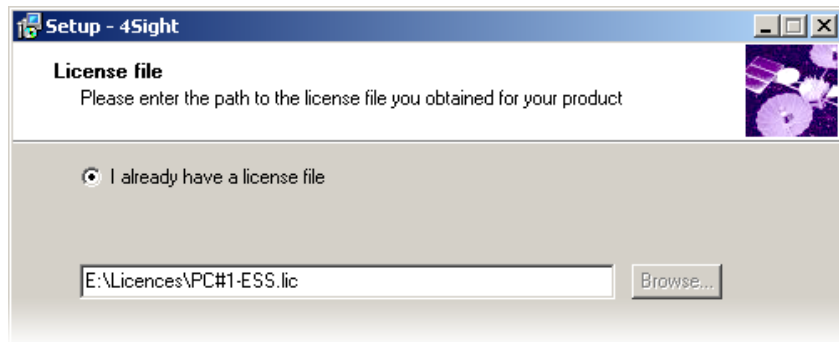
3. This creates a registration file that you have to save on your disk before mailing it to your sales contact.



4. When you receive the ZIP file, uncompress it and save the licensing file (LIC extension) into a temporary folder.

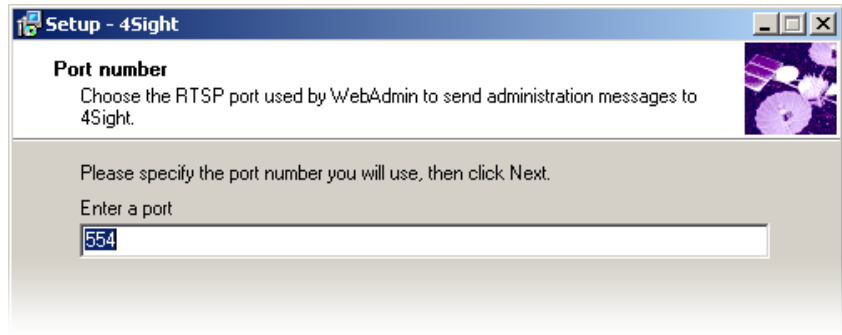
Installing 4Sight

5. Relaunch the installer, indicate the path to the license file. Then follow the installation instructions.



Configuration

6. You will have to choose an administration port, by default port 554 is used.



NOTE

By default, the movie directory is set to
`c:\Program Files\Envivio\4Sight XXX\Movies.`

Uninstallation under Windows

To uninstall 4Sight, stop the service (if 4Sight was started as a service), and select the Uninstall 4Sight 3.1 command from the Start>Program Files>Envivio>Envivio 4Sight 3.1 menu.

Installation on Linux

1. To install 4Sight on Linux, enter the following command:
rpm -ivh 4SightSE-XXX.i386.rpm
where **XXX** is the version number.
2. To create the license registration file, run the following command:
/etc/rc.d/init.d/ess register
3. You will be asked for the path to the license file. If you are not registered, enter y (yes) and give your identifier (your host name) and company name. This will create a registration file that you will have to email to your sales contact.

The following window should be displayed:

```
=====
The license file for the 4Sight must be placed in /usr/Envivio/4Sight/sbin/
If you do not have a license file, you need to register in order to obtain one.
Do you want to register now? (y/n) y

The registration process is done on a per machine basis.
You can use the host name to identify the license file for this particular machine.

Please enter an identifier for this machine (default is ESSserver)
> ESSserver
"ESSserver" will be used as an identifier for this license.

Please enter the name of your company
> envivio

A registration file has been created. (/usr/Envivio/4Sight/sbin/ESSserver-license.id)
Please send this file to your contact at Envivio to obtain a license for the 4Sight.

[root@ESSserver sbin]#
```

4. When you receive the license file (LIC extension) copy it to the
/usr/Envivio/4Sight/sbin directory.
5. Restart the server using the following command:
/etc/rc.d/init.d/ess start

The following executables will be installed in `/usr/Envivio/4Sight/sbin`:

- `4Sight`: 4Sight server binary
- `EnvivioPlaylistBroadcaster`: Playlist Broadcaster binary
- `streamingadminserver.pl`: Admin Server startup script
- `qtpasswd`: Password utility

The following startup scripts will be installed in `/usr/bin`:

- `envivio_server.sh` (obsolete, use scripts in `/etc/rc.d/init.d`)
- `envivio_admin.sh` (obsolete, use scripts in `/etc/rc.d/init.d`)
- `envivio_playlist.sh`

The following scripts will be installed in `/etc/rc.d/init.d`:

- `ess`
- `ess_admin`

The following configuration files will be installed in `/usr/Envivio/4Sight/Config`:

- `streamingserver.xml`
- `streamingadminserver.conf`
- `relayconfig-sample.xml`

The movie directory is set by default to `/usr/Envivio/4Sight/Movies`.
Playlists will be created by default in the movie directory.
Log files will be created in `/usr/Envivio/4Sight/Logs`.

Uninstallation under Linux

To stop 4Sight, use the command:
`/etc/rc.d/init.d/ess stop`

Then stop the WebAdmin with the following command:
`/etc/rc.d/init.d/ess_admin stop`

To uninstall 4Sight, run the following command:
`rpm -ev 4SightSE`

Installing the Envivio Linux Kernel

Envivio has developed a specific Linux kernel patch to fix some Linux kernel bugs.

You must install this patch if you want to use the SDP broadcaster in relay (see “Using 4Sight as a relay” on page 42) or switcher (see “Stream Switcher” on page 75) configurations.

WARNING

Envivio Linux kernel patch is only valid for Linux version 2.4.18.

Installing the Linux kernel patch

To install the Linux kernel patch, follow these steps:

1. If 4Sight is started, stop it.
2. Enter the following command:
rpm -ivh EnvivioKernel-2.4.18-14Envivio.i386.rpm
3. Go to `/usr/Envivio/4Sight/Config` and open the file `streamingserver.xml`
4. Set the `forward_announce_messages` parameter to `true`
5. Then reboot

Uninstalling the Linux kernel patch

To uninstall the Linux kernel patch, follow these steps:

1. If 4Sight is started, stop it.
2. Enter the following command:
rpm -e EnvivioKernel

CHAPTER 4

Quick start

Overview

This section describes step by step instructions for using 4Sight.

Starting 4Sight

1. Put the MPEG-4 movie file(s) you want to stream in your movie folder.
2. Launch the application:

Under Windows,

- if you have installed 4Sight as a service, go to “Chapter 5 on page 51”,
- otherwise, go to the Start/Programs/Envivio/4Sight 3.1 menu, and launch 4Sight.

Under Linux, type `/etc/rc.d/init.d/ess start`

Streaming your movie

1. To view your movie, open your MPEG-4 player (for example, EnvivioTV™) on any PC connected to the same network as the server.
2. Open the following URL:
`e-rtsp://<server_ip_address>:7070/mp4profile.mp4`

Obtaining the status of 4Sight

Type `/etc/rc.d/init.d/ess status`

Stopping 4Sight

Under Windows,

- if you have installed 4Sight as a service, go to “Chapter 5 on page 51”,
- otherwise, close the 4sight windows.

Under Linux, type `/etc/rc.d/init.d/ess stop`

CHAPTER 5

Streaming Server

Version 3.1

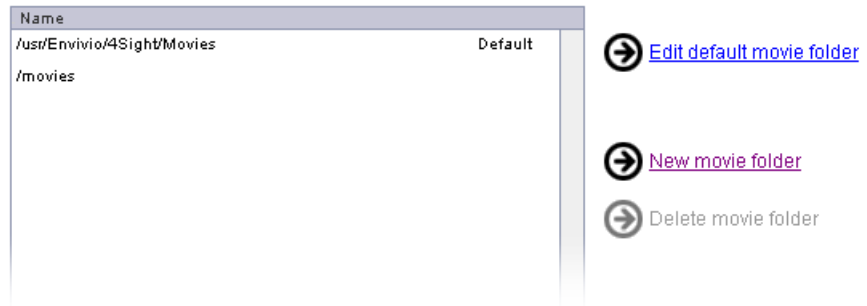
Setting up Movie Folders

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 71 for further details).
2. Click on the Movie Folders menu item. You will see a list of movie folders.

Movie Folders Settings



3. To change the default movie folder, select one, then 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.

5. Click the Saves Changes button to add this new folder.

Movie folder details

Movie folder name:

Save Changes

Using 4Sight as a reflector

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 an RTSP server that has the ability to join a multicast, and to convert the multicast stream into a series of individual unicast streams. A reflector feeds each client with an individual stream.

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

If you want the EnvivioTV™ client to connect to the reflected broadcast, use the following URL:

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

where live.m4e is the name of the SDP created by the Playlist Broadcaster (or any other broadcaster).

WARNING

If you use reliable UDP (see “reliable_udp” on page 113), check that the M4E file does not contain the following line:

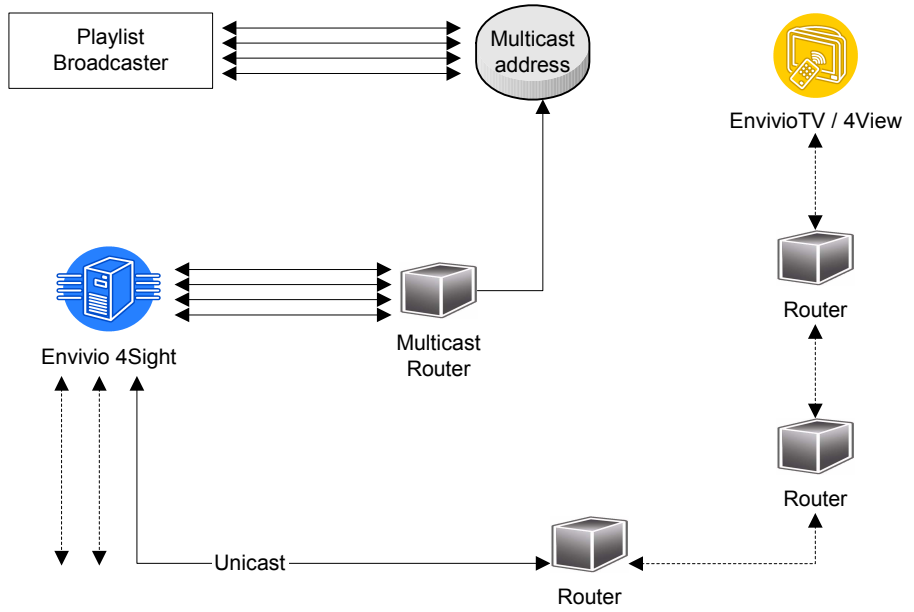
b=RR:0

This may prevent the player from sending the RTCP packets that are mandatory for using reliable UDP.

NOTE

There is no fixed limit to the number of live broadcasts that a server can reflect, nor to the number of clients that can connect to one reflected broadcast. The only limitations are the CPU, the memory and the bandwidth.

Figure 1. Reflecting



Using 4Sight as a relay

4Sight acts as a relay server when it accepts a stream from a source (a broadcaster or another streaming server) and sends it or "relays" it to one or more other streaming servers.

Basic setup

Use the WebAdmin to set up a relay (see "WebAdmin" on page 71).

Preliminary setup of the server

Before configuring 4Sight as a relay, you need to check your streaming server configuration file.

In the `streamingserver.xml` file, look for the `ESSRelayModule`. You should see something like:

```
<MODULE NAME="ESSRelayModule" >  
  <PREF NAME="relay_prefs_file" >  
    /usr/Envivio/4Sight/Config/relayconfig.xml  
  </PREF>  
</MODULE>
```

You can modify the `relay_prefs_file` parameter and specify the target file you want to use. This file is the only file needed by the streaming server to configure relays. If this file does not exist, it will be created.

Setting up a relay using the WebAdmin

1. Using a web browser, connect to the WebAdmin (see “WebAdmin” on page 71).
2. Click on 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:

Relay Details

Relay Name:

Status: Enable at server startup

Source Settings



Fill in the different fields and then click the Save Changes button.

NOTE

Common use cases are detailed later (see page 44).

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

NOTE

You can click the  or  buttons to respectively enable or disable the relay.

Relay Settings



Common configurations

Relay a UDP stream to one or more destinations

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

1. Specify the sender 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 must be separated by semicolons or spaces).

4. Provide the information about the destination (you can add more destinations if needed). Then click the Save Changes button.

Source Settings

Description (optional):

Source Hostname or IP Address:

Relay UDP Source:

UDP Port List:

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

Destination Settings (1 of 1)

Description (optional):


Hostname or IP Address:

UDP Ports:

Multicast TTL:

Local Output Interface (optional):

SAP Session Name (optional):

 [Remove Destination](#)

- If a local interface is specified then the output stream will be directed to this interface.
- If a SAP session name is specified, then the output will also broadcast an SAP announce message with the given name. If no SAP session is broadcasted from the source or if no session name is given in the output, there will not be any SAP output for the destination.

NOTE

The multicast time to live (TTL, see “Glossary” on page 131) 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 announced stream

If you are using a broadcaster that supports RTSP announces (such as the Playlist Broadcaster) in conjunction with 4Sight, you can relay the announced stream. For this, you need to set up your relay with the IP address of your broadcaster and select the Wait for announced streams radio button.

Source Settings

Description (optional):

Source Hostname or IP Address:

Relay UDP Source:

 UDP Port List:

Request incoming stream

 MPEG4 or SDP File:

 User Name:

 Password:

Wait for announced stream(s)

 SDP File (optional):

Wait for an SAP announced session

 SAP session to listen:

The announce can be either announced directly to the server or the server can look for a specific announced session using SAP. In the first case, the SDP filename will act as filter and server will only get announces with the specified filename. In the second case however, the session name is mandatory.

Relay a 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.

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.

Source Settings

Description (optional):

Source Hostname or IP Address:

Relay UDP Source:

UDP Port List:

Request incoming stream

MPEG4 or SDP File:

User Name:

Password:

Wait for announced stream(s)

SDP File (optional):

Wait for an SAP announced session

SAP session to listen:

Relay a UDP stream using a specific network interface card

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.

For example, imagine that you have two network interface cards, the first one with the IP address 192.178.106.22 and the second one with the IP address 193.178.105.126 and you want to use the second one for sending packets. Enter the following lines:

Destination Settings (1 of 1)

Description (optional):	<input type="text" value="Destination of MyRelay"/>
Hostname or IP Address:	<input type="text" value="237.52.51.50"/>
UDP Ports:	<input type="text" value="6606, 6608"/>
Multicast TTL:	<input type="text" value="15"/>
Local Output Interface (optional):	<input type="text" value="193.178.105.126"/>
SAP Session Name (optional):	<input type="text"/>

 [Remove Destination](#)

NOTES

If you do not specify the Local Output Interface parameter, the default network interface card will be used. If you do not know which card is the default one, ask your system administrator.

If the source is using either SDP announce or SAP announce and a session name has been given for the destination, then the SAP message will be forwarded with the new session name.

Status of the configured relays

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

The following window should be displayed:

Relay Status

Display entries

Page Refresh Interval:

List of active relays

Fri, 10. Oct 2003 16:21:43

Relay Name ▲	Source	Destination	Bitrate/Output	Total bytes relayed
MyRelay	My Source	My Destination	695.281Kbps	1.139 MB

1 active relay

Hints

Passwords

Passwords are not encrypted in the configuration file. You should ensure that your relay configuration files are adequately protected on your system.

Relay verification

To see if streams are getting relayed properly, you can use a traffic analyzer. For instance, you can find an analyzer on <http://analyzer.polito.it>.⁽¹⁾

1. This analyzer requires WinPcap, available at <http://winpcap.polito.it>.

Running 4Sight as a service

NOTE


This feature is available only on Windows.

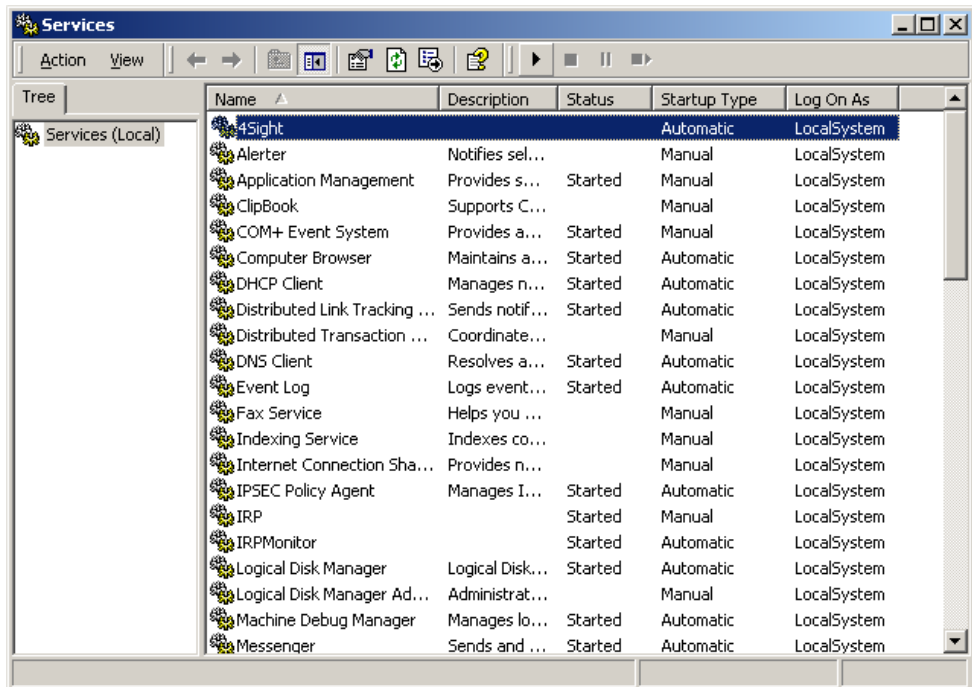
Installing 4Sight as a service


During the installation process, you are prompted to install 4Sight as a service. If you did not select this option, you can install it by following these steps:

1. Open an MS-DOS window.
2. Go into the Bin directory where you installed 4Sight and type:
4Sight -i

Starting, stopping and removing the 4Sight service

To start the 4Sight service, go to the Start/Settings/Control panel/Administrative Tools/Services menu and click the  button to start the 4Sight service.



To stop the 4Sight service, go to the Start/Settings/Control panel/Administrative Tools/Services menu and click the  button to stop the 4Sight service.

To remove the 4Sight service, open an MS-DOS window, go into the directory where you installed 4Sight and type:

4Sight-r

Embedding streamable content in a web page

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

To embed streamable content in a web page, two solutions are available:

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

```
<a href="yourfile.m4e">My content</a>
```

where yourfile.m4e contains:

```
url=rtsp://nameoftheserver/yourfile.mp4
```

2. Or use an anchor with an RTSP URL:

```
<a href="e-rtsp://nameoftheserver/yourfile.mp4">My content</a>
```

CHAPTER 6

Playlist Broadcaster

Version 3.1

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 71 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: Repetition: Items only repeat after other items have played

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

Available Movie Folders:

Movie Folder Content	Order	Playlist Content	Weight
<input type="checkbox"/> diehard			
<input type="checkbox"/> mib			
<input type="checkbox"/> moulin			
<input type="checkbox"/> mp4playlist			
<input type="checkbox"/> riviere			
<input type="checkbox"/> shakira			
<input type="checkbox"/> mp4profile.mp4			
<input type="checkbox"/> RivieresPourpres-720x576-25fp...			
<input type="checkbox"/> RivieresPourpres-720x576-25fp...			
<input type="checkbox"/> RivieresPourpres-720x576-25fp...			
<input type="checkbox"/> SHAKIRA_3000_p.mp4			
<input type="checkbox"/> SHAKIRA_3001_p.mp4			
<input type="checkbox"/> SHAKIRA_3002_p.mp4			

Open Folder

Refresh Folder

Remove Item

Send this playlist to a remote host or address

Hostname or IP Address:

Destination Base Port:

Multicast TTL:

Output interface:

Log this playlist's activity

Select an announce mode for this playlist

- No announce for this playlist
- Broadcast SDP on the same address port 9875
Session name:
- Broadcast SDP using SAP
Session name:
- RTSP Announce (For unicast addresses only, requires authentication and write access)
- IP Address:
- RTSP Port:
- Destination SDP filename:
- User Name:
- Password:

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.

Playlists

Available Playlists:

Name	Status	
mp4playlist	Stopped	▶
myPlaylist	Stopped	▶

- ➔ [New MPEG4 Movie Playlist](#)
- ➔ [Edit Playlist](#)
- ➔ [Delete Playlist](#)

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.

Playlists

Available Playlists:

Name	Status	
mp4playlist	Stopped	▶
myPlaylist	Stopped	! ▶

- ➔ [New MPEG4 Movie Playlist](#)
- ➔ [Edit Playlist](#)
- ➔ [Delete Playlist](#)

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

Setting up a playlist using the command line interface

You can also use the Playlist Broadcaster as a command line program. To do this, you need to create two files (that are automatically created by the WebAdmin).

- The first one is called the config file and is named using the playlist name to which you append a CONFIG extension.
- The second one is the list of files; its name is the name of the playlist to which you append the PLAYLIST extension.

You should then have two files:

- myPlaylist.config
- myPlaylist.playlist

The config file will contain the following set of parameters:

```
playlist_file /usr/Envivio/4Sight/Movies/ myPlaylist/myPlaylist.playlist
  play_mode sequential_looped
  destination_ip_address 127.0.0.1
  #broadcast_name "myPlaylist"
  sdp_file /usr/Envivio/4Sight/Movies/myPlaylist.m4e
  destination_sdp_file /usr/Envivio/4Sight/Movies/myPlaylist.m4e
  broadcast_SDP_is_dynamic enabled
  pid_file /usr/Envivio/4Sight/Movies/ myPlaylist/myPlaylist.pid
```

The playlist file will contain the following lines:

```
*PLAY-LIST*
  /usr/Envivio/4Sight/Movies/ mp4profile.mp4
```

The optional number associated to each movie is the weight of the movie. This number is used when you are doing a weighted random playlist. The higher the number, the more often the movie will be played.

You can then start the playlist using the following commands:

- `EnvivioPlaylistBroadcaster -p myPlaylist.config`
- `EnvivioPlaylistBroadcaster myPlaylist.config`

NOTE

`EnvivioPlaylistBroadcaster` is available under `/usr/Envivio/4Sight/sbin`.

The first command reads and parses the config file and the playlist file, in order to detect incompatibilities or errors. If the first command is successful, use the second one to start the broadcast of the playlist.

Common scenarios: Broadcasting a playlist

Single server configuration

If you have 4Sight and Playlist Broadcaster on the same machine, one simple way to broadcast a playlist is to configure your playlist to drop the SDP file directly into the movie folder of your streaming server, and to use the server as a reflector (see “Using 4Sight as a reflector” on page 40).

This is done by giving the full path to the SDP file in the movie folder.

1. For example, if your movie folder is `/usr/Envivio/4Sight/Movies/`, and your SDP file is called `myPlaylist.m4e`, the path you will have to use is:
`/usr/Envivio/4Sight/Movies/myPlaylist.m4e`
2. Either set the SDP File text box to this path, or set the two parameters `sdp_file` and `sdp_destination_file` to this path.
3. When you start the playlist, the SDP file will be created in the movie folder and will be accessible using your player by connecting to a URL like:
`e-rtsp://ip_address_of_the_server/myPlaylist.m4e`

Multiple servers: Unicast configuration

You have two machines, one with Playlist Broadcaster, the other with 4Sight, and you want to make a playlist accessible through 4Sight.

When you create or edit the playlist, you need to add some parameters to enable the Playlist Broadcaster to broadcast the playlist directly to the server.

1. If the address of your server is 192.163.25.14, then this address needs to be set as the destination address. To do so, in the WebAdmin, enter the address in the Hostname or IP Address field.

Send this playlist to a remote host or address

Hostname or IP Address:	<input type="text" value="192.163.25.14"/>
Destination Base Port:	<input type="text" value="5004"/>
Multicast TTL:	<input type="text" value="1"/>
Output interface:	<input type="text"/>

2. Now you have two choices to set up the broadcast.
 - You can copy the SDP file (generated when the Playlist Broadcaster starts) to the movie folder of the server and you can then access the URL:
e-rtsp://ip_address_of_the_server/myPlaylist.m4e
 - Or you can set up a relay using the IP Address of the Playlist Broadcaster as a source (see “Using 4Sight as a relay” on page 42).

Multiple servers: Unicast and announce

This is a more automated alternative to the previous setup. Using an announce, the Playlist Broadcaster will send the SDP file directly to 4Sight. You will not have to copy the SDP file to the server. Additional setup is required.

1. First, you need to use the `qtpasswd` utility to enable write permission on the movie folder of the server (see page 89 for further details).
2. Then you need to modify your playlist configuration to provide a user name and a password: in the WebAdmin, fill in the User Name and Password fields.

Send this playlist to a remote host or address

Hostname or IP Address:

Destination Base Port:

Multicast TTL:

Output interface:

Log this playlist's activity

Select an announce mode for this playlist

- No announce for this playlist
- Broadcast SDP on the same address port 9875
- Session name:
- Broadcast SDP using SAP
- Session name:
- RTSP Announce (For unicast addresses only, requires authentication and write access)
- IP Address:
- RTSP Port:
- Destination SDP filename:
- User Name:
- Password:
-

3. You can test your broadcast in the same way as for basic unicast mode.

Broadcast to a multicast address

Broadcasting to a multicast address is similar to broadcast to a unicast address. All you need to do is to replace your unicast address with the multicast address you want to broadcast to.

Send this playlist to a remote host or address

Hostname or IP Address:

Destination Base Port:

Multicast TTL:

You can then:

- relay the multicast stream using a streaming server by setting the multicast address as a source,
- if you do not use SDP broadcast or SAP, you should distribute the SDP file to allow clients to connect directly to the multicast stream by opening it.

NOTE In this case, only the Playlist Broadcaster needs to be running.

- use the server as a reflector (see “Using 4Sight as a reflector” on page 40 for further details).

Configuration file parameters

This section describes the different parameters that you can modify in your playlist configuration file.

Table 1. List of parameters

Keyword	Value
playlist_file	Playlist filename and path. The path can be absolute or relative.
play_mode	Three modes are available <ul style="list-style-type: none">• 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.• weighted_random: The media is broadcast in random order using the specified weights to determine how often an item plays. The media continues to play in random order until you stop the broadcast.
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

Table 1. List of parameters

Keyword	Value
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.
recent_movies_list_size	The minimum number of files that must play before a file may be repeated. You can specify this only if the play mode is <code>weighted_random</code> . By default this value is zero, so that all the media files play randomly based only on the weight you have assigned to each one. Note: Be sure to set this value to a number less than the number of media files in the playlist. If you set it to a number equal to or greater than the number of media files, the broadcast will play each media file once then stop.
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 <code>sdp_file</code> .
announce_type	Specify the announce type for this playlist. Value can be "SAP", "SDP_Broadcast", "RTSP" or "None". If not specified, value will be "None"
session_name	Name of the SAP session if <code>announce_type</code> is SAP.
broadcast_SDP_is_dynamic	Can be either "enabled" or "disabled".
broadcaster_name	User name if one needs to be provided to the destination address.

Table 1. List of parameters

Keyword	Value
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.
rtsp_port	RTSP port to use in case of an RTSP announced 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.
rtsp_destination	In case of an RTSP announce, announce to this destination address (can be different from the movie broadcast address)
sap_session_name	In case of an SAP announce, use this session name
sdp_session_name	In case of an SDP broadcast, use this session name
multicast_interface	Use this local interface to broadcast the playlist

Command line parameters

The Playlist Broadcaster takes a playlist configuration file as a mandatory parameter.

Table 2. Optional flags

[-v]	Displays the version of the Playlist Broadcaster
[-h]	Displays the help screen
[-p]	Reads and verifies the playlist configuration file and movie list
[-c]	Shows the current movie played by the Playlist Broadcaster
[-a]	Sends an RTSP announce to the destination server
[-t]	Sends the broadcast over TCP to the server
[-i]	Specifies the destination IP address (Note: this setting overrides the configuration file value)
[-e file]	Logs errors to the specified filename
[-f]	Forces the Playlist Broadcaster to create a new SDP file for each execution

Troubleshooting

The Playlist Broadcaster and the WebAdmin

If you are using the WebAdmin to configure, start, and stop playlists, make sure the following parameters are set correctly in the `streamingadminserver.cfg` file

- `Plroot`: set it to your playlist root.
By default, it is set to `path_to_movie_folder/playlist`
- `QtssPlaylistBroadcaster`: the path to the `EnvivioPlaylistBroadcaster` executable
- `Qtssname`: the path to the 4Sight executable

Playlist configuration file

The parameter giving the name of the broadcast is `#broadcast_name`.

NOTE

The `#` sign at the beginning is mandatory and is not interpreted as a comment.

Make sure to set the `sdp_file` and the `destination_sdp_file` parameters to the same value in the configuration file.

Passwords are not encrypted in the configuration file. You should ensure that your playlist configuration files are protected on your system.

Playlist files

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).

Command line mode

When you are using the Playlist Broadcaster in command line mode, if you stop a broadcast and change the broadcast configuration file, be sure to delete the SDP file before restarting the broadcast. Otherwise the Playlist Broadcaster will not create a new SDP file, and your changes will not take effect.

CHAPTER 7

WebAdmin

Version 3.1

Overview

The WebAdmin allows you to remotely monitor and configure 4Sight.

Starting the WebAdmin

You can launch the WebAdmin from the Envivio menu on Windows, or from the `envivio_admin.sh` script on Linux.

NOTE

This feature requires Perl. You can find it at <http://www.perl.com>.

To monitor 4Sight, you can either:

- launch 4Sight Administration from the Envivio menu on Windows
- or open a browser and type:
 - `http://127.0.0.1:1220`, on the local machine, or
 - `http://ip_address_of_the_server:1220`, on a remote machine.

The minimum browser versions are Netscape 6.2 and Internet Explorer 6.0.

NOTE

You can define a password to restrict access to this service (see “Authentication and Authorization” on page 89).

Stopping the WebAdmin

Type `/etc/rc.d/init.d/ess_admin stop`

Obtaining the status of the WebAdmin

Type `/etc/rc.d/init.d/ess_admin status`

Configuration parameters

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.
proot	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).

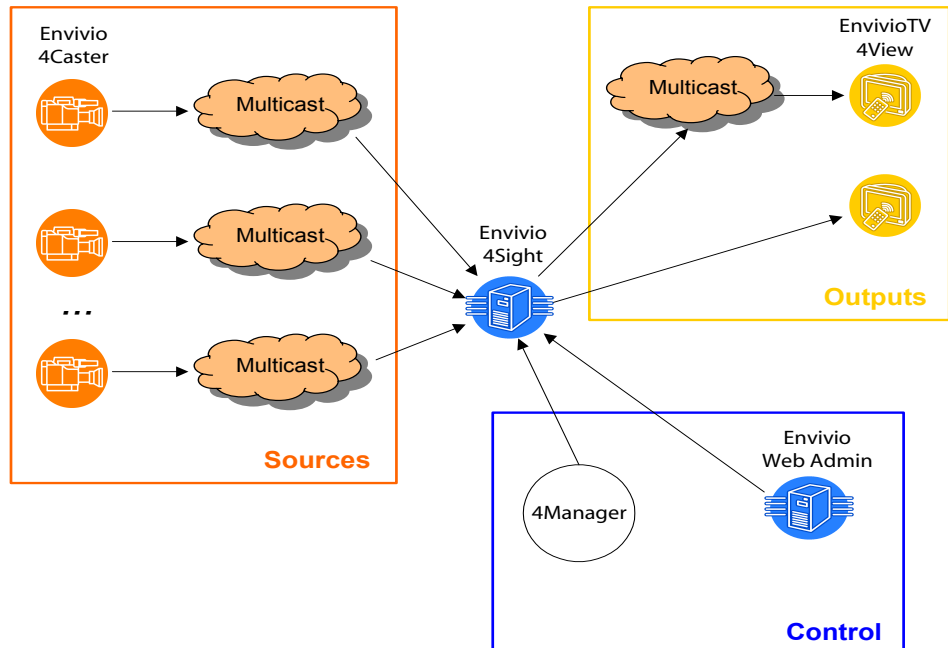
CHAPTER 8

Stream Switcher

Version 3.1

Overview

The Stream Switcher is a tool for relaying one stream out of a number of sources and then permits switching to a different source. The switch to a different source is seamless for the player.



The switch between sources can be controlled by an external application such as Envivio 4Manager or by the WebAdmin.

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 71 for more details).
2. Click on 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:

Switcher Details




Switcher Name:

Status: Enable at server startup



4. Fill in the different fields you need to set up your Stream Switcher and then click on the Save Changes button.
5. Your relay will then be listed in the relay list, followed by its status that can be either Enabled or Disabled.

Switcher Settings

Switcher List:

Name	Status	
MySwitcher	Enabled <input type="checkbox"/>	 New Switcher
		 Edit Switcher
		 Delete Switcher

NOTE

You can click the  or  buttons to respectively enable or disable the relay.

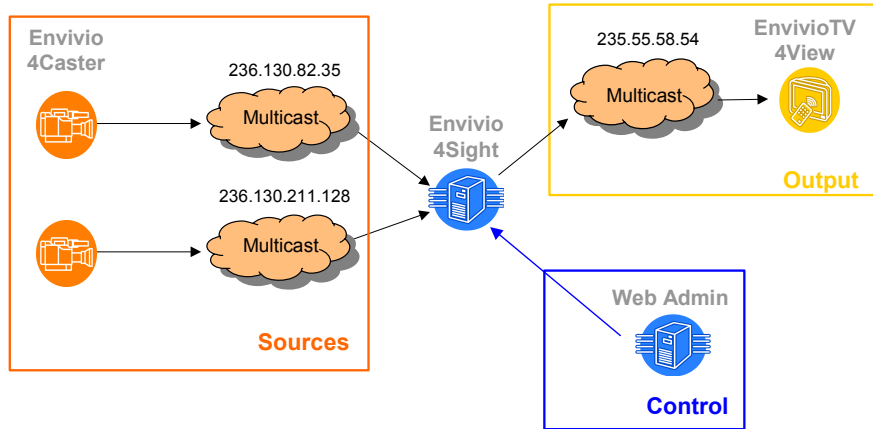
Using a specific network interface card as output

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.

See “Using 4Sight as a relay” on page 42 for further details.

Configuration example

We will use the following network structure to demonstrate the configuration of the Stream Switcher.



To switch a stream, you need to follow these steps:

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
Mpeg4 payload type:	<input checked="" type="radio"/> ASP <input type="radio"/> H264
Local Output Interface (optional):	193.178.105.126

NOTE

For stream switching, the server needs to know the payload type of the streams that used. Specify the stream type using the radio boxes. Payload type different from ASP and H264 are not supported for stream switching.

2. Provide the information about the input. Enter the hostname or IP address, then specify the ports you want to listen to. Ports must be separated by commas or spaces.

Source Settings (1 of 1)Description (optional): Hostname or IP Address: UDP Port List:  [Remove Source](#)

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

Source Settings (1 of 2)Description (optional): Hostname or IP Address: UDP Port List:  [Remove Source](#)

Source Settings (2 of 2)Description (optional): Hostname or IP Address: UDP Port List:  [Remove Source](#)

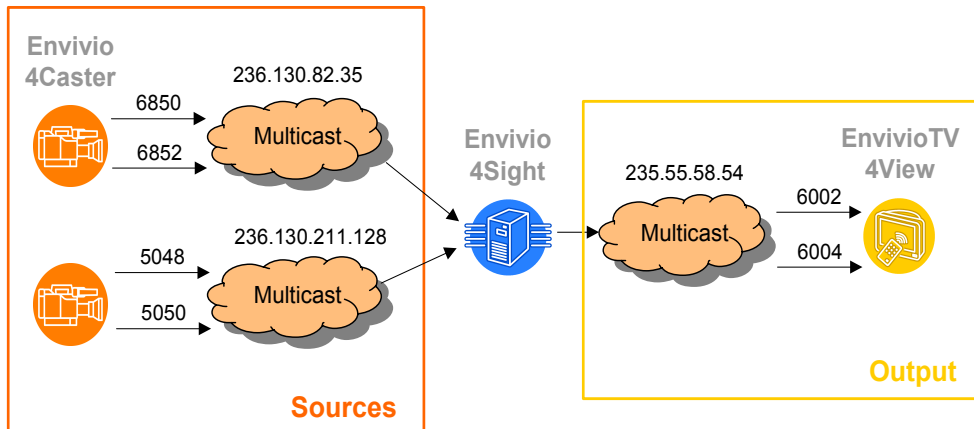
4. Then click the Save Changes button.

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.

Accessing the stream generated by the switcher

We will use the following network structure to demonstrate how to access the stream generated by the Stream Switcher.



The first Envivio 4Caster broadcasts two streams to the IP multicast address 236.130.82.35:

- the first stream (video) is broadcast to port 6850
- the second stream (audio) is broadcast to port 6852

The second Envivio 4Caster broadcasts two streams to the IP multicast address 236.130.211.128:

- the first stream (video) is broadcast to port 5048
- the second stream (audio) is broadcast to port 5050

Viewing the active source

To view the output of the switcher:

- If the broadcasters broadcast the SDP to port 9875, the switcher will automatically transfer the SDP to port 9875 of the output address. See “Installing the Envivio Linux Kernel” on page 32 for further details on using this feature.
- If the broadcasters broadcast the SDP using SAP, the switcher will not be able to send the SDP to the output address. In this case you need to modify the SDP file (see next paragraph), and open it from your player.
- Otherwise, modify the SDP file created by one of the sources (see next paragraph), and open it from your player.

Modifying an SDP file

To create an SDP file that the player can use to view the active source:

1. Retrieve the SDP file created by one of the sources, for example, the first 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**, and

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

Your SDP file should look like this:

```
v=0
o=- 3557664794 484063668 IN IP4 192.168.105.42
s=ELB
c=IN IP4 235.55.58.54
t=0 0
a=mpeg4-iod:"data:application/mpeg4-iod;base64,AoIvAE///w/3/
wOBOgABQJxkYXRhOmFwcGxpY2F0aW9uL21wZWc0LW9kLWF1O2Jh
c2U2NCxBVmdCS2dVZkF5WUF5U0FBWIFRTkICRUJ0M1FBRzNkQUFBM
jdvQVIRQUVRQUFWK1FBQUFENkNBQUFBQUFBd0VxQXA4REpnQmxJ
QUFCQkExQUZRQUdBQUFDN2dBQUFY0FCaEFBUkFBQXJFUUFBQVBv
SUFBUFBQUQEDQEFAADIAAAAAAAAAAAGCQEAAAAAAAAAAAAANpA
AJARmRhdGE6YXBwbGljYXRpb24vbXBIZzQtYmlmcy1hdTtiYXNINjQsd
0JBU2dUQXFVCVzBtRUVIOEFBQUlVQUFBQkVLQ0tDbjQEEgINAAAgAAA
AAAAAAAAFAwAAYAYJAQAAAAAAAAAAAA"
a=ISMA-compliance:1,1,1
m=video 6002 RTP/AVP 96
a=rtpmap:96 MP4V-ES/90000
a=fmtp:96 profile-level-id=247;
config=000001B0F7000001B5A91D4141415F000001000000012000CB
88B0D450B042414183
a=mpeg4-esid:201
a=x-envivio-verid:0002102E
m=audio 6004 RTP/AVP 97
a=rtpmap:97 mpeg4-generic/44100/2
a=fmtp:97 profile-level-id=15; config=1210; streamtype=5;
mode=AAC-hbr; SizeLength=13; IndexLength=3; IndexDeltaLength=3
a=mpeg4-esid:101
a=x-envivio-verid:0002102E
```

Status of the configured switchers

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

The following window should be displayed:

Switcher Status

Display entries

Page Refresh Interval:

List of active switchers

Fri, 10. Oct 2003 16:44:19

Switcher Name ▲	Sources	Output	Bitrate	Bytes relayed
MySwitcher	My first source (Active) My second source	Switcher Output	842.672Kbps	2.706 MB

1 active switcher

You can switch to the other source by clicking the link "My second source".

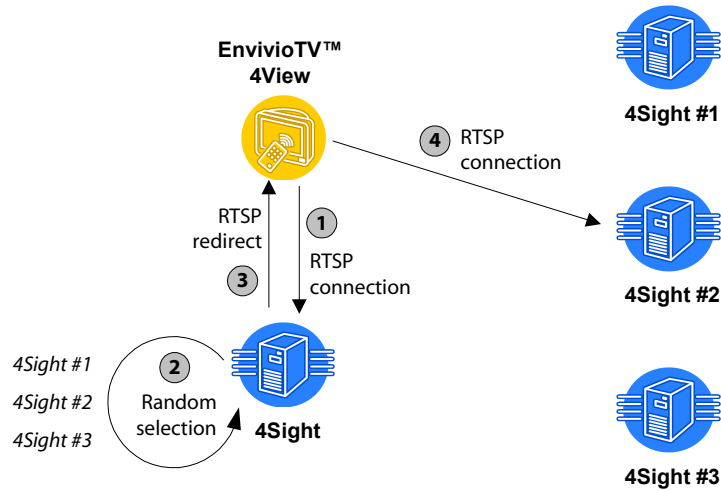
CHAPTER 9

Load Balancer

Version 3.1

Overview

The Load Balancer distributes traffic efficiently among streaming 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 on New Slave server to add a slave server to the list. The following page is displayed:

Slave server details

Slave server IP:

Save Changes

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

NOTE


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


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

Load Balancing Settings

Status: Enable Load Balancing

Server Address
192.167.156.23:7072
192.167.156.144:7070
192.178.152.17:7080

 [New slave server](#)

 [Delete slave server](#)

- 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. 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.

CHAPTER 10

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 streaming 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.

There should be one user file and one group file per streaming server. The paths to these files are defined in the `QTSSAccessModule` of your `streamingserver.xml` file. 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" >qtpasswd </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 the Config Directory: `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
```

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". <ul style="list-style-type: none"> • 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	<p>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.</p> <p>Two specific cases can be used:</p> <ul style="list-style-type: none">• require any-user any user is allowed• require valid-user any authenticated user is allowed
<Limit [READ WRITE]>...</Limit>	<p>Specifies access control for read and/or write privileges.</p> <p>If this parameter is not present or set, all parameters apply for read privileges.</p> <p>The parameters AuthName, AuthUserFile, AuthGroupFile and require can be defined between the two "Limit" tags.</p>

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 [-option] <username>
```

The following are optional flags that can be used:

[-F]	Does not confirm when deleting users or overwriting existing files
[-f]	Password file to modify (defaults to the value set in streamingserver.xml)
[-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

Common scenario

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 streaming 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
<code>AuthName</code>	ESS for US	ESS for Europe
<code>AuthUserFile</code>	<code>/usr/Envivio/4Sight/Config/qtusers_US</code>	<code>/usr/Envivio/4Sight/Config/qtusers_EU</code>
<code>require user</code>	<code>user_US1 user_US2 ... user_USN</code>	<code>user_EU1 user_EU2 ... user_EUN</code>

3. Create the password file with a specific realm by running:
`qtpasswd -f qtusers_US -c -r "ESS US" user_US1`
`qtpasswd -f qtusers_EU -c -r "ESS 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`

Restrict 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
```

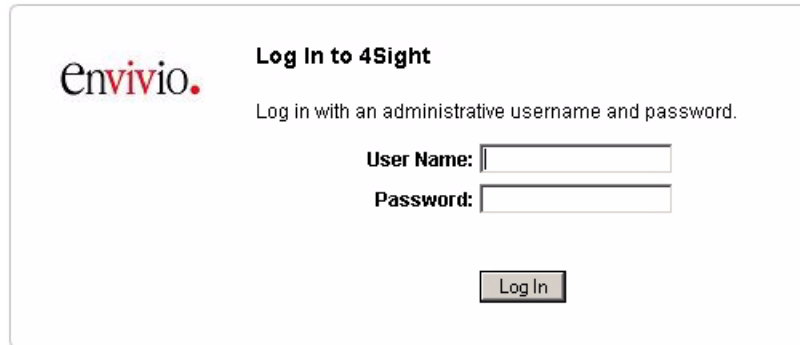
This allows access to the WebAdmin from local host and 192.168.105.187 and explicitly denies it from 192.168.105.113.

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 71).
2. Click on the Server Configuration>General Settings menu item.
3. Check 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 the WebAdmin.

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

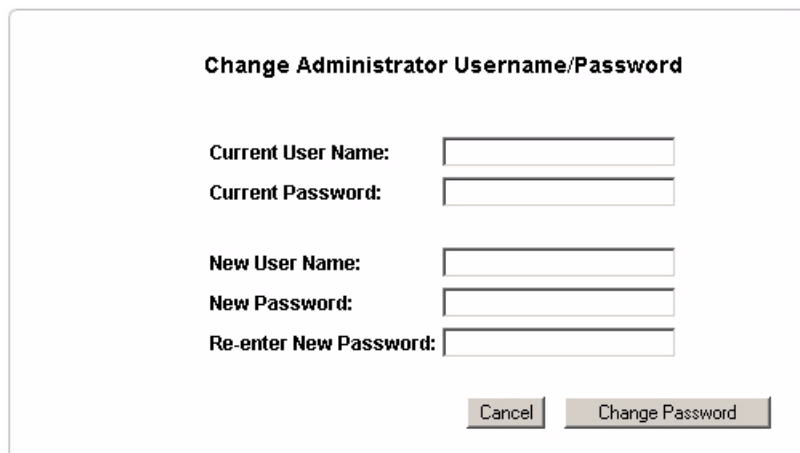


envivio. **Log In to 4Sight**
Log in with an administrative username and password.

User Name:
Password:

Changing the WebAdmin default user name and password

5. Using a web browser, connect to the WebAdmin (see “WebAdmin” on page 71).
6. Click on the Server Configuration>General Settings menu item.
7. Click the Change Admin Username/Password... link. The following page should be displayed:



Change Administrator Username/Password

Current User Name:
Current Password:

New User Name:
New Password:
Re-enter New Password:

8. Fill the different fields, then click Change password button. You will be disconnected from the WebAdmin.
9. Reconnect to the WebAdmin using the new user name and password.

Troubleshooting

Access file

- Parameter names and values are case sensitive. Respect the case for the following words:
AuthName, AuthScheme, AuthUserFile, AuthGroupFile, require, Limit, basic, digest
- Parameters in the access file take precedence over preferences in the streamingserver.xml file. Any value set for the realm or the authentication scheme in the access file will override the value set in the streamingserver.xml file (for the folder in which the access file is located).

qtpasswd

- User names cannot be more than 255 characters long and must not include a colon.
- Passwords cannot be more than 80 characters long.
- If the user name/password contains white space or characters that can be interpreted by the shell, enclose it in single quotes to avoid interpretation.

Remote access to the WebAdmin

If you are trying to access the WebAdmin from a remote machine, make sure that the LocalAccessOnly preference in the QTSSAdminModule section of the streamingserver.xml file is set to false and that enable_remote_admin is set to true.

CHAPTER II

External authorization module

Version 3.1

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 106.

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:
`/usr/Envivio/4Sight/Plugins`
3. Copy your authorization library into the Plugins directory.
4. Open the server configuration file, `streamingserver.xml`, located in:
`/usr/Envivio/4Sight/Config`
5. Update the following lines in the ESSEExternalAuthModule section:

```
<MODULE NAME="ESSEExternalAuthModule">  
<PREF NAME="enabled" TYPE="Bool16" >true</PREF>  
<PREF NAME="auth_lib_file">  
/usr/Envivio/4Sight/Plugins/your_lib  
</PREF>  
</MODULE>
```

NOTE

The value of the `enabled` preference may be `true` or `false`. It enables or disables the use of the external authorization module.

The `auth_lib_file` preference is the path of your authorization library. Replace `your_lib` by the file name of your library.

6. Restart the server

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;
}
```

APPENDIX A

Description of the configuration file parameters

General parameters

This chapter provides an overview of the different parameters for the Streaming Server. You can modify them to fit your needs.

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 streaming 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.
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).
error_logfile_verbosity	This controls how much information to log. A value of 0 means only fatal errors. 1 means fatal errors and warnings. 2 means fatal errors, warnings, and asserts. 3 means fatal errors, warnings, asserts, and debug messages.
error_logfile_size	Maximum allowed size in bytes for the error log before it rolls over. A value of 0 means unlimited.
error_logfile_interval	Amount of time in days between rolls of the error log. A value of 0 means unlimited.
error_logfile_dir	Error. Path to the error log file.
error_logfile_name	Name of the error log file.
force_logs_close_on_write	If true, log file will be closed after every write.
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.
movie_folder	Path to the root folder for movies.
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.
auto_start	If true the streaming server will be launched when the Web Admin starts.

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_thickening_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.
auto_delete_sdp_files	Turn on or off the SDP file deleter: files are deleted after the SDP t= endtime passes.
authentication_scheme	Set this to be the authentication scheme you want the server to use. "basic", "digest", and "none" are the currently supported values.
sdp_file_delete_interval_seconds	Check SDP files every interval seconds. The internal minimum value is 1 second. Changes to sdp_file_delete_interval_seconds and auto_delete_sdp_files take effect at the end of the current interval.
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.

LCD panel

Keyword	Value
lcd_serial_port	Specify the serial port where the LCD Panel is connected (if present)
lcd_device_name	Specify the type of the LCD Panel device. Valid values are "None", "MatrixOrbital" and "CrystalFontz"

Specific parameters

These parameters are module-specific.

Reflector parameters

Keyword	Value
enable_broadcast_push	Turn on or off a broadcaster RTSP push to the server.
max_broadcast_announce_duration_secs	Force announced SDPs to this maximum duration in seconds. 0 = infinite duration.
allow_duplicate_broadcasts	Turns on or off accepting setups on an existing broadcast stream.
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 <code>enforce_static_sdp_port_range</code> .
minimum_static_sdp_port	The minimum value for a static port. See <code>enforce_static_sdp_port_range</code> .
timeout_stream_SSRC_secs	If no packets for the current SSRC are received in time-out seconds, use the next SSRC received. This time-out is only in use when <code>use_one_SSRC_per_stream</code> is true.

use_one_SSRC_per_stream	Only reflect packets from a single SSRC. Setting this value to false allows all packets from any source to be passed to the clients.
kill_clients_when_broadcast_stops	When a broadcaster RTSP session goes down also teardown the clients watching the stream.

Access log file parameters

Keyword	Value
request_logfile_interval	Time in days between rolls of the access log. A value of 0 means unlimited.
request_logfile_size	Maximum allowed size in bytes for the access log. A value of 0 means unlimited.
request_logfile_dir	Path to directory where access logs are stored.
request_logfile_name	Name of the access log file.
request_logtime_in_gmt	Either "true" or "false". Logs date/time field in GMT if true and in local time if false.
request_logging	Either "true" or "false". This toggles access logging on and off.

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 <code>loss_thin_tolerance</code> , over the course of <code>num_losses_to_thin</code> 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 <code>loss_thick_tolerance</code> , over the course of <code>num_losses_to_thick</code> 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.
relay_stats_url	Enter this URL into a web browser and get a web page with current stats on your active relays.

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.

APPENDIX B

Configuration examples

Overview

The purpose of this appendix is to show different streaming solutions based primarily on Envivio 4Caster and 4Sight.

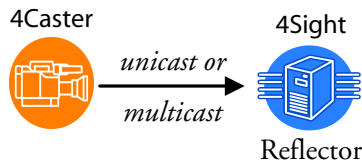
The configuration you choose will depend on whether you are streaming content to the Internet or on a LAN network (or both), the number of clients you want to reach, and your hardware configuration. The situation will be different if you want to stream MPEG-4 files to 100 clients or to 50,000 clients. It will also be different if you want to stream at the same time on a LAN and on the internet. Moreover, the considered solutions will depend on your software/hardware configuration. Is a commercial load-balancing solution necessary for your needs? How should you use 4Sight? How many machines with what sort of performance? These are the questions we would like to clarify in this appendix.

Configuration examples

Example #1

This is a basic configuration using 4Sight as a reflector to stream (or broadcast) a live event.

Figure 1. Simple reflecting

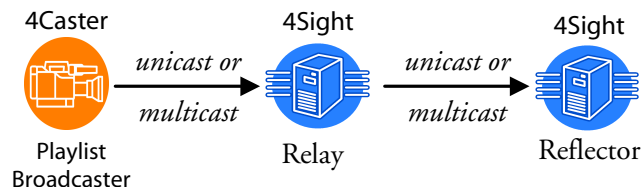


Example #2

In these configurations, a live stream (or simulated live stream) is relayed to one or more 4Sight servers, which distribute streams to the Internet.

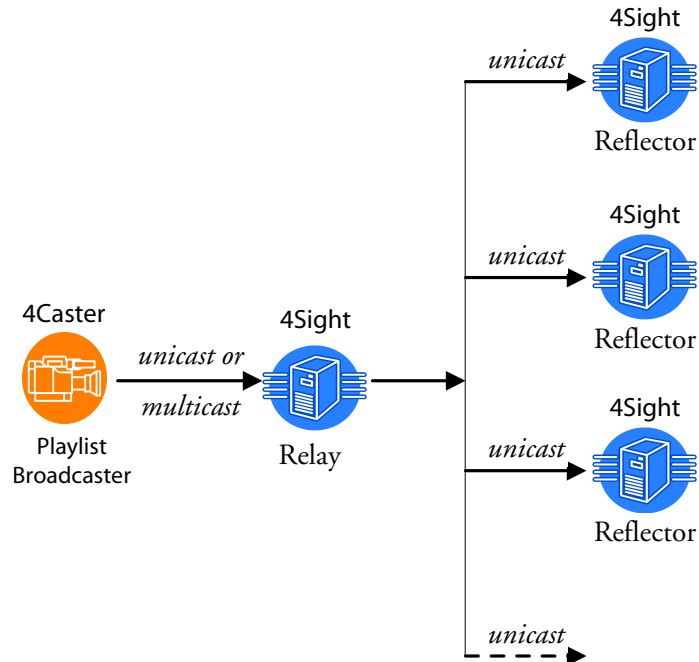
Using a relay, we can relay this live stream (or simulated live stream in the case of Playlist Broadcaster) to one or several 4Sight server(s), which will serve as stream distributors for the Internet.

Figure 2. Simple relay



A relay toward several 4sight servers will then be:

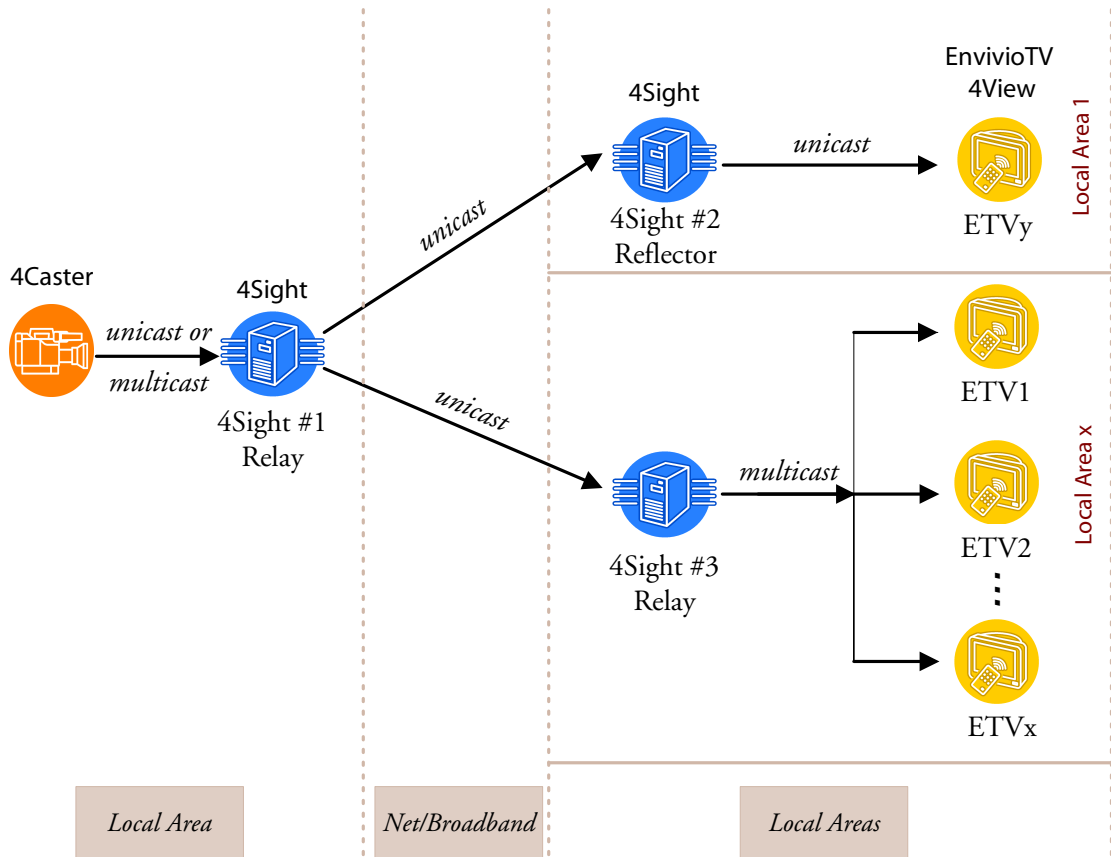
Figure 3. Multiple relay



Example #3

This example is a more complex configuration built on top of the former configuration examples.

Figure 4. Relay in a complex configuration



APPENDIX C

Streaming Server access log file

Version 3.1

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&avgbandwidth=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 coder-decoder or compression-decompression. 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

Numerics

4Caster 20, 23, 40, 81, 82, 122

4Coder 16

4Manager 76

A

access file 94, 101

access log 116
file 116

Access module

access file 116

group list 117

user list 117

add

destination 45, 80

group file 93

movie folder 38

user 96

admin module 117

admin request 117

administration port 29

announce 63

announced SDP

maximum duration 115

asserts 109

authentication 12, 90

realm 101

scheme 101, 112

authorization 12, 90

B

bandwidth 22, 40

client 114

server 110

statistics 110

stream 111

broadcast 23

playlist 55, 61

to multicast address 64

broadcaster

RTSP push 115

RTSP session 116

browser version 72

C

client 19, 20

available bandwidth 114

DNS 128

IP address 128

command line 70

interface 59

parameters 68

CONFIG extension 59

configuration file

parameters 49, 65

CPU 40, 130

D

debug messages 109

debugging

error messages 114

DNS 128

DNS round robin 108

E

embed streamable content 52

encrypted password 69, 93

EnvivioTV 16, 18, 34, 40, 70

error logging 109
 error messages
 debugging 114
 error resilience 18
 extension
 CONFIG 59
 LIC 28, 30
 M4E 133
 PLAYLIST 59

F

fast backward
 scale 118
 fast forward
 scale 118
 fatal errors 109
 file
 access 94, 101
 access log 116
 hinted 17, 70
 LIC 30
 licensing 27, 28, 30
 log 17, 31, 58, 67
 M4E 23, 40, 52, 133
 PID 67, 109
 registration 27, 28
 SDP 70
 user 90, 91, 93, 94, 96, 97
 firewall 108

G

GET request 108
 group file 90, 91, 93
 add 93
 modify 93
 remove 93
 GUID value 128

H

hint track 23
 hinted file 17, 70
 file
 hinted 23
 host program 128
 HTTP 19
 GET request 108
 return code 128
 tunneling 16

I

IETF 16
 install
 4Sight 27, 30
 4Sight on Linux 30
 4Sight on windows 27
 installation 27
 installer 27, 28
 Internet 19, 22, 122, 123
 Internet Explorer 72
 IP address 23, 46, 47, 48, 62, 65, 68, 73,
 108, 110
 IP network 16
 IP networks 11
 IPAccessList 117

L

LAN network 122
 launch
 WebAdmin 72
 LIC extension 28, 30
 LIC file 30
 file
 LIC 28
 licensing file 27, 28, 30
 Linux 30, 34, 35
 live event 123

- live stream 123
- live track 20
- log
 - asserts 109
 - client DNS 128
 - client IP address 128
 - debug messages 109
 - fatal errors 109
 - host program 128
 - HTTP return code 128
 - player ID 128
 - player language 128
 - player version 128
 - user agent 128
 - warnings 109
- log file 17, 31, 58, 67, 109
- login screen 98

M

- M4E file 23, 40, 52, 133
- memory 40
- modify group file 93
- monitoring
 - offline 17
 - online 17
- movie directory 29
- movie folder 34, 38, 61, 62, 63, 97
 - add 38
 - change 38
 - permission 63
- multicast 19, 20, 22, 44, 79
 - address 21, 22, 45, 64
 - forwarding 22
 - IP address 65
 - packet 67
 - playlist 67
 - session 21

- stream 40, 64
- streaming 16, 21
- time to live 45

N

- Netscape 72
- network traffic 21
- number of live broadcasts 40

O

- offline monitoring 17
- online monitoring 17

P

- packet loss 18
- parameters
 - command line 68
 - configuration file 65
 - password 47, 63, 69, 72, 91, 93, 97, 98, 101
 - encrypted 69, 93
- perl 72
- permission 63
- PID file 67, 109
- play/stop icon 58
- player
 - ID 128
 - language 128
 - version 128
- playlist 56
 - broadcast 55, 61
 - configuration file 69
 - set up 55, 56, 59, 77, 87
- Playlist Broadcaster 16, 20, 23, 40, 46, 53, 54, 55, 59, 61, 62, 63, 67, 68, 69, 70, 123
- PLAYLIST extension 59
- port

for client connection 108

Q

qtaccess 90, 91, 97
 qtgroups 90, 93
 qtpasswd 31, 63, 90, 91, 93, 96, 101
 qtusers 90, 93
 QuickTime 24

R

realm 73, 93, 94, 96, 97, 101, 108
 rebuffering management policy 18
 receiver 19, 20, 21
 red stop sign 58
 reflecting 12, 16, 40, 41, 115, 123
 reflector 22, 23, 40, 61, 64, 123
 registration file 27, 28
 registration form 27
 relay 42, 43, 44, 45, 46, 77, 123
 announced stream 46
 configuration 47
 configuration file 49
 list 44, 77
 server 23, 42
 set up 43
 status 49
 UDP stream 44, 48
 relaying 12, 16, 76
 reliable UDP 19, 40, 113
 slow start 114
 remove
 group file 93
 service 51
 root directory 73
 router 22
 RTCP
 incoming packets 113
 packets 40

status packet 110

RTP 19

 client 110

 packets 19, 20, 23, 24, 40

RTSP 19, 20, 67, 68

 announce 46

 client 111

 debugging parameters 114

 error 453 110

 error response 114

 request 111

 server 40

 Speed header 118

 tunneling through HTTP 108

 URL 52

S

scenarios 61

SDP 20

 file 21, 23, 57, 58, 61, 62, 63, 64, 66,
 68, 70

 file deleter 112

 information 23

 returned to the client 118

 standard 23

send buffer 108, 109

sender 20, 21

 address 44

server 16, 19, 20, 22, 23, 34, 38, 40, 44,
 47

 bandwidth 110

 source address 112

set up

 playlist 55, 56, 59, 77, 87

simultaneous users 17

source address

 server 112

- specific network interface card 48, 78
- SSRC 116
- static port 115
- statistics
 - bandwidth 110
- stop
 - service 51
- stream 19
 - bandwidth 111
 - switcher 12, 75, 76, 79, 81, 120
- streaming protocol 19
- streaming session 20
- system requirements 26

T

- TCP 19, 68
 - send buffer 108
- TCP/IP protocol 132
- thinning algorithm 111
- traffic analyzer 49
- transport header 112
- TTL 45, 67

U

- UDP 19
 - reliable 113
 - stream 44
- UDP/IP 23
- unicast 44, 62, 79
 - address 45
 - streaming 16, 20
- URL 34
- user agent 128
- user file 90, 91, 93, 94, 96, 97
 - create 96
- user name 47, 63, 91, 93, 95, 98, 101

V

- video on demand 16

W

- warnings 109
- web page
 - embed streamable content 52
- WebAdmin 16, 38, 43, 55, 56, 59, 62, 63, 69, 76, 77, 87, 98, 99, 100, 101
 - access restriction 98
 - parameters 117
 - remote access 101
 - restrict access 98
- web-based interface 12, 17
- Windows 12, 27, 34, 35
 - 2000 advanced server 26
 - 2003 advanced server 26

