



REALSERVER ADMINISTRATION GUIDE

RealServer 8.0

RealNetworks, Inc.
2601 Elliott Avenue, Suite 1000
Seattle, WA 98121
U.S.A.

<http://www.real.com>
<http://www.realnworks.com>

© 2000 RealNetworks, Inc. All rights reserved.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of RealNetworks, Inc.

Printed in the United States of America.

RealNetworks, the Real Bubble, RealSystem, RealSystem G2, RealServer, RealPlayer, RealJukebox, RealProducer, RealProxy, RealAudio, RealVideo, RealText, RealPix, RealPresenter, RealPublisher, SureStream, Real Accessory Zone, RBN, Real Broadcast Network, Real.com, Real.com Guide, Real.com Take 5, Real G2 with Flash, RealMP, RealSites, RealSlideshow, and RealStore.com are trademarks or registered trademarks of RealNetworks, Inc.

Other product and corporate names may be trademarks or registered trademarks of their respective companies.



CONTENTS

	INTRODUCTION	1
	Overview	1
	How This Manual Is Organized	1
	Conventions Used in This Manual	4
	Available Features	6
	Additional RealSystem Resources	6
	Technical Support	7
1	QUICK START	9
	Overview	9
	Starting RealServer	10
	Using RealSystem Administrator to Test Your RealServer	10
	Playing Sample Files	11
	Creating and Streaming Your Own On-Demand Clips	12
	Step 1: Creating a Music Clip	12
	Step 2: Putting the Music Clip in the Content Directory	14
	Step 3: Creating a Link (Optional)	14
	Step 4: Playing the Sample Clip	14
	Creating and Broadcasting Live Events	15
	Step 1: Encoding an Event	15
	Step 2: Creating a Link (Optional)	17
	Step 3: Playing the Clip	17
2	WHAT'S NEW IN REALSERVER 8?	19
	New Features in RealServer Version 8	19
	New Features in RealServer Version 7	20
	Compatibility with Earlier Releases	21
3	OVERVIEW OF REALSERVER	23
	What Is RealServer?	23
	What Is RealSystem?	24
	How RealServer Works	25
	Channels and Protocols	26
	Communication Between Encoders and RealServer	26
	Communication Between RealServer and RealPlayers	27

Streaming Media Delivery Methods	28
Choosing a Delivery Method	29
On-Demand Streaming	29
Live Event Broadcasting	30
Simulated Live Event Broadcasting	30
Summary	31
Linking to RealSystem Content	32
Working with Other Webcasting Professionals	32
RealServer Features	34
Using RealServer Features Together	36
License Files	38
Information Stored in the License	38
Reading the License	38
Minimum Licensed Features	39
4 SOURCES OF CONTENT	41
Overview	41
Creating an On-Demand Source with RealProducer Plus	42
Step 1: Creating the Clip	42
Step 2: Copying the Clip to RealServer	43
Step 3: Linking to the On-Demand Clip	44
Creating a Live Source with RealProducer Plus	44
Step 1: Starting the Live Encoding with RealProducer Plus	45
Step 2: Linking to the Live Event	46
Creating a Live Source with G2SLTA	48
When to Use G2SLTA	49
G2SLTA Used with Other Features	49
Setting Up and Running G2SLTA	51
Stopping G2SLTA	57
Optional G2SLTA Features	58
Using G2SLTA for Splitting and Multicasting	60
Files Required by G2SLTA	61
Working with Redundant Sources	61
Redundant Sources Used with Other Features	61
Setting Up Redundant Sources	63
5 UNDERSTANDING LINK FORMATS	67
Overview	67
Parts of a Link	68
Protocol	69
Address	70
Port	70
Mount Point	71

	Path	73
	File	74
	Sharing Information for Links	74
	Metafiles.....	74
	Ram Files and Ramgen	74
	SMIL Files	76
	Where to Put On-Demand Clips.....	77
	Where to Put Live Clips.....	83
6	STARTING AND STOPPING REALSERVER	85
	Windows	85
	UNIX.....	88
7	CUSTOMIZING REALSERVER FEATURES	91
	Overview.....	91
	Customizing RealServer Using RealSystem Administrator.....	91
	Starting RealSystem Administrator.....	92
	Using RealSystem Administrator	93
	Restricting Access to RealSystem Administrator	93
	Configuration File	94
	Common Settings	95
	Port Numbers	95
	Mount Points.....	96
	MIME Types	97
8	ADVANCED FEATURES	99
	Displaying Source Code for SMIL Files and Media Clips	99
	View Source Used with Other Features.....	101
	Changing View Source Settings	101
	Optional View Source Features	101
	Browsing Your Content.....	104
	RealServer Caching Features	104
	Caching used with Other Features and with RealProxy.....	105
	Changing Cache Settings	107
	Optional Caching Features	107
	Distributed Licensing.....	109
	Setting Up Distributed Licensing	109
	Monitoring Distributed Licensing.....	111
	Reserving IP Addresses for RealServer's Use.....	111
	Running Web Servers and RealServer on the Same System	112
	Port Hinting.....	113
	Features Specific to the Operating System	114
	Windows NT-Only Features	114

	UNIX-Only Features	115
9	FIREWALLS AND REALSERVER	117
	Overview	117
	Who Should Read This Chapter.....	117
	Highlights of This Chapter.....	118
	Firewalls and Their Interaction with RealServer Features ...	119
	Protocols Used by RealServer	120
	How Firewalls Can Affect the User Experience	121
	Communicating with Software Behind Firewalls.....	123
	Communicating with Clients	123
	Communicating with Encoders	126
	Communicating with Receivers	127
	Communicating with RealProxys.....	127
	Firewall Configurations (for Firewall Administrators)	128
	Application-Level Proxy Firewall	128
	Transparent Proxy Firewall	129
	Packet Filter Firewall	129
	Stateful Packet Filtering Firewall.....	129
	SOCKS Firewall.....	130
	Network Address Translation Firewall	130
	Summary of Firewall Types and Characteristics.....	131
	Best Firewall Configurations	131
	Ports Used in Streaming and Unicasting	132
10	STREAMING ON-DEMAND PRESENTATIONS	137
	Overview	137
	When to Use Streaming.....	137
	On-Demand Streaming Used with Other Features.....	138
	Storing On-Demand Clips	139
	Streaming On-Demand Clips	139
	RealServer Settings	140
	Linking to On-Demand Clips	141
	Working with SureStream Clips.....	141
11	UNICASTING LIVE PRESENTATIONS	143
	Overview	143
	When to Use Live Unicasting	144
	Live Unicasting Used with Other Features	144
	Unicasting Live Clips	146
	Configuring RealServer for Live Unicasting.....	146
	Creating the Link to the Live Unicast	148
	Optional Live Unicasting Features	149

	Archiving Live Broadcasts	150
	When to Use Live Archiving.....	151
	Live Archiving Used with Other Features	151
	Setting Up Live Archiving	152
	Disabling Live Archiving.....	153
	Linking to Archived Files	153
	Optional Live Archiving Features	154
12	SPLITTING LIVE PRESENTATIONS	159
	Overview.....	159
	Example Splitting Scenario.....	160
	Splitting Used with Other Features	160
	Calculating Latency and Bandwidth	162
	Compatibility with Earlier Versions of RealServer.....	164
	Setting Up Splitting	165
	Step 1: Setting Up the Transmitter	165
	Step 2: Setting Up the Receiver.....	167
	Step 3: Linking to Split Content	169
	Optional Configurations.....	170
	Broadcasting Over Multiple Network Paths	171
	Using Receivers as Transmitters.....	171
	Using Backup Transmitters	172
	Chaining Receivers	173
	Pull Splitting.....	176
13	MULTICASTING LIVE PRESENTATIONS	181
	Overview.....	181
	When to Use Multicasting	182
	RealServer Multicasting Methods	182
	Back-Channel Multicasting	182
	Scalable Multicasting	184
	Choosing the Method of Multicasting	185
	Multicasting Used with Other Features	186
	Additional Resources.....	190
	Setting Up Both Types of Multicasting.....	191
	Setting Up the Network for Multicasting.....	191
	Allocating Addresses and Ports in RealServer	191
	Publicizing Your Multicasts	195
	Multicasting with Multiple Network Interface Cards.....	196
	Setting Up Back-Channel Multicasting	196
	Configuring RealServer for Back-Channel Multicasting	196
	Linking to Back-Channel Multicasts	198
	Optional Back-Channel Multicasting Features	199

	Setting Up Scalable Multicasting.....	200
	Settings Used in Scalable Multicast.....	201
	Setting Up a Live Channel.....	201
	Linking to Scalable Multicasts.....	203
	Optional Scalable Multicast Features.....	204
14	LIMITING ACCESS TO REALSERVER.....	211
	Overview.....	211
	Controlling Access to HTTP Streams.....	212
	Limiting Access by Number of Connections or Bandwidth....	212
	Limiting Access by RealPlayer Version.....	213
	Limiting Access to Back-Channel Multicast Reception.....	213
	Limiting Access with the IP Address.....	214
	Access Control Used with Other Features.....	215
	Understanding Access Control Rules.....	216
	Setting Up IP Access Control.....	218
15	AUTHENTICATING REALSERVER USERS.....	223
	Overview.....	223
	When to Use Authentication.....	224
	Authentication Used with Other Features.....	224
	Adding User Names and Passwords.....	226
	Authenticating Encoder Users.....	227
	Authenticating RealSystem Administrator Users.....	228
	Authenticating Content Users.....	228
	Step 1: Adding User Names and Passwords.....	229
	Step 2: Giving Users Access to Content.....	229
	Step 3: Creating the Content.....	231
	Step 4: Linking to Authenticated Content.....	231
	Optional Content User Authentication Features.....	232
	Optional Authentication Features (for All Types of Users)....	238
16	STORING AUTHENTICATION DATA.....	245
	Overview.....	245
	Using Text Files for Authentication Data.....	246
	Using a Database for Authentication Data.....	250
	Setting Up Other Types of Data Storage.....	253
17	ISP HOSTING.....	255
	Overview.....	255
	Links to Users' Hosted Content.....	255
	Account Information.....	256
	ISP Hosting Used with Other Features.....	257

	Tracking Account Usage	258
	Dedicating RealServer to ISP Hosting	259
	Compatibility with Earlier Versions of RealServer.....	260
	Example ISP Hosting Scenario—Northwest ISP.....	260
	Users' Directory Structures.....	261
	Setting Up ISP Hosting	262
	Step 1: Creating the User List	262
	Step 2: Configuring RealServer	267
	Step 3: Linking to ISP Content	269
18	MONITORING REALSERVER ACTIVITY	273
	Java Monitor.....	273
	Java Monitor Used with Other Features	274
	Using Java Monitor	275
	Configuring Java Monitor Settings.....	275
	Optional Java Monitor Features	275
	Windows NT Performance Monitor.....	280
19	REPORTING REALSERVER ACTIVITY	283
	Access Log	283
	Access Log Files Used with Other Features	283
	Reading an Access Log	287
	Customizing Information Reported by the Access Log	297
	Using the GET Statement to Identify Delivery Method	300
	Error Log	303
	Log File Rolling	304
	Cached Requests Log	305
20	STREAMING TARGETED ADS	307
	How Ad Streaming Works.....	307
	Quick Start for Testing Ad Banner Insertion	308
	General Steps for Setting Up Ad Streaming	309
	Getting Ad URLs from an Ad Server.....	310
	Understanding Ad Types.....	310
	Guidelines for Ads in Streaming Presentations	311
	Integrating RealServer Directly with an Ad Server.....	312
	Setting up a Target HTML Page on a Web Server	313
	Requesting SMIL Files from an Ad Server	314
	Configuring RealServer to Stream Ads	315
	Understanding Ad Streaming Mount Points.....	316
	Creating Ad Streaming Mount Points	318
	Setting Up Rotating Banner Ads.....	322
	Changing Timeouts Values.....	323

	Overriding Mount Point Settings through SMIL	325
	Overriding the Target URL Location	325
	Overriding Banner Rotation Settings	325
	Generating SMIL Files for Ads	326
	Limitations on Automatic SMIL Generation.....	326
	Understanding SMIL Generation Mount Points	327
	Creating SMIL Generation Mount Points.....	328
	Setting SMIL Options	330
21	TROUBLESHOOTING	333
	Overview	333
	General Troubleshooting Steps	333
	Step 1: Make sure RealServer is running.....	333
	Step 2: Try different ways of connecting.	336
	Step 3: Check the production tools.	337
	Step 4: Check the remaining areas.....	337
	Step 5: Work with the system or network administrator...	338
	Troubleshooting RealSystem Administrator.....	338
	Troubleshooting On-Demand Streaming.....	339
	Troubleshooting Live Unicasting	340
	Troubleshooting Live Archiving	341
	Troubleshooting G2SLTA	341
	Troubleshooting Splitting.....	341
	Troubleshooting Multicasting.....	343
	Troubleshooting Access Control	345
	Troubleshooting Authentication	345
	Troubleshooting Monitoring.....	346
	Troubleshooting Ad Streaming	346
	Troubleshooting SMIL File Issues	348
	Troubleshooting Other Issues	350
	Troubleshooting Problems in RealPlayer.....	351
	Common Mistakes to Avoid	354
	Contacting RealNetworks Technical Support.....	355
	Determining the Server Version	357
A	SUMMARY OF LINK FORMATS	359
	The Subject of a Link.....	359
	How Authenticated Content Is Different	360
	Multiple Mount Points in Links.....	360
	Port Numbers in Links.....	361
B	CONFIGURATION FILE SYNTAX	379
	Configuration File Components.....	379

XML Declaration Tag	379
Comment Tags	379
List Tags	380
Variable Tags	380
C CONFIGURATION FILE CONTENTS	383
Editing the Configuration File	383
RealSystem Administrator and the Configuration File.....	384
RealServer Features in the Configuration File.....	385
Ad Streaming.....	385
Access Control	385
Allowance.....	386
Authentication and Commerce	387
Broadcast Redundancy.....	395
Caching.....	396
Encoders	397
File Systems (FSMount)	399
HTTP Support	400
ISP Hosting.....	402
IP Binding.....	408
Distributed Licensing.....	408
Live Archiving.....	408
Logging	411
MIME Types	413
Multicasting	413
Passwords	418
Paths.....	418
Ports	420
Ramgen.....	420
RealSystem Administrator.....	421
Splitting.....	424
UNIX-Only Settings	429
View Source	430
Features Available Only Through Direct Editing.....	433
INDEX	435



INTRODUCTION

Welcome to RealNetworks® RealServer™, the most powerful server software available for streaming media files across an intranet or the Internet. This manual will help you use and optimize RealServer for real-time delivery of multimedia files.

Overview

This guide is intended for technical system administrators who will be managing RealServer and its activities, but not necessarily creating the material to be streamed. Information on creating content is available in a companion book, the *RealSystem Production Guide*.

Information services professionals, server administrators, Web masters, and others who provide Web pages for the Internet and for intranets may also find this book useful.

RealServer Administration Guide is also available online at <http://service.real.com/help/library/index.html>.

How This Manual Is Organized

This book contains the following chapters:

Chapter 1, “Quick Start”

This chapter gives step-by-step instructions for getting RealServer started and running quickly.

Chapter 2, “What’s New in RealServer 8?”

If you’re familiar with previous versions of RealSystem®, this chapter will give you a quick update on the new features in RealServer version 8.

Chapter 3, “Overview of RealServer”

This chapter presents the “big picture” of how RealServer works with a Web server to stream media to client software such as RealNetworks RealPlayer®.

Chapter 4, “Sources of Content”

To serve clips to users, you first need to get the content. This chapter describes two tools for creating content, RealProducer® Plus and G2SLTA.

Chapter 5, “Understanding Link Formats”

This chapter describes how to construct the links to your content.

Chapter 6, “Starting and Stopping RealServer”

This chapter is a guide to starting and stopping RealServer. Depending on which platform you run RealServer on, different automatic options are available.

Chapter 7, “Customizing RealServer Features”

Modifying RealServer by changing settings in the configuration file is the key to fine-tuning RealServer features. Whether you use the RealSystem Administrator or edit the configuration file directly, this chapter describes how to make changes to RealServer.

Chapter 8, “Advanced Features”

This chapter discusses media caches, firewalls, the assignment of IP addresses for RealServer’s use, and differences between RealServer running on different platforms.

Chapter 9, “Firewalls and RealServer”

If you’re delivering content to users on the Internet, you’ll want to know how RealServer and other RealSystem products interact with firewalls. This chapter provides that information.

Chapter 10, “Streaming On-Demand Presentations”

This chapter gives instructions for delivering prerecorded or prepared clips.

Chapter 11, “Unicasting Live Presentations”

Live clips are streamed much like static clips, but with a few differences. Use this chapter to learn how to make broadcasting work as smoothly as possible.

Chapter 12, “Splitting Live Presentations”

This chapter explains how splitting can help you make the best use of bandwidth and provides the highest possible quality of reception.

Chapter 13, “Multicasting Live Presentations”

This chapter discusses multicasting, which is a way of sending a single live stream to multiple clients, rather than sending a separate stream to each individual client. Clients connect to the stream rather than to the RealServer computer.

Chapter 14, “Limiting Access to RealServer”

This chapter shows you how you can limit access to RealServer by specifying restrictions such as maximum bandwidth and IP addresses.

Chapter 15, “Authenticating RealServer Users”

With RealServer, you can control and limit who can view your content. This chapter describes the different RealServer authentication methods and the advantages of each one.

Chapter 16, “Storing Authentication Data”

RealServer comes with some different methods for tracking authentication information, as described in this chapter. You can use this data for billing purposes or to track who’s watching what.

Chapter 17, “ISP Hosting”

If you’re an Internet service provider (ISP), you can host streaming media on behalf of your customers. This chapter explains how.

Chapter 18, “Monitoring RealServer Activity”

To provide the highest possible quality of service, you’ll likely want to keep track of how many people are accessing your RealServer. This chapter describes the different methods of tracking RealServer activity.

Chapter 19, “Reporting RealServer Activity”

At some point, you’ll want to look at trends and see what content is most popular. RealServer can report player behavior with a customizable degree of detail. As this chapter explains, errors are reported in their own log, which can help you troubleshoot any problems that might arise.

Chapter 20, “Streaming Targeted Ads”

You can have RealServer automatically insert advertisements into presentations. This chapter describes the many options available within this feature.

Chapter 21, “Troubleshooting”

Running into problems? This chapter provides helpful steps you can take when you’re not sure what’s wrong. It also lists error messages and tells you what to do about them.

Appendixes

Appendix A, “Summary of Link Formats”

This appendix recaps the structure of URLs for all of the different types of content and delivery formats for streamed media.

Appendix B, “Configuration File Syntax”

This appendix presents a discussion of the XML syntax used by the configuration file.

Appendix C, “Configuration File Contents”

For those who prefer to edit the configuration file directly rather than use RealSystem Administrator, this is a guide to the configuration file contents.

Conventions Used in This Manual

This section explains some conventional terms and formats used throughout the book.

Terminology

Because this manual is aimed at the RealServer administrator, the term “you” refers to the administrator. Those who play clips served by RealServer are referred to as “visitors,” “viewers,” or “users.”

RealSystem clients, such as computers running RealPlayer, are referred to generically as “clients”. Wherever information applies specifically to RealPlayer or RealPlayer Plus, this is clearly stated. Although most clients currently in use are computers running RealPlayer, RealNetworks also makes a software development kit (SDK) that enables other companies to develop their own players with which they can receive the various types of streamed data.

RealSystem production tools, which are used to create the files and data that RealServer streams, are referred to simply as “encoders.”

“Clips,” “content,” “media files,” “files”, and “presentations” are used interchangeably to indicate the material that RealServer streams.

The following table explains the typographic conventions used in this manual.

Notational Conventions

Convention	Meaning
<code>syntax</code>	This font is used for syntax of configuration files, URLs, or command-line instructions.
<i>variables</i>	Italic text represents variables. Substitute values appropriate for your system.
emphasis	Bold text is used for emphasis.
...	Ellipses indicate nonessential information omitted from examples.
[]	Square brackets indicate optional material. If you choose to use the material within the brackets, don't type the brackets themselves. An exception to this is in the access log, where statistics generated by the StatsMask variable are enclosed in regular brackets.

Sample Links

Links that point to RealServer take the following form:

realserver.example.com

where:

- *realserver* is a placeholder for the name of the computer on which you are running your RealServer. Substitute the name of your organization's computer wherever you see this syntax.
- *example.com* is a placeholder for a domain name. Substitute the domain name of your organization's computers wherever you see this syntax.

Default Locations and Values

In all of the examples given in this book, it's assumed that you've installed RealServer in the default location for your operating system and that you're using default values for all settings. Of course, you can customize RealServer however you want to meet your specific needs; default values are used here for clarity of illustration.

On Windows-based platforms, the default installation directory is C:\Program Files\Real\RealServer. On UNIX-based platforms, the default installation directory is /usr/local/RealServer.

Available Features

Depending on which RealServer product you purchased, some of the features described in this manual may not be available to you or may be limited in some way (such as the number of streams you can transmit simultaneously). Consult your license file for a list of which features are enabled in your version of RealServer. If you'd like to augment your RealServer's capabilities, contact RealNetworks or your reseller.

Additional Information

For instructions on reading license files with RealSystem Administrator, see "License Files" on page 38.

Additional RealSystem Resources

In addition to this manual, you may need one or more of the following RealNetworks resources, which are available at <http://service.real.com/help/library/index.html>.

- RealServer Readme File

This file contains supplemental information not covered in this guide. It is available at <http://service.real.com/help/library/guides/server8/readme.html>.

- *RealSystem Production Guide*

This manual explains the basics of creating streaming files with the RealSystem tools. You'll learn how to calculate bandwidth needs and how to put a multimedia presentation together. To view this guide, click **Resources** under **Help** in RealSystem Administrator.

- *RealProxy Administration Guide*

If you're using RealNetworks RealProxy™, or are working with someone who is, this manual describes the use of RealProxy and configuration information.

- RealSystem Software Development Kit (SDK)

RealNetworks has developed a software development kit (SDK) that enables you to integrate applications with RealSystem or create new plugins for RealServer or RealPlayer. Programming knowledge is a prerequisite

for using the SDK. Register for and download the SDK from <http://www.realnetworks.com/devzone/>.

Technical Support

General troubleshooting steps and information about contacting RealNetworks for technical support are given in Chapter 21, “Troubleshooting”.

Chapter 1

QUICK START

This chapter gives step-by-step instructions for getting RealServer started and running quickly.

Overview

In this chapter, you'll learn how to:

- Start RealServer
- Use RealSystem Administrator to test RealServer
- Play sample files
- Create and stream your own on-demand clips
- Create and broadcast live events

Prerequisites

Instructions in this chapter assume that you've already installed RealServer. You'll also need the following software and hardware, although it's not necessary that they reside on the same computer as RealServer:

- RealPlayer version 7 or later
- Multimedia equipment: CD player, sound card, speakers, and software that allows you to play and hear music CDs
- RealProducer Plus version 7 or later
- A text editor for creating an HTML page (optional)
- A web browser (optional)

Note that for playing sample clips, you'll need both a second computer with RealPlayer installed and the multimedia equipment referenced in the preceding list.

RealProducer Plus and RealPlayer are included with some RealServer packages, and they are also available in free download versions from the RealNetworks Web site at <http://www.realnetworks.com>.

Starting RealServer

The two most common methods for starting RealServer are described in the following two paragraphs. For details on these, and for other startup options, see Chapter 6, “Starting and Stopping RealServer”. If you cannot get your RealServer to start, consult Chapter 21, “Troubleshooting”.

Windows NT

When you install RealServer on Windows NT, by default it installs itself as a service and runs automatically. If it isn’t running, click the Windows NT **Start** button, and then click **Programs>Real>RealServer 8**.

UNIX

From the main RealServer directory, type the following:

```
Bin/rmservice rmserver.cfg
```

Using RealSystem Administrator to Test Your RealServer

RealSystem Administrator is the Web-based system you’ll use to manage RealServer. From RealSystem Administrator, you can customize RealServer, play sample clips, and monitor activity.

Note

After you install RealServer, RealSystem Administrator automatically appears. If RealSystem Administrator is currently running, you can skip this section.

If you cannot get your RealSystem Administrator to start, refer to Chapter 21, “Troubleshooting”.

► To use RealSystem Administrator:

1. Start a Web browser from anywhere on your network.
2. In the browser’s address or location box, type the following URL, substituting your values for *address* and *AdminPort*:

```
http://address:AdminPort/admin/index.html
```

The setup program generates a random value for AdminPort if you didn't supply one. If you're not sure what number to use, refer to "How do I figure out which port number to use for RealSystem Administrator?" on page 338.

3. You're prompted for your user name and password. Use the same ones you created during setup.

If you don't remember your user name or password, see "How do I look up my user name and password?" on page 338.

4. Click **OK**.

RealSystem Administrator starts.

Playing Sample Files

In the left-hand frame in RealSystem Administrator, click **Samples**. A new frame appears on the right, with links to sample clips. Click any one of these to test RealServer. Note that the computer you're using needs a sound card and speakers so that you can hear the audio portion of the clips.

- To see RealVideo[®], click **Play** in the **RealVideo 8.0** area.
- To hear RealAudio[®], click **Play** in the **RealAudio 8.0** area.
- To see a SMIL presentation that uses RealPix[™] and RealText[®], click the first **Play** button in the **RealPix, RealText, and SMIL** area.

If the clip plays correctly, you're ready to begin streaming! If you encounter any difficulties, consult Chapter 21, "Troubleshooting".

You can also play any sample clip by typing its address in RealPlayer. Start RealPlayer, and click **File>Open Location**, and then type one of the following:

- To play a RealVideo clip, type `rtsp://address:554/real8video.rm`
- To play a RealAudio clip, type `rtsp://address:554/real8audio.rm`
- To play a SMIL presentation, type
`rtsp://address:554/presentation/presentation.smil`

where *address* in each case is your computer's IP address or Domain Name System (DNS) name. Click **OK** to play the presentation.

As you'll see later in this chapter, the format for URLs you type in RealPlayer is slightly different from the format used in the HTML code for Web pages.

Examples of both formats are given in each chapter throughout this manual.

Creating and Streaming Your Own On-Demand Clips

This section explains how to create a very simple music file and then stream it from RealServer. To do this, you'll need a music CD and RealProducer Plus. Other versions of RealProducer Plus may involve slightly different steps than the ones shown in the following procedures; if you have a different version, use these steps as a general guide.

Step 1: Creating a Music Clip

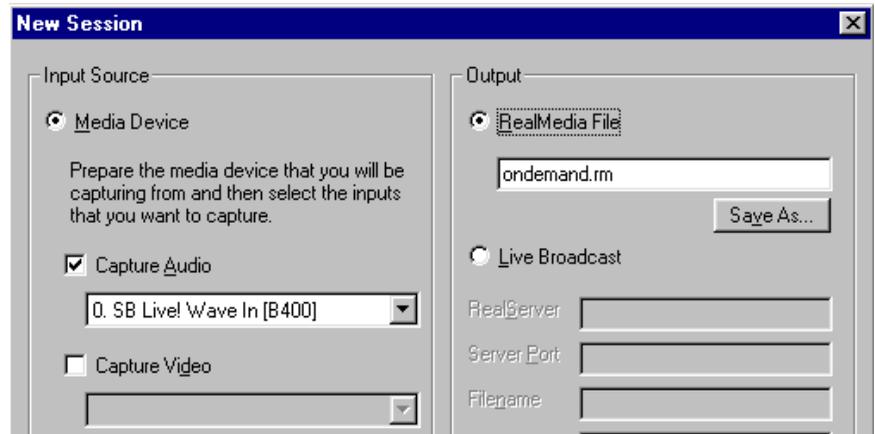
In this example, you'll encode your music CD directly to a file and then move it to a location from which RealServer can stream it. There are many ways you can optimize the encoding to create the best possible listening experience; these instructions have been kept brief so that you can hear results quickly.

Additional Information

To learn more about options for encoding, refer to the *RealProducer Plus User's Guide*, which is available at <http://service.real.com/help/library/index.html>.

► To create the music clip:

1. Put a music CD in your computer's CD player, and then start playing it, using the CD player. (Don't use RealJukebox, as it won't initialize the audio device needed for encoding.)
2. Start RealProducer Plus.
 - In RealProducer Plus for Windows or the Macintosh, the **New Session-Choose Recording Wizard** dialog box appears. Select the **Don't Use Recording Wizards** check box. Click **OK**, and then click **Cancel**.
 - In RealProducer Plus for UNIX, the main RealProducer Plus application is visible.
3. Click **File>New Session**. The **New Session** dialog box appears.
4. Under **Input Source**, select **Media Device**.
5. Select the **Capture Audio** check box, and clear the **Capture Video** check box.
6. Under **Output**, select **RealMedia File**.
7. In the text box below the option button, type the file name `ondemand.rm` (always use the `.rm` extension).



8. Click **OK**.

The **New Session** dialog box closes, returning you to the RealProducer Plus main window.

9. Verify that **Multi-rate SureStream** is selected.

10. Under **Target Audience**, make sure that **28K Modem** and **56K Modem** check boxes are selected.

11. Leave all other text boxes blank.

12. Click **Controls>Start**.

A message appears, asking whether you want to add clip information.

13. Click **No**.

RealProducer Plus begins recording your music CD. The word “Encoding” appears in the lower-left corner.

14. Wait at least one minute, and then click **Stop**.

A message appears, asking whether you want to stop encoding.

15. Click **Yes**.

16. Click **Close**.

17. Click **Yes**.

RealProducer Plus halts the recording and creates a file named `ondemand.rm` in the RealProducer Plus directory.

Step 2: Putting the Music Clip in the Content Directory

Copy the `ondemand.rm` clip you created in the preceding section, from its current location in the main RealProducer directory to the RealServer Content directory.

On Windows NT, the path is `C:\Program Files\Real\RealServer\Content`.

On UNIX, the path is `/usr/local/RealServer/Content`.

Step 3: Creating a Link (Optional)

Create a link for the clip in a Web page. (The Web page can be local; it doesn't have to be on a remote Web server.)

In a Web page, type the following link and save the page (substitute your RealServer's computer name or IP address for *address*):

```
<a href="http://address:8080/ramgen/ondemand.rm">Click here to listen to my CD</a>
```

The word "ramgen" tells RealServer to instruct the Web browser to start RealPlayer. The Ramgen feature is described in "Ram Files and Ramgen" on page 74.

Step 4: Playing the Sample Clip

If you added the link to the Web page in Step 3, use a Web browser to view the page. Click the link that says "Click here to listen to my CD". When RealPlayer starts, you'll hear the clip you created.

To play this in RealPlayer without using a Web page, start RealPlayer and then click **File>Open Location**. Type the following URL:

```
rtsp://address:554/ondemand.rm
```

and then click **OK**.

Creating and Broadcasting Live Events

In this section, you'll encode your music CD directly to RealServer and broadcast it while it encodes. You'll listen to it from a second computer.

Follow the instructions in this section to create a demonstration audio clip, using a music CD and RealProducer Plus. Other versions of RealProducer Plus may involve slightly different steps than the ones shown in the following procedures; if you have a different version, use these steps as a general guide.

Step 1: Encoding an Event

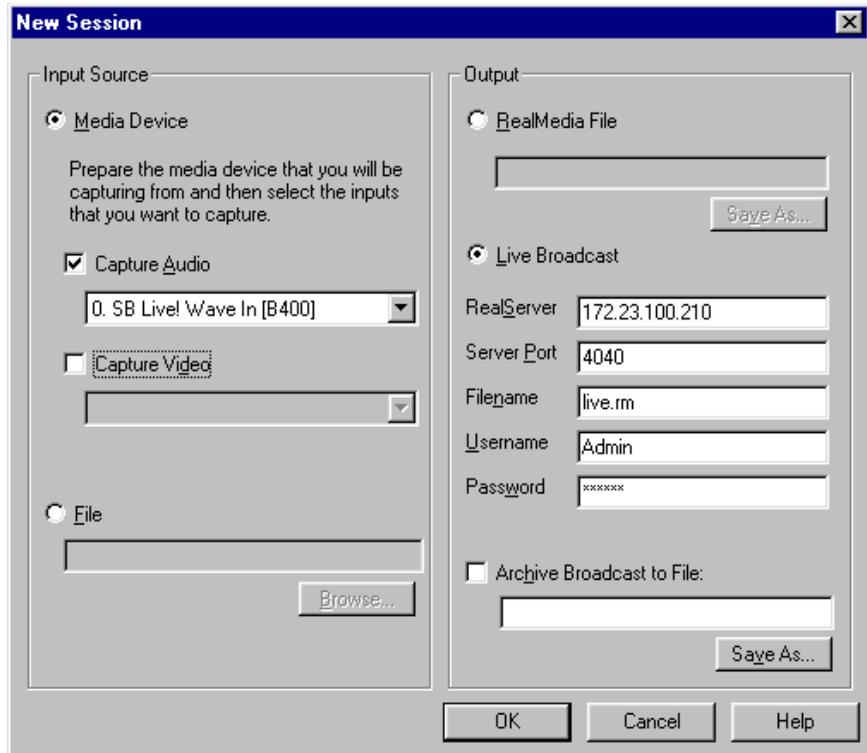
There are many ways you can optimize encoding to produce the best possible listening experience; these instructions have been kept brief so that you can hear results quickly.

Additional Information

To learn more about options for encoding, see the *RealProducer Plus User's Guide*, which is available at <http://service.real.com/help/library/index.html>

- ▶ To create a live clip:
 1. Put a music CD in your computer's CD player, and then start playing it, using the CD player. (Don't use RealJukebox, as it won't initialize the audio device needed for encoding.)
 2. Start RealProducer Plus.
 - In RealProducer Plus for Windows or the Macintosh, the **New Session-Choose Recording Wizard** dialog box appears. Select the **Don't Use Recording Wizards** check box. Click **OK**, and then click **Cancel**.
 - In RealProducer Plus for UNIX, the main RealProducer Plus application is visible.
 3. Click **File>New Session**. The **New Session** dialog box appears.
 4. Under **Input Source**, select **Media Device**.
 5. Select the **Capture Audio** check box, and clear the **Capture Video** check box.
 6. Under **Output**, select **Live Broadcast**.
 - a. In the **RealServer** box, type the IP address of the computer on which you installed RealServer.

- b. In the **Server Port** box, leave the default setting 4040.
- c. In the **Filename** box, type live.rm (always use the .rm extension).
- d. In the **Username** box, type the same user name you use for logging in to RealSystem Administrator.
- e. In the **Password** box, type the password you use for RealSystem Administrator.



7. Click **OK**.
The New Session dialog box closes, returning you to the RealProducer Plus main window.
8. Verify that **Multi-rate SureStream** is selected.
9. Under **Target Audience**, make sure that the **28K Modem** and **56K Modem** check boxes are selected.
10. Leave all other text boxes blank.
11. Click **Controls>Start**.

A message appears, asking whether you want to add clip information.

12. Click **No**.

RealProducer Plus begins encoding your music CD.

Step 2: Creating a Link (Optional)

Create a link for the live broadcast in a Web page. (The Web page can be local; it doesn't have to be on a remote Web server.)

In an existing Web page, type the following link and save the page (substitute your RealServer name or IP address for *address*):

```
<a href="http://address:8080/ramgen/encoder/live.rm">Click here to listen to my CD</a>
```

Step 3: Playing the Clip

Go to a different computer that has a sound card, speakers, and RealPlayer installed. Using a Web browser, view the page that you just edited. Click the link that says "Click here to listen to my CD". RealPlayer starts, and you'll join the music broadcast in progress.

To play this in RealPlayer without using a Web page, start RealPlayer and then click **File>Open Location**. Type the following URL:

```
rtsp://address:554/encoder/live.rm
```

and then click **OK**.

The word "encoder" tells RealServer to look for live input from RealProducer Plus or other encoding software.

Notice that there are a few seconds of delay on your computer. Encoding isn't instantaneous, also the music must travel over the network to the second computer. This delay ensures reliability and cannot be eliminated.

When you've finished with this demonstration, click **Stop** in RealProducer Plus to halt the live encoding.

Chapter 2

WHAT'S NEW IN REALSERVER 8?

RealServer 8 is designed on a new architecture that facilitates greater extensibility and interoperability with third-party solutions. This chapter discusses the features that have been added to this version of RealServer, the features that were new in version 7, and the compatibility of version 8 with earlier versions of the product.

New Features in RealServer Version 8

The following features are making their first appearance in this version of RealServer.

Distributed Licensing

This feature enables a set of RealServers within an organization to use the same license file. Features can be configured independently on individual RealServers, while sharing a pool of connections. For more information, see “Distributed Licensing” in Chapter 8.

Port Hinting

For those clients that can receive RealServer content only by way of the method known as “HTTP cloaking”, you can now create URLs that list the ports to try. This way, the client doesn’t waste time by trying to discern the correct port to use. For more information, see “Port Hinting” in Chapter 8.

Redundant Encoders

You can now use parallel sources as input for RealServer. Should one stream become unavailable, RealServer will automatically switch all users to the next available stream. For more information, see Chapter 4, “Sources of Content”.

Splitting

The splitting feature has been completely redesigned to provide greater scalability and reliability. Most notably, multicasting between transmitter and receiver is now possible. For more information, see Chapter 12, “Splitting Live Presentations”. If you’ve used splitting in earlier versions of RealServer, see “Compatibility with Earlier Versions of RealServer” on page 164 for special instructions.

Stream Encryption

RealServer 8 comes with an optional stream encryption feature; start RealSystem Administrator and click **Security>Encryption** for more details.

Support for Additional Data Types

RealServer 8 supports even more data types than before: MPEG Audio (MP3) and Real G2 with Flash version 4 can be streamed from this version of RealServer. QuickTime version 4 files can be streamed to Apple’s QuickTime 4 player. Refer to the Readme file, located at <http://service.real.com/help/library/guides/server8/readme.html>, for more information.

New Features in RealServer Version 7

This section lists and describes the features that were new to RealServer 7.

Ad Server Integration with RealServer

Dynamically add a streaming advertisement to a requested media clip. Seamlessly integrate RealServer with ad servers and services. For more information, see Chapter 20, “Streaming Targeted Ads”.

ISP Hosting Support

As in versions 3, 4, and 5, RealServer 7 and later segment streams and host content on behalf of other users. For more information, see Chapter 17, “ISP Hosting”.

Log Rolling for Both Access Log and Error Log Files

In RealServer 7, you can limit error logs to a specified size. For more information, see Chapter 19, “Reporting RealServer Activity”.

Multicast Shift to Unicast Feature

Scalable multicasting now includes a feature that enables clients to receive unicast transmissions if their multicast connections fail. See “Using Unicasting as a Backup Method” on page 205.

Pending Changes Page in RealSystem Administrator

As you make changes to RealServer using RealSystem Administrator, those modifications that require you to restart RealServer are listed on a Pending Changes page. The **Restart Server** button at the top of the RealSystem Administrator window changes color to indicate that new changes are ready to be implemented, and a **Pending Changes** button appears as well. For more information, see “Using RealSystem Administrator” on page 93.

SureStream Support for G2SLTA and Live File Archiving

Earlier versions of RealServer didn't archive SureStream files, nor could SureStream files be included in a simulated live event. For more information, see “Creating a Live Source with G2SLTA” on page 48.

View Source Code of SMIL Files

The view source feature enables users of RealPlayer 7 to view the source code for SMIL presentations or media clips. You can also browse the on-demand content available to your RealServer. For more information, see “Displaying Source Code for SMIL Files and Media Clips” on page 99.

Compatibility with Earlier Releases

RealServer version 8 is fully compatible with RealServer 3 and later, with the exception of the splitting feature. Changes to splitting are described in “Compatibility with Earlier Versions of RealServer” on page 164.

Presentations created with earlier versions of RealSystem tools still work seamlessly with RealServer 8; just be sure to place them in the Content directory. To use encoding software that was developed before RealSystem 6 (such as RealVideo Encoder 4), use the information provided in “Pre-G2 Encoders” on page 147.

To use new features (RealText, for instance) in an existing presentation, you must first update the presentation by creating a SMIL file and modifying the URL that refers to the presentation.

Chapter 3

OVERVIEW OF REALSERVER

Welcome to RealNetworks RealServer, the streaming media solution! RealServer streams audio, video, images, animation, text, and other data types to client computers. This newest version of RealServer has been designed to keep pace with your multimedia needs as they continue to change. This chapter introduces the core RealServer concepts and features.

To begin serving streamed content right away, see Chapter 1, “Quick Start”.

What Is RealServer?

RealServer is server software that streams both live and prerecorded media over a network. The streamed data can originate either on the Internet or within an intranet. The client receives the media in real time, and without having to wait for clips to be downloaded.

RealServer consists of the following components:

- **Executable file**—RealServer’s main component, called `rmserver.exe` for Windows-based platforms and `rmserver` for UNIX-based platforms.
- **Plug-ins**—files that provide the functionality of RealServer’s individual features. Because of this open architecture, third parties can create custom features, enabling you to extend the capabilities of your RealServer.
- **Configuration file**—a text file in XML format that stores all of your RealServer’s customized information. The configuration file is named `rmserver.cfg`.
- **License file**—one or more files that control the features enabled in your RealServer.
- **RealSystem Administrator**—a Web-based console for customizing and monitoring your RealServer.

- **Tools**—additional software such as the Java Monitor, which enables you to view how many clips are being served at any given time, and G2SLTA, which broadcasts prerecorded clips as if they were live events.
- **Other files**—this depends on the particular RealServer package you've installed. Your installation may have other files that perform additional functions, such as commerce or ISP hosting.

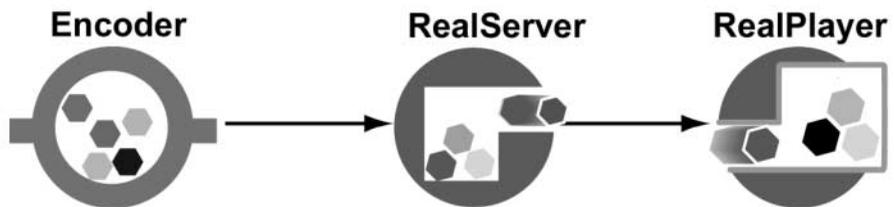
What Is RealSystem?

RealServer is a member of the RealSystem family of software tools. Three components make up RealSystem:

- **Production tools**—such as RealProducer Plus, to create media clips (such as audio, video, or animation). These are also called *encoders*.
- **RealServer**—to stream media files.
- **Client software**—such as RealPlayer, to play the streamed clips.

The following diagram illustrates how these RealSystem components work together.

RealSystem Components



Production Tools

The person who designs the content that you serve from your RealServer uses various production tools to create the content. These tools convert audio, video, or animation to a data type format that RealServer can stream.

The content creator can also create a Synchronized Multimedia Integration Language (SMIL) file to synchronize several clips within a presentation. A SMIL file coordinates the layout and playing of two or more media clips in parallel (simultaneously) or in sequence.

Because RealServer can deliver content in many different formats, there are a number of tools that people can use in creating content. Production tools can

optimize the content for efficient delivery over the Internet, based on the nature of the material and the capabilities of the client computers.

The content creator can either prepare media clips in advance or encode a live event as it happens. In this manual, we use the generic term *encoder* to refer to the software (such as RealProducer, for example) that converts live or preexisting media into a format that RealServer can deliver.

RealServer

Just as a Web server delivers pages to Web browsers over the Internet, RealServer serves media clips to clients (clips are created with the production tools described earlier in this chapter). It enables users to stream the media clips rather than download them. By streaming the content, the user can begin to watch the clip almost immediately and doesn't have to wait for the entire file to be downloaded.

Client Software

A client such as RealPlayer plays the streamed media.

Other Software

In addition to the RealSystem software, you may want to work with other software as well, such as:

- A Web server
- A Web browser
- Firewalls
- Networking software
- Database software, if you're using commerce authentication features
- An ad server or services, if you're using advertising features

How RealServer Works

RealServer streams media to clients over networks and the Internet. It's usually employed in conjunction with a Web server. Note that you can use some RealServer features with third-party products to create specialized functions, such as report analysis.

Channels and Protocols

RealServer uses two connections, known as *channels*, to communicate with clients: one for communication with the client, and one for actual data. The communication channel is known as the *control channel*, as it is over this line that RealServer requests and receives passwords and clients send instructions such as fast-forward, pause, and stop. Media clips themselves, on the other hand, are actually streamed over a separate *data channel*.

Every link to content begins with a protocol identifier, such as *rtsp*, *pnm*, or *http*.

RealServer uses two main protocols to communicate with clients: Real Time Streaming Protocol (RTSP) and Progressive Networks Audio (PNA).

Occasionally, RealServer will use HTTP for metafiles that point to RealServer content, and for HTML pages that it serves (such as the Web-based RealSystem Administrator). It may also be used to deliver clips to clients that are located behind firewalls.

Within these channels, RealServer uses two other protocols for sending instructions and data:

- **Transport Control Protocol (TCP)**—for sending commands from the client (such as “start” and “pause”) and sending commands from RealServer to clients for specific information (such as the clips’ titles).
- **User Datagram Protocol (UDP)**—for sending actual streamed content.

For more information on RealServer’s use of ports, see Chapter 9, “Firewalls and RealServer”.

Occasional Exceptions

Because many firewalls are configured to allow only TCP connections or HTTP traffic through them, you may need to make some adjustments to receive data from an encoder or to work with clients if there’s a firewall between the encoder or a given client and your RealServer. For more information, see Chapter 9, “Firewalls and RealServer”.

Communication Between Encoders and RealServer

When an encoder connects to RealServer and sends encoded data, it uses a one-way UDP connection to communicate with RealServer as illustrated in the following diagram.

UDP Connection Between Encoder and RealServer



Some firewalls don't permit UDP packets, so RealProducer includes a setting that uses two-way TCP connections to send the same encoded content, as many firewalls do allow TCP traffic to pass through. This type of connection is illustrated in the following diagram.

TCP Connection Between Encoder and RealServer



Communication Between RealServer and RealPlayers

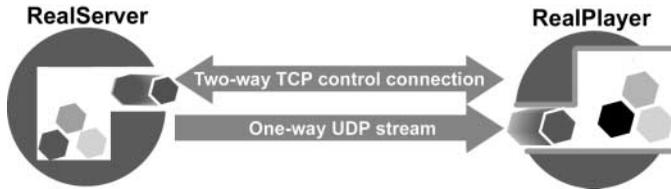
When the user clicks a link that points to a streaming media presentation, RealPlayer opens a two-way connection with RealServer. This connection uses TCP to send information back and forth between RealPlayer and RealServer as shown in the following diagram.

Initial TCP Control Connection



As RealServer approves the request, it sends the requested clip to a RealPlayer along a one-way UDP channel, as shown in the following diagram.

UDP Data Connection



As it receives the streamed clip, RealPlayer plays it at high fidelity.

Streaming Media Delivery Methods

There are two main ways that users access and experience media clips:

- **On-demand**—as with renting a video at a 24-hour video store, a clip is available to a given user whenever she wants it. The user can fast-forward, rewind, or pause the clip, and RealServer will send the right part of it. This type of clip is prerecorded or preassembled.
- **Live**—as with a live telecast of the Olympic Games, a user can tune in to the action that is happening at any given time. Note that a user can not fast-forward or rewind through the clip, because the event is happening in real time. Of course, the delivery of content as a live event requires an actual live event, and it's possible only if you or the content creator have the software and hardware needed to capture the content and convert it to a media format that RealServer can broadcast.

There's also a third, less common method, which uses on-demand clips but delivers them as if they were live:

- **Simulated live**—Just as television broadcasts sometimes record live events and then broadcast them later, such as Olympic sports that wouldn't be seen live everywhere because of time-zone differences, simulated live broadcasts take prerecorded events and broadcast them as live ones. Thus, although the content is prerecorded, users view the events as if they were live.

The following table summarizes the three types of user participation.

User Participation Comparisons		
On-demand (through streaming)	Live delivery (through unicasting, splitting, or multicasting)	Simulated live delivery (through unicasting, splitting, or multicasting)
Users can access presentations at any time.	Users can only access presentations while they're in-progress.	Users can only access presentations while they're in-progress.
Files are stored on disk.	Presentations don't exist as files.	Files are stored on disk.
Presentations always begin streaming at the beginning of the file.	Everyone sees the same part of the presentation at the same time—late-comers join in the middle.	Everyone sees the same part of the presentation at the same time—late-comers join in the middle.
Users can fast-forward through the clip or pause it at any point.	User plays the clip all the way through.	User plays the clip all the way through.
It's analogous to a videotape of past Olympic event highlights.	Similar to live television coverage of an Olympic event.	Similar to a previously recorded Olympic event, being delayed on television because of time-zone differences.

Choosing a Delivery Method

Once you've determined how you want the user to experience a given clip (on-demand or live), you choose the delivery method you'll configure RealServer to use:

- **On-demand**—The choice is simple: streaming is the only delivery method.
- **Live and simulated live**—There are three ways to deliver the clip: unicasting, splitting, or multicasting.

On-Demand Streaming

Prerecorded clips are delivered, or *streamed*, to users upon request. A user who clicks a link to an on-demand clip watches the clip from the beginning. The user can fast-forward, rewind, or pause the clip. For more information, see Chapter 10, "Streaming On-Demand Presentations".

Live Event Broadcasting

Live clips can be delivered in several different ways. As the administrator, you will decide which method to use, based on your network needs. A user who clicks a link to a live clip joins the live event in progress. Because the event is happening in real time, fast-forward, rewind, and pause capabilities are not available.

Live clips are broadcast as they're created. These clips don't exist as files, because they're created as the live event happens. You can save live content as files by using the live archiving feature. The archived files become on-demand content and are handled accordingly.

Unicasting

This is the simplest and most popular method of live broadcasting, as it requires little or no configuration. For more information, see Chapter 11, "Unicasting Live Presentations".

Splitting

Splitting is the term used to describe how one RealServer can share its live media streams with other RealServers. Clients connect to these other RealServers, called *splitters*, rather than to the main RealServer where the streams originate. Splitting reduces the traffic load on the source RealServer, enabling it to distribute other broadcasts simultaneously. This method moves the broadcasts closer to clients, improving the quality of service transmission for them. For more information, see Chapter 12, "Splitting Live Presentations".

Multicasting

Multicasting is a standardized method for delivering presentations to large numbers of users over a network or the Internet. For more information, see Chapter 13, "Multicasting Live Presentations".

Simulated Live Event Broadcasting

The same delivery options are available as for live broadcasting: unicasting, splitting, and multicasting. The only difference is that, with simulated live broadcasting, the event has already been recorded, and no connection to a production tool or encoder is needed. The **G2SLTA** program included with RealServer sends the on-demand file to RealServer the same as if it were an

actual live event. For more information, see “Creating a Live Source with G2SLTA” on page 48.

Summary

The following table lists the delivery methods, uses, and requirements for the different user participation styles.

Comparison of Delivery Methods			
User participation style	Delivery method	Appropriate use	Requirements
On-demand	Streaming	Presentations that are limited to the number of licensed connections, CPU speed, amount of RAM, and available bandwidth on a single computer.	Sufficient bandwidth to handle the number of connected clients.
Live and simulated live	Unicasting	Broadcasts that are limited to the number of licensed connections, CPU speed, amount of RAM, and available bandwidth on a single computer running RealServer.	Sufficient bandwidth to handle the number of connected clients.
	Splitting	Broadcasts that are limited to the number of licensed connections, CPU speed, amount of RAM, and available bandwidth on all of the connected computers running RealServer.	At least two RealServers. Both the transmitter and the receiver RealServers require configuration.
	Multicasting	Broadcasts that will be viewed by an unlimited number of users around the globe on a multicast-enabled network.	A multicast-enabled network. You can use multicasting in conjunction with splitting to cover a greater geographic region in which networks are not multicast-enabled.

In some cases, you can use more than one live delivery method for the same broadcast, to reach the maximum number of users while minimizing network bandwidth consumption.

- The combination of splitting and multicasting is described in “Splitting and Multicasting” on page 187.

- The combination of unicasting and multicasting is described in “Requiring Multicasting Access Rather Than Unicasting” on page 200 (for back-channel multicast) and “Using Unicasting as a Backup Method” on page 205 (for scalable multicast).

Linking to RealSystem Content

Links to media clips served by RealServer are made up of several components which tell RealServer where to look for each clip and how to serve it.

Content creators tend to embed most links in Web pages. A user looking at a content creator’s site will click the link, and through the process described in “How RealServer Works” earlier in this chapter, the user will receive the media.

For example, the following link for a RealVideo file would appear in a Web page (the URL for the media clip is in bold type):

```
<a href="http://realserver.example.com:8080/ramgen/Concerts/French/debussy.rm">Click here to watch today's concert!</a>
```

The clip may be prerecorded, live, or prerecorded but delivered as live.

Additional Information

For detailed instructions for creating links to RealSystem clips, see Chapter 5, “Understanding Link Formats”.

Working with Other Webcasting Professionals

This manual assumes that you (the RealServer administrator) are managing your RealServer, and that a second person (the content creator) is generating media clips and SMIL presentations and putting links in Web pages. In reality, you may be performing both roles, especially when you’re setting up a feature and you want to do some quick tests. For the purpose of this discussion, however, it’s easier to discuss the roles as being performed by different people.

The RealServer administrator needs to provide the content creator with certain types of information so that she can create the correct links in her

SMIL files and Web pages. If the content creator is encoding live material, she will need to know where to direct the live data.

Responsibilities of the RealServer Administrator and the Content Creator

RealServer administrator	Content creator
Configures and maintains RealServer.	Performs encoding and assembles presentations.
Supplies information needed to create Web links.	Creates Web links.

Content Creators of On-Demand Content

Content creators will need the following information:

- The location where they should place their files
- The address or name of the specified RealServer
- The port numbers for each protocol (but only if you've changed them from the recommended default settings)
- Information about whether Ramgen is being used (Ramgen is defined in "Ram Files and Ramgen" in Chapter 5).

Content Creators of Live Content

To encode a live stream for a specific RealServer, content creators need the following information:

- The address or name of the RealServer
- The port number to connect to
- Authentication information such as passwords, if required
- The URL to use in Web pages that point to a live broadcast or multicast
- The URL to use in a SMIL file

Other RealServer Administrators

RealServer can broadcast to other RealServers, which can redistribute the presentations to clients, thus reducing the load on the original RealServer. This feature is called *splitting*. If you're working with the administrator of the other RealServer (the receiver), you'll need to give that person certain information about your RealServer settings. That information is discussed in Chapter 12, "Splitting Live Presentations".

Firewall Administrators

If there are users within your network who either cannot receive presentations from RealServers on the Internet or who receive poor-quality streams, the information in Chapter 9, “Firewalls and RealServer”, will help the firewall administrator understand what changes can be made to enhance the users’ experience.

Network Administrators

In determining both how much bandwidth is available on your network and how much should be allotted to RealServer, network administrators can help you arrive at suitable numbers.

RealServer Features

In addition to the streaming media delivery methods described earlier in this chapter, there are many other RealServer features that benefit users and help you administer your RealServer.

Ad Streaming

With RealServer, you can have ads dynamically inserted into streaming presentations. Offering integration with any HTML-based ad serving system, RealServer uses SMIL to lay out ads and requested content in RealPlayer.

Access Control

The access control feature enables you to associate specific client addresses with the ability or permissions required to connect to specific ports.

Authentication

Authentication verifies the identity of a user or RealPlayer computer that’s requesting streamed media. This verification can take the form of asking for a name and password, or it can be entirely hidden from the user.

Firewalls

Firewalls aren’t specifically a RealServer feature, but they’re very important in networked environments. A firewall is a software program that monitors, and in some cases controls, all transmissions between an organization’s internal network (or intranet) and the Internet. Whether your network consists of a company’s local area networks (LANs), wide area networks (WANs), and the

Internet, the firewall's role is to ensure that all online communication, in both directions, conforms to your organization's security policies and practices.

ISP Hosting

RealServer works with your existing user accounts and directory structure to make users' media files available for streaming. You allocate a minimum and maximum number of connections for each account, based on the number of streams permitted by your license. Allocating on a per-connection basis, rather than by stream, ensures that all files (including SMIL files that reference multiple streams) will always be served.

Monitoring

RealSystem Administrator includes Java Monitor, a real-time tool that displays activity on your RealServer. It shows who's using your RealServer, the peak use times, which files are requested the most, and other information.

RealProxy

RealProxy is RealNetworks software that stores streamed media. Although it's not part of RealServer, RealProxy can work with RealServer to share the distribution load, thereby conserving bandwidth over an intranet and enabling RealServers to send streams to a wider audience. RealProxy is generally installed on an intranet or on a large ISP. When a client on the intranet or hosted by the ISP requests a streamed media file, RealProxy intercepts the request and sends it on behalf of the client. RealProxy then stores the requested media and streams it to any other clients who subsequently request the same material.

RealSystem Administrator

RealSystem Administrator is the Web-based console for customizing RealServer features. You can run it from any browser on your network. Note that it's password-protected when it's first installed; you can create additional user names and passwords for anyone else who'll be helping you administer your RealServer.

Redundancy

For important events for which parallel sources are required, RealServer can use multiple sources for the same event. Should one source become unavailable, RealServer will automatically switch to the next source.

Reporting (Log Files)

Using RealServer, you can create reports containing historical data that enable you to see trends and gather information. Track who visited your site and for how long, what clips they watched, and whether they watched them all the way through to the end. This information is stored in the access log. Any error messages that occur are recorded in the error log. Requests for streams that will be cached are stored in the cached requests log.

View Source

Just as users can right-click an HTML file and view the HTML code that was used to create a Web presentation, RealServer contains a feature that enables users of RealPlayer 7 and later to view the source code for SMIL presentations or information about media clips. When a user right-clicks a presentation or clicks **View>Clip Source**, RealServer sends the user's browser a Web page that contains the text of the SMIL file or data about the clip.

Using RealServer Features Together

You can use RealServer components in combination with one another to conserve bandwidth and deliver high-quality presentations. The following table summarizes RealServer features and how they work together.

For more information on exactly how any two of these features work in tandem, refer to the chapters that are devoted to those features.

Interoperability of RealServer Features

	Streaming	Unicasting	Archiving	Live simulation (G2SLTA)	Push splitting	Pull splitting	Back-Channel multicasting	Scalable multicasting	Redundancy	RealProxy access	Firewalls ¹	Access control	Authentication	ISP hosting	Monitoring	Reporting (log files)	Ad streaming
Delivery Methods																	
Streaming	Y	—	—	—	—	—	—	—	—	Y	Y	Y	Y	Y	Y	Y	Y
Unicasting	—	Y	Y	Y	Y	Y	*	*	Y	—	Y	Y	Y	—	Y	Y	Y
Archiving	—	Y	Y	*	—	—	*	*	Y	—	—	—	—	—	—	—	—
Live simulation (G2SLTA)	—	Y	*	Y	*	*	*	*	Y	—	—	Y	Y	—	Y	Y	Y
Push splitting	—	Y	—	*	Y	*	*	*	Y	—	Y	Y	Y	—	Y	Y	*
Pull splitting	—	Y	—	*	*	Y	*	*	Y	Y	Y	Y	Y	—	Y	Y	*
Back-channel multicasting	—	*	*	*	*	*	Y	—	—	—	Y	Y	Y	—	Y	Y	*
Scalable multicasting	—	*	*	*	*	*	—	Y	—	—	Y	Y	Y	—	—	*	*
Other Features																	
Redundancy	—	Y	Y	Y	Y	Y	—	—	Y	—	—	Y	Y	—	—	Y	—
RealProxy access	Y	—	—	—	—	Y	—	—	—	Y	—	Y	Y	Y	—	Y	Y
Firewalls ¹	Y	Y	—	Y	Y	Y	Y	Y	—	—	Y	Y	Y	Y	—	—	Y
Access control	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Authentication	Y	Y	*	Y	Y	Y	Y	Y	—	Y	Y	Y	Y	—	Y	Y	*
ISP hosting	Y	—	—	—	—	—	—	—	—	Y	Y	Y	—	Y	Y	Y	—
Monitoring	Y	Y	—	Y	Y	Y	Y	—	—	—	—	Y	Y	Y	Y	—	Y
Reporting (log files)	Y	Y	—	Y	Y	Y	Y	*	Y	Y	—	Y	Y	Y	—	Y	Y
Ad streaming	Y	Y	—	Y	*	*	*	*	—	Y	Y	Y	*	—	Y	Y	Y

Y Features work together automatically, with no additional configuration required beyond normal setup.

* Requires that special steps be taken for these features to work together.

— Not applicable; features are unrelated.

¹ Firewall information is based on the assumption that the firewalls allow streaming media traffic and multicast traffic.

License Files

The list of features available to your RealServer is stored in a file in the `License` directory. License files are written in XML format.

To upgrade your license so that you can use more of RealServer's features, contact RealNetworks or your reseller.

If your organization uses several RealServers that all use the same set of features, you can configure these to share a single license. Refer to "Distributed Licensing" on page 109.

Information Stored in the License

The following features are controlled by the license:

- Number of streams
- Splitting—whether the RealServer can act as a transmitter or receiver
- Multicasting
- Authentication
- ISP hosting
- Ad streaming
- Data types

Reading the License

You can read the file with RealSystem Administrator by clicking **About** in the left-hand frame. A second browser window appears, displaying the values for your license file. If you have multiple license files, RealServer will show the values for all of them at once.

You can also read the file with any text editor. Although you can read the file with a text editor, you cannot make changes. Any changes to the file invalidate it. If you have multiple files, you will need to read each file individually and calculate any additive features (such as number of streams) yourself.

The `LicenseDirectory` variable in the configuration file tells RealServer where to look for license information.

Additional Information

To learn about the configuration file, see "Configuration File" on page 94.

If the license file is invalid, RealServer will report an error message, add the error to the error log file, and will not start. To resolve this, remove the license file, and restart RealServer. It will use minimum settings, as described in the “Minimum Settings” table. Contact RealNetworks for a correct license file.

Minimum Licensed Features

If your RealServer suddenly allows fewer connections or otherwise appears to be using minimum settings, either your license has expired or RealServer is unable to start using the settings you’ve selected. The table below lists the minimum settings present in every RealServer.

Minimum Settings	
Feature	Value
Number of streams	25
RealPlayer versions	Only RealPlayer versions 5 and later are allowed to connect.
Splitting	Acts as pull splitting transmitter. Cannot act as pull splitting receiver, push transmitter, or push receiver.
Multicasting	Disabled
Authentication	Encoder and RealSystem Administrator users can be authenticated. Links to content cannot be authenticated.
Data types	RealVideo and RealAudio are enabled; all other types (Real G2 with Flash, WAV, AVI, VIVO) are disabled.

Note

Evaluation versions may have lower minimum values and additional features.

Chapter 4

SOURCES OF CONTENT

This chapter gives quick instructions on how to create an on-demand or a live file with RealProducer Plus. It also describes G2SLTA, the tool for broadcasting an on-demand file as if it were live.

RealServer can stream many other file types, which are not described in this chapter. Consult the RealNetworks Web site for information about additional file types.

Overview

A content creator can make a file and then place it in a location available to your RealServer, or she can encode it (using encoding software such as RealProducer Plus) and send it to your RealServer as it happens.

Once you have created or started encoding your source, you'll create a link for it, so that users can receive your content. The link will go in a Web page or a Ram file.

RealServer can serve the following file formats (check your license to see which of these your RealServer can serve):

- **Audio File Types**—RealAudio, WAV, AU, MPEG-1, MPEG-2, MP3
- **Video File Types**—RealVideo, AVI, QuickTime
- **Other File Types**—RealPix, RealText, GIF, JPEG, SMIL, Real G2 with Flash

In addition to serving your own on-demand and live files, you can also serve content distributed by another RealServer. This is called splitting. It is described in Chapter 12, “Splitting Live Presentations”.

Synchronized Multimedia Integration Language files, or SMIL files, are files that coordinate the delivery of several clips. A SMIL file (pronounced “smile”) tells the client what clips to play, in what order, and where to show them on the screen. SMIL files can perform basic or sophisticated timing and layout. A

SMIL file can refer to both on-demand and live clips. For detailed information on creating SMIL files, see *RealSystem Production Guide*. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

This chapter describes two methods for creating source files and sending them to RealServer:

- RealProducer Plus
- G2SLTA

Creating an On-Demand Source with RealProducer Plus

Instructions in this section create a brief, one-minute demonstration audio clip, using a music CD, and RealProducer Plus 8. Other versions of RealProducer Plus may have slightly different steps than the ones shown below; if you have a different software version, use these steps as a general guide.

Additional Information

To learn more about options for encoding, refer to *RealProducer Plus User's Guide*, available at <http://service.real.com/help/library/index.html>.

Step 1: Creating the Clip

1. Put a music CD in the computer's CD player and start playing it, using your system CD player. (Do not use RealJukebox, as it will not initialize the audio device needed for encoding.)
2. Start RealProducer Plus.
 - In RealProducer Plus for Windows or Macintosh, the **New Session-Choose Recording Wizard** dialog box appears. Place a check mark in the **Don't Use Recording Wizards** box. Click **OK**, then click **Cancel**.
 - In RealProducer Plus for UNIX, the main RealProducer Plus program is visible.
3. Choose **File>New Session**. A dialog box appears.
4. In the **Input Source** section, select **Media Device**.
5. Place a check mark in **Capture Audio** and uncheck **Capture Video**.
6. In the Output area, select **RealMedia File**.

7. In the box below it, type the file name `ondemand.rm`. (Always use the `.rm` extension.)

If you are encoding on the same machine as RealServer, you can type the complete path in RealProducer Plus.

Note

File names must consist of one word, with no spaces.

8. Click **OK**.

The New Session dialog box closes, returning to the RealProducer Plus main window.

9. Verify that **Multi-rate SureStream** is selected.

10. In the **Target Audience** area, make sure the boxes **28K Modem** and **56K Modem** are selected.

11. Leave all other fields blank.

12. Choose **Controls>Start**.

A message appears, asking if you want to add clip information.

13. Click **No**.

RealProducer Plus begins recording your music CD. The word “Encoding” appears in the lower left corner.

14. Wait one minute, then click **Stop**.

A message appears, asking if you want to stop encoding.

15. Click **Yes**.

16. Click **Close**.

RealProducer Plus halts the recording and creates a file named `ondemand.rm` in the RealProducer Plus directory, or in the directory location you specified in Step 7.

Step 2: Copying the Clip to RealServer

Copy the file `ondemand.rm` clip you created in the previous section, that is currently located in the main RealProducer directory, to the RealServer Content directory.

In Windows 95, Windows 98, and Windows NT, the path is `C:\Program Files\Real\RealServer\Content`.

In UNIX, the path is `/usr/local/RealServer/Content`.

Step 3: Linking to the On-Demand Clip

Create a link for the clip in a Web page. (The Web page can be local; it does not have to be on a remote Web server.)

In a Web page, type the following link and save the page (substitute your RealServer's machine name or IP address for *address*):

```
<a href="http://address:8080/ramgen/ondemand.rm">Click here to listen to my CD</a>
```

You can now view this Web page in a browser. When you click the link, RealPlayer will start and will play your `ondemand.rm` file.

You can also play the clip by starting RealPlayer, clicking **File>Open Location**, and typing the following in the dialog box that appears:

```
rtsp:address:554/ondemand.rm
```

Information on how to create links to your content is described in Chapter 5, "Understanding Link Formats".

Note

Notice that you must use the same capitalization in the link as you did when you created the file name, as they are case-sensitive.

Creating a Live Source with RealProducer Plus

Instructions in this section create a demonstration audio clip, using a music CD, and RealProducer Plus 8. Other versions of RealProducer Plus may have slightly different steps than the ones shown below; if you have a different software version, use these steps as a general guide.

Additional Information

To learn more about options for encoding, refer to refer to *RealProducer Plus User's Guide*, available at <http://service.real.com/help/library/index.html>.

Other sources of live content are described elsewhere in this chapter; see "Creating a Live Source with G2SLTA".

There are two steps to setting up and running RealProducer Plus:

1. Starting the live encode.
2. Creating a link to the live event.

Step 1: Starting the Live Encoding with RealProducer Plus

1. Put a music CD in the computer's CD player and start playing it, using your system CD player. (Do not use RealJukebox, as it will not initialize the audio device needed for encoding.)
2. Start RealProducer Plus.
 - In RealProducer Plus for Windows or Macintosh, the **New Session-Choose Recording Wizard** dialog box appears. Place a check mark in the **Don't Use Recording Wizards** box. Click **OK**, then click **Cancel**.
 - In RealProducer Plus for UNIX, the main RealProducer Plus program is visible.
3. Click **File>New Session**. A new dialog box appears.
4. In the **Input Source** section, select **Media Device**.
5. Place a check mark in **Capture Audio** and uncheck **Capture Video**.
6. In the **Output** area, select **Live Broadcast**.
 - a. In the **RealServer** box, type the IP address of the machine on which your RealServer is installed.
You can type the name (such as realserver.example.com) of your RealServer instead.
 - b. In the **Server Port** box, leave the default setting of 4040.
 - c. In the **Filename** box, type live.rm. (Always use the .rm extension.)
If this broadcast is to be authenticated, use the path /secure/live.rm.

Note

File names must consist of one word, with no spaces.

- d. In the **Username** box, type the same user name you use for logging in to RealSystem Administrator.
(To create an additional user name and password for each person who will be encoding to your RealServer, see "Authenticating Encoder Users" on page 227.)

- e. In the **Password** box, type the password you use for RealSystem Administrator.
7. Click **OK**.
The New Session dialog box closes, returning to the RealProducer Plus main window.
8. Verify that **Multi-rate SureStream** is selected.
9. In the **Target Audience** area, make sure the boxes **28K Modem** and **56K Modem** are selected.
10. Leave all other fields blank.
11. Click **Start**.
A message appears, asking if you want to add clip information.
12. Click **No**.
RealProducer Plus begins encoding your music CD.

Step 2: Linking to the Live Event

These instructions describe a unicasting link in a Web page. For more sophisticated delivery methods, consult Chapter 12, “Splitting Live Presentations” and Chapter 13, “Multicasting Live Presentations”.

Create a link for the live broadcast in a Web page. (The Web page can be local; it does not have to be on a remote Web server.)

In an existing Web page, type the following link and save the page (substitute your RealServer name or IP address for *address*):

```
<a href="http://address:8080/ramgen/encoder/live.rm">Click here to listen to my CD</a>
```

Tip

Be sure to use the same file name extension in the link as you typed in the encoder. RealServer will not supply a missing or incorrect extension.

The word `/encoder/` alerts RealServer that this is a live broadcast. Everything after `/encoder/` is a file name or a path and file name. The path can be an actual path that matches directories in RealServer, or it can be a virtual path that you use to distinguish this broadcast from others. Virtual paths are described in greater detail later in this chapter.

You can also play the clip by starting RealPlayer, clicking **File>Open Location**, and typing the following in the dialog box that appears:

```
rtsp:address:554/encoder/live.rm
```

Information on how to create links to your content is described in Chapter 5, “Understanding Link Formats”.

Note

Be sure to use the same capitalization in the link as in the file name, as file names are case-sensitive.

Virtual Paths

In some cases, in the encoding software, you may want to supply a virtual path that doesn’t actually exist. The notion of a virtual path is only applied to live content. In RealProducer Plus, the path name is typed in the **Filename** box.

Virtual paths can be useful in segmenting your streams. For example, if the accounting department and the marketing department regularly encode announcements made by their department heads, you can tell each department to use its department name so that you don’t have to worry about each department using the same name for its broadcast. The accounting department can encode to /accounting/update.rm, and the marketing department can encode a clip named /marketing/update.rm. Both streams will be named update.rm, but because of the virtual path name, each department’s stream remains distinct, and viewers in the different departments will see the right clip as they click the appropriate link.

The Connection Between Encoder Paths and Links

The filename and path you type in the encoding software will always be reflected in the link to the resulting content. For example, if you type videos/familyreunion/1999/reunion.rm in the encoder, the link in the Web page to the clip will look like:

```
http://realserver.example.com:8080/ramgen/encoder/videos/familyreunion/1999/reunion.rm
```

If you are using the directory structure created by RealServer at installation, you don’t have any directories named encoder or videos or familyreunion or 1999. But because /encoder/ is the mount point, and because you typed the rest of the path in the encoding software and matched them in the link, RealServer is able to find the clip.

Creating a Live Source with G2SLTA

The **G2SLTA** (Simulated Live Transfer Agent) software tool converts an on-demand stream to a live event. It simulates the encoder's connection to RealServer. Just as in an actual live broadcast, viewers who watch a presentation join the event in progress; no matter when visitors connect, they all see the same thing at the same time.

This feature also allows you to create a playlist that cycles through a set of prerecorded clips in a certain order.

You can stream RealAudio, RealVideo, AU, and WAV clips using **G2SLTA**. Only audio and video files can be delivered as live content with **G2SLTA**. Data types such as RealText and RealPix cannot be used; they have their own live delivery utilities. For more information, see *RealPix Authoring Guide* and *RealText Authoring Guide* at <http://service/help/library/index.html>.

You start **G2SLTA** from a command line. Just as if you were creating a live encoding session, you assign a name to the "live" broadcast, and supply a user name and password.

G2SLTA Command Line

```
C:\Real\RealServer>Bin\g2slta.bat realserver.example.com 4040 pbrown  
swordfish annual.rm Content\Annual_Report.txt
```

As **G2SLTA** runs, it displays the name of the file it is broadcasting. A line of asterisks indicates how much of the file has been broadcast, in percent. (A row of asterisks that lines up below the number five indicates that the file is approximately 50 percent complete.)

G2SLTA Showing Progress Through Playlist

```
5 files.
Encoding CompanyLogo.rm...
0---1---2---3---4---5---6---7---8---9---10
*****
Encoding Welcome.rm...
0---1---2---3---4---5---6---7---8---9---10
*****
Encoding President.rm...
0---1---2---3---4---5---6---7---8---9---10
*****
Encoding Treasurer.rm...
0---1---2---3---4---5---6---7---8---9---10
*****
Encoding Conclusions.rm...
0---1---2---3---4---5---6---7---8---9---10
*****
Done.
```

After playing the files according to the `-r` and `-n` switches in the command line (if any), **G2SLTA** displays the word **Done**. If you use the `-n` switch without a number (to create an infinite loop), **G2SLTA** will not ever show **Done** because it will always be playing a file.

When to Use G2SLTA

The following are examples of when to use **G2SLTA** instead of live broadcasting:

- Rebroadcasting a live event for a later time zone
- Creating a looped presentation of one or more files
- Simulating a radio station by creating a playlist with a long list of files
- Testing your system in anticipation of an actual live broadcast
- Periodically re-broadcasting popular content, such as a concert during a telethon, or news headlines throughout the day

G2SLTA Used with Other Features

G2SLTA works with all other RealServer live broadcasting features.

On-Demand Streaming and G2SLTA

On-demand clips are “converted” to live clips as **G2SLTA** sends them to RealServer.

Live Unicasting and G2SLTA

Use **G2SLTA** to test your equipment for doing live broadcasting before the event starts, and work out any bugs. Make sure you have the correct links.

Archiving G2SLTA Broadcasts

The live archiving feature can create static files of all live files that arrive from an encoder. RealServer will use the same archive settings as for other broadcasts, whether it is configured to create one large file from each broadcast or several smaller files. If the live archiving feature is configured to save one large file, it will use the name you specified in *livefile* in the command line (see “G2SLTA Syntax” on page 54). For small files, it will use the name specified by *livefile*, and appending numbers at the end (see “Small Files” on page 151).

Note

If you start **G2SLTA** with the infinite loop instruction (omit the `-n` switch from the command line), and the live archive feature is set up to create a single large file from the broadcast, RealServer will not create an archive file until you end the broadcast. Then it will create one large file.

If the live archiving feature is turned on for all arriving streams (the `*` in the **Virtual Directories** list), all broadcasts from **G2SLTA** will automatically be archived.

Splitting and G2SLTA

You can use simulated broadcasts as a live source for splitting. See “Creating a Source for Splitting” on page 60.

Multicasting and G2SLTA

You can use simulated broadcasts as a live source for multicasting. See “Creating a Source for Multicasting” on page 60.

Access Control, Authentication, and G2SLTA

In order for a broadcast to be authenticated, it must use the `/secure/` mount point. The value you use for *livefile* must include the `/secure/` mount point.

A client that connects to any live broadcast that uses **G2SLTA** as its live source will be authenticated in the usual manner.

Java Monitor and G2SLTA

As with all live events, you can monitor the number of clients connected to a live broadcast by using the Java Monitor.

The broadcast created by **G2SLTA** appears in the Java Monitor as a typical encoder connection.

Reporting and G2SLTA

Just as in any other live broadcast, a record is created in the access log for any client that connects to a live broadcast simulated by **G2SLTA**.

Setting Up and Running G2SLTA

There are four steps to setting up and running **G2SLTA**:

1. Configuring RealServer.
2. Creating a playlist.
3. Running **G2SLTA**.
4. Creating the link to the simulated live broadcast.

Use the following instructions to set up and run **G2SLTA**.

Step 1: Configuring RealServer

In these steps you will:

- set up RealServer as for encoding
- confirm the encoder password information
- get other broadcasting information.

Configuring RealServer to Work with **G2SLTA**:

G2SLTA uses the same configuration settings as live unicasting. See “Encoders” on page 146.

Looking Up Password Information

You will need to supply a user name and password in the **G2SLTA** command line. By supplying a user name, you allow RealServer to authenticate you and verify that the simulated stream is authorized.

Generally, you use the user name and password that you supplied during RealServer setup. This information was automatically added to the list of authorized encoder users.

► To look up password information:

1. In RealSystem Administrator, click **Security**. Click **Authentication**.
2. In the **Realms** list, select SecureEncoder.
3. Click **Browse Users in Realm**. A new browser window appears.
 - a. Examine the names in the list. There may be only one (the user name you use for connecting to RealSystem Administrator, which you created during Setup).
 - b. Click **Close**. You are returned to RealSystem Administrator.
4. If you did not see a user name that you want to use when running **G2SLTA**, click **Add a User to Realm**. Refer to instructions in Chapter 15, “Authenticating RealServer Users” for information on how to customize the new user name and password.
5. Make a note of the user name and password you want to use. This information will go on the command line, as described in “G2SLTA Syntax” on page 54.

Looking Up Broadcasting Information

To run **G2SLTA**, you need to know the correct port number to use.

► To look up encoding information:

1. In RealSystem Administrator, click **Broadcasting**. Click **Encoder**.
2. Make a note of the values for **Port** and for **Mount Point**.

Step 2: Creating a Playlist

The playlist is a text file that contains a list of the files that **G2SLTA** will stream.

- If you want to simulate a live broadcast of only one file, either create a playlist that refers to just that file, or substitute the file’s name for the *playlist* parameter in the command line.

- If you have a series of files you want to play during your simulated live broadcast, list them in sequence in the playlist. An optional command plays files in a playlist in random order.

You can include as many files as you like in the playlist.

All files in the playlist must have been encoded at the same bit rates. Any SureStream files in the playlist must contain the same quantity of bit rates, and the bit rates must be the same among all files.

Warning

Do not combine both SureStream files and non-SureStream files in the playlist.

When you start **G2SLTA**, you give a name to the stream that will be used as the file name that will be included in the URL. The playlist name is not included in the URL.

► To create a playlist:

In a text file, list each file that you want RealServer to play, one per line. Files are played in the order shown in the file.

Format of Playlist

```
first_file  
second_file
```

If the files are not in the same directory as the playlist, be sure to include their full paths, whether absolute or relative to the location of the playlist.

Example Playlist

For example, a file named `Annual_Report.txt` might contain the following items:

```
CompanyLogo.rm  
Welcome.rm  
President.rm  
Treasurer.rm  
Conclusions.rm
```

For more playlist features, see “Optional G2SLTA Features” later in this chapter.

Step 3: Running G2SLTA

The instructions below show how to run the file **G2SLTA.BAT** (Windows) or **g2slta.sh** (UNIX). These files set two environment variables: **G2SLTA_PLUGIN_PATH** and **G2SLTA_SUPPORT_PATH**, with the values of variables **PluginDirectory** and **SupportPluginDirectory**. They are customized with your directory values when you install RealServer. (To view the values of **PluginDirectory** and **SupportPluginDirectory**, search for them in the configuration file; they are not shown in RealSystem Administrator.)

Warning

Running **g2slta.exe**, rather than the batch file or shell script, will result in error messages. Only the batch file and shell script contain the custom settings for your computer.

► To run G2SLTA:

1. At a command line in the **Bin** directory, run **G2SLTA.BAT** (Windows) or **g2slta.sh** (UNIX), using the syntax shown below.

G2SLTA Syntax

The **G2SLTA** program uses the following format:

Windows

```
g2slta.bat host port username password livefile playlist [-r] [-nN] [-bN]
```

UNIX

```
g2slta.sh host port username password livefile playlist [-r] [-nN] [-bN]
```

where:

<i>host</i>	Name of the RealServer system and domain name, or IP address.
<i>port</i>	Port number specified in the Encoder list, usually 4040. See “Looking Up Broadcasting Information” on page 52.
<i>username</i>	Name of the encoder user as defined in the encoder realm. Often the same as the username for RealSystem Administrator. See “Looking Up Password Information” on page 52. If no username has been defined, type two quotation marks: "".
<i>password</i>	The corresponding user’s password. Often the same as the password for RealSystem Administrator. See “Looking Up Password Information” on page 52. If no password has been defined, type two quotation marks: "".

<i>livefile</i>	Name of the broadcast that you want to include in the URL that links to this event.
<i>playlist</i>	Name of your playlist. If it is in a different directory than the RealServer directory, include its path. If you are broadcasting a single file, you can give its full path and name here, instead of referencing a playlist.
<i>-r</i>	Optional. Indicates that RealServer should randomly play the files in the playlist. See “Playing Files in Random Order” on page 56 for further discussion.
<i>-nN</i>	Optional. Gives the number of files in the playlist for RealServer to play. When this switch is omitted, the list of files plays indefinitely. See “Specifying Number of Times to Play Files” on page 56 for more information.
<i>-bN</i>	Optional. Gives the target bandwidth to stream from a SureStream file; RealServer will stream the bandwidth of <i>N</i> . Give the number in bits per second. For example, <i>-b20000</i> will stream the 20 kilobit bit-rate stream. See “Controlling Bandwidth” on page 57 for additional information.

G2SLTA Example

The following example command starts a simulated live broadcast. The user name is `pbrown` and the password is `swordfish`. The filename that users will connect to is `annual.rm`. Files are specified by the playlist named `Annual_Report.txt`. Switches for random play and bandwidth are omitted; therefore the files will play in the order listed in the playlist, and all the bandwidths are available to clients. Since there are five files in the playlist, `-n5` is used to indicate that each file should be played once. In this example, the command is typed from the main directory, thus the commands are preceded with the path to the `Bin` directory.

Windows

```
Bin\g2slta.bat realserver.example.com 4040 pbrown swordfish annual.rm  
Content\Annual_Report.txt -n5
```

UNIX

```
Bin/g2slta.sh realserver.example.com 4040 pbrown swordfish annual.rm  
Content/Annual_Report.txt -n5
```

Optional G2SLTA Syntax

This section discusses the three optional command line options, which are also shown in “G2SLTA Syntax” on page 54:

- Playing files in random order

- Specifying number of times to play files
- Controlling bandwidth from all SureStream files

Playing Files in Random Order

The `-r` switch instructs **G2SLTA** to stream the files in the playlist in random order.

Tip

Use both the `-r` and `-nN` switches, where `N` is a multiple of the number of files in the playlist, to cycle randomly through the playlist `N` times.

Specifying Number of Times to Play Files

The `-nN` switch gives the total number of files to play from the playlist. Notice that it does not indicate the number of times each file should be played.

To play each file in the playlist once, count the number of files in the playlist, and use that value for `N`.

To indicate that RealServer should play seven files, include `-n7` in the command line. If a playlist contains three files, RealServer will play the file sequence twice, and will play the first item a third time, for a total of seven files played.

Using the example playlist that contains five files (shown in “Example Playlist” on page 53), the switch `-n7` would play the following files:

CompanyLogo.rm (first time)
Welcome.rm (first time)
President.rm (first time)
Treasurer.rm (first time)
Conclusions.rm (first time)
CompanyLogo.rm (second time)
Welcome.rm (second time)

To play through each file in the playlist `x` times, multiply `x` by the number of files in your playlist and use it for `N`.

Another way to think of this switch:

- If the value for `N` is the same as the number of the files in the playlist, each file in the playlist will be played once.
- If the value for `N` is less than the number of files in the playlist, only the first `N` files in the playlist will be streamed.

- If the value for N is greater than the number of files in the playlist, every item in the playlist will be played at least once.

To cycle through the playlist indefinitely, omit the -n switch.

Controlling Bandwidth

Use the -b switch bandwidth switch when your playlist consists of SureStream files, and you want only a specific bandwidth to be broadcast. Ordinarily, clients will choose the best possible SureStream rates for their connections.

Step 4: Linking to the Simulated Live Broadcast

Links to simulated events use the same format as for actual live events, including the mount point that corresponds to the port number you specified in the command line.

Additional Information

See “Step 2: Linking to the Live Event” on page 46.

For example, if you started the simulated live event using the example shown in “G2SLTA Example”, the link in the Web page would look like the following. Notice that the Ramgen mount point is included.

```
http://realserver.example.com:8080/ramgen/encoder/annual.rm
```

If you want to test this broadcast before creating the Web page, type the following directly in the RealPlayer **Open Location** dialog box. Notice that the Encoder mount point is included.

```
rtsp://realserver.example.com:554/encoder/annual.rm
```

Tip

Because the URL is linked to the list of streamed files via the *livefile* name, you can use different playlists with different names, yet keep the same link on the Web page.

Stopping G2SLTA

The **G2SLTA** program will stop automatically when it has finished playing all the files in the playlist, according to the command line instructions.

To stop **G2SLTA** before it is completed, either press **CTRL+C** at the command line from which you started **G2SLTA** (Windows), or use the **KILL** command with the process ID of the **G2SLTA** process (UNIX).

Ordinarily, you will not need to do this unless you want to stop the broadcast prematurely, or if you want to stop an infinite looped broadcast.

Optional G2SLTA Features

The **G2SLTA** program has additional options that you can configure:

- Customizing title, author, and copyright information displayed by playlists
- Changing playlists while **G2SLTA** is running

Customizing Title, Author, and Copyright Information

When a clip is initially encoded, the content creator can fill out information about the title, author, and copyright (TAC) information. You can view this information for any clip (if the content creator supplied it) by choosing **Help>About this Presentation** in RealPlayer. Other client software may have a different method of showing the TAC information.

Ordinarily, **G2SLTA** sends the TAC information for each clip as it is broadcast. If you check the About this Presentation information as each clip is played, you will see the information changing.

By using options within the playlist, you can override the encoded TAC information and supply your own:

- A single set of TAC information can apply to all files in the playlist.
 - Each clip in the playlist can display different TAC information.
- To include title, author, and copyright information for the entire playlist

Type the following at the beginning of the file. The rest of the file lists the files to be played:

```
Title: your title
Author: your author
Copyright: your copyright information
first_file
second_file
```

All files in the playlist will stream with the same TAC information.

In the following example, the information title, author, and copyright information appears the same for every clip in the presentation:

Title: Example.com's Annual Report
Author: Chris Lee, Executive Assistant
Copyright: Copyright 1999, Example.com
CompanyLogo.rm
Welcome.rm
President.rm
Treasurer.rm
Conclusions.rm

- To include separate title, author, and copyright information for individual clips:

Add the TAC information to the end of each line, using the following format:

```
first_file?title="title_info"&author="author_info"&copyright="copyright_info"
```

where *first_file* is the name of the file; *title_info*, *author_info*, and *copyright_info* are strings of any length.

If you have included an overall TAC at the beginning of the playlist, including information about separate files will “turn off” the TAC at the beginning of the file; subsequent clips will then stream with their own TAC information.

In the following example, separate TAC values are supplied for each clip:

```
CompanyLogo.rm&title="Our Founder"&author="P. Brown, artist"&copyright="1999"  
Welcome.rm&title="Welcome to the Annual Meeting"  
President.rm&title="Lee Adams, President"  
Treasurer.rm&title="Chris Anderson, Treasurer"  
Conclusions.rm&copyright="Example.com, 1999"
```

Changing Playlists while G2SLTA is Running

If you are using the `-n` switch, either to loop the playlist infinitely or to play all clips a specified number of times, you can take advantage of the fact that RealServer re-reads the playlist after it plays all the clips in the playlist.

Note

If you used TAC information to provide an overall set of information for all the clips in the presentation (by listing that information at the beginning of the playlist), that information will remain the same, even though the list of files to be played is different.

- To change the playlist while G2SLTA is Running:

1. Start **G2SLTA**.
2. Make changes to the playlist, and save them. Or, create a new playlist and save it with the name of the existing playlist.

3. RealServer will use the modified playlist as soon as it plays all the clips in the current playlist.

Using G2SLTA for Splitting and Multicasting

Creating a Source for Splitting

Instructions in this section give high-level overview steps; they assume you already have the splitting feature enabled. For instructions on individual steps, consult the appropriate section of this manual.

- ▶ To use G2SLTA as the live event for push splitting:
 1. Run **G2SLTA**, using the instructions in “Step 3: Running G2SLTA”. Make note of the value you used for *livefile*; you will use it in Step 2 and in Step 3.
 2. If the path you typed for *livefile* does not already exist in the **Directory Sources** section, add it now.
 3. In the Web page that points to this split live stream, create a link that points to *livefile*. Use the format described in “Step 3: Linking to Split Content” on page 169.
- ▶ To use G2SLTA as the live event for pull splitting:
 1. Run **G2SLTA**, using the instructions in “Step 3: Running G2SLTA”. Make note of the value you used for *livefile*; you will use it in the next step.
 2. In the Web page that points to this split live stream, create a link that points to *livefile*. Use the format described in “Linking to Pull Split Content” on page 178.

Creating a Source for Multicasting

Instructions in this section give high-level overview steps; they assume you already have the multicast feature enabled. For instructions on individual steps, consult the appropriate section of this manual.

- ▶ To use G2SLTA as the live event for back-channel multicasting:
 1. Run **G2SLTA**, using the instructions in “Step 3: Running G2SLTA”. Make note of the value you used for *livefile*; you will use it in the next step.
 2. In the Web page that points to this live multicast, create a link that points to *livefile*. Use the format described in “Linking to Back-Channel Multicasts” on page 198.

- To use G2SLTA as the live event for scalable multicasting:
 1. Run **G2SLTA**, using the instructions in “Step 3: Running G2SLTA”. Make note of the value you used for *livefile*; you will use it in the next step.
 2. On the scalable multicasting page, add a channel for *livefile*.
 3. In the Web page that points to this live multicast, create a link that points to *livefile*. Use the format described in “Linking to Scalable Multicasts” on page 203.

Files Required by G2SLTA

The files used by the **G2SLTA** program are shown below.

Files Required by G2SLTA

Streamed File Type	Windows File Name	UNIX File Name	Configuration File Variable Showing Location of File
All	encn3260.dll	encn.so.6.0	SupportPluginDirectory
	slta3260.dll	sltalib.so.6.0	SupportPluginDirectory
	enco3260.dll	encoplin.so.6.0	PluginDirectory
RealAudio and RealVideo	rmff3260.dll	rmffplin.so.6.0	PluginDirectory

Other data types may require other Windows and UNIX files.

Working with Redundant Sources

For important events in which multiple backup sources are required, RealServer includes a feature to use multiple sources for the same event. Should one source become unavailable, RealServer will automatically switch the stream to use the next source.

If one source fails, the following message appears in the error log: “Switched live source from *address1* to *address2*.”

Redundant Sources Used with Other Features

Redundant sources work with all other RealServer live broadcasting features.

On-Demand Streaming and Redundant Sources

Only live events use this feature, but you can use **G2SLTA** to simulate a live broadcast with an on-demand file. See “Setting Up the Source” on page 64.

G2SLTA and Redundant Sources

G2SLTA can be used to “convert” an on-demand file to a live file; see “Setting Up the Source” on page 64.

Archiving Redundant Broadcasts

RealServer archives the incoming source, but doesn’t store the source number. In the example shown in “Setting Up the Source”, the various sources are named `live.rm.1` and `live.rm.2`; RealServer archives a file named simply `live.rm`, without the delimiter and number.

Splitting and Redundant Sources

A transmitter that uses the redundant sources feature sends out its streams just like any other live broadcast.

To create a system with multiple layers of backups, you should configure multiple transmitters to name their broadcasts with the same file name, plus the delimiter and a unique number.

Multicasting and Redundant Sources

Redundant live sources can be multicast, as with any other live content.

Access Control, Authentication, and Redundant Sources

Use access control and authentication the same as with any other live content.

ISP Hosting and Redundant Sources

ISP hosting does not work with live broadcasts.

Java Monitor and Redundant Sources

Redundant sources are displayed like any other live broadcasts.

Reporting and Redundant Sources

Just as in any other live broadcast, a record is created in the access log for any client that connects to a live broadcast originating from redundant sources. You can’t determine which redundant source is in use, only that the

redundancy feature is in use. See “Using the GET Statement to Identify Delivery Method” on page 300 to see how redundant sources appear in the access log.

Setting Up Redundant Sources

To use this feature, you perform two steps:

1. Add a unique number to the file name in RealProducer or in **G2SLTA**. The number is separated from the file name by the delimiter character shown in the **Delimiter** field (described later in this section).
2. Add the broadcast redundancy mount point to the link.

To minimize latency when RealServer changes stream sources, start all sources as close to the same time as possible.

This feature is automatically enabled; you don't need to make any changes unless you want to disable the feature, or to change the settings it uses. To view the settings currently in use for broadcast redundancy, start RealSystem Administrator and click **Broadcasting>Redundancy**:

- **Enabled**—turns on this feature.
- **Delimiter**—the character you want to use in the source application (such as RealProducer) to indicate the source number. This is normally a period (.); other options are a single quote ('), a tilde (~), or a caret (^).

Additional Information

To see how to use the delimiter in the source application, see “Setting Up the Source” on page 64.

- **Timeout**—the number of seconds RealServer should wait for an inactive source before switching to another one. The recommended value is 2. This can range from 0 to 30.
- **Reconnect**—indicates how users will get the new stream should the original become unavailable:
 - **Automatic reconnection**—to cause client software to automatically switch to the new stream, select Automatic (this is the default selection).
 - **Manual reconnection**—to display the reconnect error message (shown in the **Error Message** field, described below), and require users to click

a button in their client software to switch to the new stream, select Manual.

- **Mount Point**—the mount point to use in the links to the redundant broadcasts. Typically, this value is /redundant/. If this field is blank, a default error message will appear.
- **Error Message**—text of the message that will appear to users, if **Reconnect** is set to Manual. This message tells users how to connect to the new stream. The supplied message is “Broadcast timed out; click Play button to restart.”

Note

For streams delivered with the PNM protocol, users must always reconnect manually, regardless of the setting in the **Reconnect** list.

Setting Up the Source

Regardless of the source for the live broadcast, you must add a delimiter and a unique number to the file name in each source. Note that the number doesn't indicate the order in which the streamed packets arrive at RealServer; it is used as a unique identifier.

- To set up the source for RealProducer:

In one session of RealProducer, type the name of the file, and type a delimiter and number after it. For example, if the file name is live.rm, and the delimiter is a period (.), you would type live.rm.1 in the **Filename** box.

In the next session of RealProducer, type the name of the file as before, but use a different number: live.rm.2.

Continue this numbering for each additional source.

- To set up the source for G2SLTA:

Add a period and a number to the *livefile* parameter. For example,

```
Bin/g2slta.sh realserver.example.com 4040 pbrown swordfish annual.rm.1
Content/AnnualReport.txt
```

Use another number for the second session of **G2SLTA**:

```
Bin/g2slta.sh realserver.example.com 4040 pbrown swordfish annual.rm.2
Content/AnnualReport.txt
```

Continue this numbering for each additional source.

Linking to Redundant Content

Links to live events with multiple sources use the same format as links to single-source live events, but substitute the broadcast redundancy mount point for the encoder mount point.

- To link to redundant content from a Web page:

The link to a live event with multiple sources has the following format:

`http://address:HTTPPort/ramgen/redundant/path/file`

RealServer Live Content URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming. Always use <code>http</code> in Web pages.
<code>address</code>	Machine and domain name of RealServer. IP address may be substituted.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<code>ramgen</code>	Ram file generator mount point.
<code>redundant</code>	Mount point for live events with redundant sources.
<code>path</code>	Optional; any actual directory, relative to the base path of the main mount point. If you typed any subdirectory in the source software, include it here.
<code>file</code>	The file name itself, including the extension.

Note

The number and delimiter you used on the source do not appear in the link itself.

For example, the link to a live event being encoded as `concert.rm.1` and `concert.rm.2` would look like this:

`http://realserver.example.com:8080/ramgen/redundant/concert.rm`

Links typed directly in RealPlayer or used in a Ram or SMIL file use the following format:

`rtsp://address:RTSPPort/redundant/path/file`

The format is nearly the same as the link used in the Web page; the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

Chapter 5

UNDERSTANDING LINK FORMATS

Links to RealServer content use special formats that activate RealServer and tell it how to deliver the requested material. This chapter describes the theory behind the different formats that RealServer uses.

Overview

This chapter explains how to construct the links to your content. Different methods use different formats, but they're all based on the same kind of structure.

This chapter covers generic link formats, as well as link formats that apply to all types of features (such as subdirectories in a link). For instructions on linking to content served with a particular delivery method, refer to that method's chapter. Also, Chapter A, "Summary of Link Formats" gives a quick review of all the various formats.

Tip

For examples of the different types of links, as well as the features of SMIL files, view the demonstrations by clicking **Samples** in the left-hand frame of RealSystem Administrator, and then clicking one of the SMIL demonstration links.

When to Skip This Chapter

This chapter provides an in-depth discussion of link anatomy, including special directory structures. The background information provided may not be of interest to you if you aren't using any of RealServer's advanced functions.

- If you are simply streaming on-demand content, and are storing it in the Content subdirectory, you can use the link format described in "Linking to On-Demand Clips" on page 141.

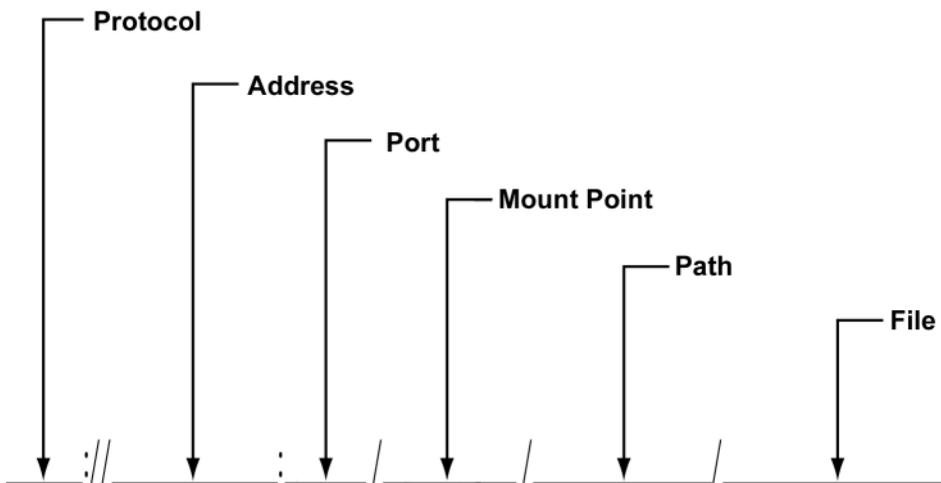
- If you are broadcasting live content, use the link format described in “Creating the Link to the Live Unicast” on page 148.

Parts of a Link

A typical link to media clips served by RealServer includes elements such as a port, a mount point, a path, and a file name.

The following illustration shows the parts of a more complex link.

Parts of a Link



All links to material served by RealServer use the same general format:

protocol://address:port/MountPoint/path/file

RealServer URL Components

Component	Meaning
<i>protocol</i>	The protocol used to initiate streaming. Either rtsp, pnm, or http.
<i>address</i>	Address of RealServer; IP address or machine and domain name.
<i>port</i>	Port number where RealServer listens for requests sent via the protocol listed at the beginning of the URL.
<i>MountPoint</i>	The mount point tells RealServer how the clip should be served. For on-demand content, usually consists of the main mount point (a single forward slash).

(Table Page 1 of 2)

RealServer URL Components (continued)

Component	Meaning
<i>path</i>	Optional; it is the subdirectory tree, relative to the base path of the mount point, where the content is located. If the file is located in the base path itself, omit <i>path</i> .
<i>file</i>	The name of the presentation, including the extension.

(Table Page 2 of 2)

Protocol

The protocol is the communication protocol that RealServer uses in sending the media clip.

RealServer uses two main protocols to communicate with clients:

- RTSP (Real Time Streaming Protocol) for clips created and read with RealSystem 6 tools
- PNA (Progressive Networks Audio) for clips created and read with earlier versions of RealSystem tools

RTSP is a client-server protocol designed specifically for serving multimedia presentations. It is an open standard, and very useful for large-scale broadcasting. Only RTSP can deliver SureStream™ files with their multiple bandwidth encoding. SMIL, RealText, and RealPix also require RTSP.

PNA is the proprietary client-server protocol designed and used by RealNetworks in RealSystem versions 5 and earlier. The ability to serve via PNA is supported in RealServer 8 for compatibility with older versions of RealPlayer.

RealServer also uses HTTP to stream HTML-based material, such as Ram files and RealSystem Administrator pages.

Additional Information

Read “Protocols Used by RealServer” on page 120.

Choosing the Right Protocol

Links to media files streamed by RealServer can appear in four places, and use different protocols, as shown in the following table. The protocol you use depends on where you are placing the link, and what type of content it points

to. Notice that Web pages require a slightly different link format than the other three venues.

Links in RealSystem Files

A link in this location...	...that points to this type of file...	...uses this protocol
Web page	Individual clip, SMIL file, Ram file, or Ramgen	http
SMIL files	Individual file or files	rtsp
Ram files	Individual file or files	rtsp or pnm
The Open Location dialog box of RealPlayer	Individual file	rtsp or pnm

Address

The address is the IP address or the machine and fully qualified domain name where your RealServer is installed. You can use either. In this book, the example address is always `realserver.example.com`, rather than the equivalent IP address.

Port

The port number is the number where RealServer is listening for the appropriate RTSP, PNA, or HTTP request.

Including the port number of the RealServer machine is optional when you use RealServer's default port settings. If you don't include a port number in the URL, the client (such as RealPlayer) will supply one on its own. It looks at the protocol, shown at the beginning of the URL, to decide which port number to use.

To check the port numbers in use on your RealServer, look in RealSystem Administrator. Click **General Setup>Ports**.

Default Port Numbers

Protocol	Port Number
http	8080
rtsp	554
pnm	7070

Reasons for Changing Port Values

You might want to change the port numbers, using RealSystem Administrator, if multiple RealServers are using the same IP address, or if you want to segregate requests for different material.

If your RealServer and Web server are on the same machine, you may need to modify the HTTP Port setting. See “Running Web Servers and RealServer on the Same System” on page 112 for information.

Note

If you change port values, you must include the new port number in the link. If RealPlayer attempts to play a clip for which the port information is incorrect, it may try to request the information via HTTP, which is a much less efficient delivery method.

Mount Point

A mount point reference appears in every URL. It is a shortcut name that tells RealServer which feature (or file system plug-in) will be handling the request. Most of the delivery methods each have their own mount point.

Mount points are listed in RealSystem Administrator.

In the case of on-demand content, though, the mount point is usually defined as a single forward slash, and is therefore “invisible” in the mount point.

Some frequently used mount points are:

- **/ (the single forward slash)**—for on-demand content located in the Content directory
- **/encoder/**—for live content digitized by encoders
- **/ramgen/**—for generating a Ram file

To determine which mount point to use (if any), you must first decide which type of delivery method you are using. To find out the correct mount point to use, consult the table below. The table is based on the default configuration your RealServer was shipped with; if you have changed these values, you will need to use the new settings.

In addition, if the link will be used in a Web page, remember to also include the Ramgen mount point. (See “Ram Files and Ramgen” on page 74 for more information.)

Typical Mount Points for Various Delivery Methods

Method of delivery	Mount point	To look up the mount point in RealSystem Administrator, click Configure, then click...
On-demand	/ (a single forward slash)	General Setup>Mount Points
Live (created by encoder and G2SLTA)	/encoder/	Broadcasting>Encoder or Broadcasting>Pre-G2 Encoder
Splitting—Push	/broadcast/	Splitting>Transmitter
Multicasting—Back-Channel	/encoder/ (same as live)	Broadcasting>Encoder or Broadcasting>Pre-G2 Encoder
Multicasting—Scalable	/scalable/	Multicasting>Scalable
Authenticated	/secure/	Security>Commerce>Protected Path

Including Multiple Mount Points in One Link

In some cases, a link will include more than one mount point. The Ramgen mount point is often used in addition to other mount points. The scalable multicast mount point is used at the same time as the live broadcasting mount point.

For multiple mount points not covered within these chapters, consult “Multiple Mount Points in Links” in Chapter A, “Summary of Link Formats”.

Using different mount points that point to the same base path or using the same file system can be an effective way of providing conceptual organization of content. For example, if content on your RealServer is being supplied by different people, you may elect to establish a different mount point for each person’s material, even though the material is stored on the same machine, though in separate locations.

Mount Points and Directories with the Same Name

RealServer looks through the list of mount points before it looks for virtual or actual directory names. Should a mount point or virtual directory have the same name as an actual directory, RealServer will ignore the actual directory.

There is one case in which you can use this to your advantage: displaying a message that says “Currently experiencing technical difficulties” when a live broadcast is interrupted. Live files are sent to a mount point that does not have a corresponding base path. Live files are streamed as they are created by an encoder, and they never exist in file form. Create an actual directory with the same name as the live mount point, and place a small file containing your message in this subdirectory. If a live stream fails to arrive at RealServer, RealServer will search for an actual directory that matches the URL. In this case, it will find the subdirectory with the error file in it.

Additional Information

See “Playing A “Please Stand By...” Message” on page 149.

Path

The path value references the subdirectory (if any) where the clip is located.

- **On-demand content**—include the path if the clip is located in a subdirectory of Content. Include only the subdirectory names under Content; you don’t include Content in the subdirectory.

If you have created an additional mount point for on-demand content, you can determine whether a path is necessary by looking at the new mount point and its base path. (The base path gives the actual location for files served by the mount point.) If the on-demand clips are located in a subdirectory of the base path, you need to include the subdirectory (or subdirectories) in the link.

- **Live content**—whether you include a path in the link depends on what the content creator typed when setting up the source for your live event. If the event is encoded, include the value for Base Directory or Virtual Directory that you typed in the encoding software’s Server information. If you created the live event with **G2SLTA**, and you typed a virtual path as part of the *livefile* value, include the virtual path.

Mount Points vs. Paths

You can’t determine which parts of a link refer to mount points and which parts refer to virtual directories just by looking at the link; you must examine the pages that list mount points to see which elements in a link are mount points.

File

Finally, you type the filename at the end of the link. The filename is either the name of the clip (in which case you must use the ramgen mount point—see the next section) or the name of a metafile.

Sharing Information for Links

You will need to give some information to the content creator so that she can create accurate links to the content she is creating. This information is summarized in the table below.

Who Provides Each Part of a Link	
Component	Supplied By
<i>protocol</i>	Content creator or RealServer administrator
<i>address</i>	RealServer administrator
<i>port</i>	RealServer administrator
<i>MountPoint</i>	RealServer administrator
<i>path</i>	Content creator
<i>file</i>	Content creator

Metafiles

Metafiles are text files that you link to in Web pages. The metafiles contain the names of the actual links. Pointing to a metafile in a link, rather than to a media clip, enables RealPlayer to contact RealServer. There are two types of metafiles:

- Ram files
- SMIL files

Ram Files and Ramgen

There are two ways to reference a clip in a link:

- Create a small metafile, known as a Ram file, and point to the metafile in the Web page link. The Ram file contains the true URL for the clip. Store the Ram file on a Web server.

- Use the Ramgen mount point in the Web page link, and include the actual clip file name in the Web page link.

Ram Files

Many browsers are not configured to start RealPlayer when a user clicks on a link to RealServer content. Because of this fact, links to RealServer content point to small text files, also known as metafiles. Web browsers can be configured to recognize this single file type and start RealPlayer. The metafile contains the “true” address of the media files, and RealPlayer can recognize these.

These metafiles are called Ram files. They are small text files that list one or more clips in sequence. They are similar in function to SMIL files, but cannot do the sophisticated presentations that are possible with SMIL.

A user can save the Ram file (by right-clicking on the link in the Web page) and use it to connect later (by opening it with RealPlayer), and skip the step of downloading it from your RealServer.

Ram files are often used for backwards compatibility with earlier versions of RealServer.

Additional Information

To learn more about Ram files, including options for start times, see *RealSystem Production Guide*. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

Ram File Format

A Ram file is simply a text file with the extension `.ram`. It can list the URL for a single clip, or it can give URLs for clips to be played in sequence:

Example Ram File

```
rtsp://address/file1  
rtsp://address/file2
```

Creating a Ram File that Lists RTSP and PNM

One reason to use Ram files in RealSystem 6 software is that most content uses the RTSP protocol, which earlier clients could not read. A Ram file can list more than one presentation type.

A Ram file that lists two different protocols for the same clip uses the following format:

Example Ram File with Two Protocols

```
rtsp://address/file
--stop--
pnm://address/file
```

Newer clients, such as RealPlayer 6 and later, stop reading a Ram file when they reach the word `--stop--`. Older clients look for the `pnm` instruction.

Ramgen: A Shortcut to Ram Files

As a shortcut to creating a Ram file for every single link you create, RealServer 8 is preconfigured with a mount point named `Ramgen`, which you can add to a link instead of creating a Ram file.

When RealServer receives a request that contains this mount point, it appears to create and send a Ram file automatically. RealServer simply converts the URL in the initial request to an URL within an HTTP message. The browser appears to download a file; the information is given to the client, which requests the correct links.

Additional Information

See *RealSystem Production Guide* for detailed information on using `Ramgen`. You can also include commands in the links that include `Ramgen` references; they are also described in *RealSystem Production Guide*. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

Use Either Ram Files or Ramgen—But Not Both

You must reference either a Ram file or `Ramgen` in a link. Some browsers are not configured to start the client when a SMIL or other streaming media file is requested, but all browsers launch the client when they receive Ram files.

SMIL Files

Synchronized Multimedia Integration Language (SMIL) is a mark-up language, based on an open standard, that specifies how and when each clip in

a file should be played. SMIL files can perform sophisticated layout and timing instructions.

Additional Information

Refer to *RealSystem Production Guide* for detailed information on creating SMIL files. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

Comparison of Ram Files, Ramgen, and SMIL

Ram File	Ramgen	SMIL
Can list multiple files	Links to single file	Can list multiple files
Can list files to be played in sequence	Lists only one file	Can list files to be played simultaneously
Cannot do any layout	Cannot do any layout	Performs sophisticated layout instructions
Can refer to version 6 content as well as previous versions	All content is version 6 and later (unless you use the altplay tag in the link)	All content is version 6 and later
You must create a special file	Fast way to test your content because you don't need to make a separate file	You must create a special file

For instructions on linking to SMIL files, see “SMIL Files” on page 377.

Where to Put On-Demand Clips

If you are just getting started with RealServer, store your media clips in the Content subdirectory of the main RealServer directory. These clips can be streamed immediately.

However, if you have many clips, it makes sense to organize them into subdirectories or even to store them on different computers. Links for these

files may become quite lengthy. Adding multiple mount points, with base paths that substitute for the lengthy paths, will shorten the links.

Summary of On-Demand Clip Locations

Location of Clips	Remarks
Content directory	When your clips are stored in this directory, links are easy to create. See “Storing Clips in the Content Directory” on page 78.
In a subdirectory of Content	Include the subdirectory name in the link. Refer to “Storing Clips in a Subdirectory of the Content Directory” on page 79.
In a different directory than Content (not a subdirectory)	Add a mount point that references the directory. Consult “Storing Clips in a Different Directory” on page 80.
On a completely different system	Configure your system to recognize the other location and add a corresponding mount point that references the other system and path. Reference “Creating Additional Mount Points” on page 80.

Storing Clips in the Content Directory

The Content directory is the main place to put clips. In the following example, the Content directory contains two clips (music.rm and music.rp) and two directories (Speeches and Concerts):

Example of Content Directory

Directory Structure	(main directory) Content music.rm music.rp Speeches Concerts
Values	Mount Point: / Base Path: C:\Program Files\Real\RealServer\Content (Windows), usr/RealServer/Content (UNIX)

The file named music.rm would have the following Web page link:

<http://realserver.example.com:8080/ramgen/music.rm>

In a Ram file, use a similar format, with a different protocol and without the ramgen mount point:

```
rtsp://realserver.example.com:554/music.rm
```

Storing Clips in a Subdirectory of the Content Directory

Files in the Content directory can be streamed without any special changes. But if you have a lot of files, you will probably want to organize them into subdirectories of the Content directory. When you do, be sure to include the names of the subdirectories in the link to the files. Substitute the subdirectories for *path* in the URL.

Example of Subdirectory of Content Directory

Directory Structure	<pre>(main directory) Content music.rm music.rp Speeches Concerts Classical bach.rm debussy.rm vivaldi.rm</pre>
Values	<pre>Mount Point: / Base Path: C:\Program Files\Real\RealServer\Content (Windows), usr/RealServer/Content (UNIX)</pre>

To refer to the file `debussy.rm`, located in the `Concerts/Classical` subdirectories, include the subdirectories in the link:

```
http://realserver.example.com:8080/ramgen/Concerts/Classical/debussy.rm
```

In a Ram file, use a similar format, with a different protocol and without the ramgen mount point:

```
rtsp://realserver.example.com:554/Concerts/Classical/debussy.rm
```

Tip

If you have many subdirectories within subdirectories, consider defining an additional mount point as a

shortcut. See “Creating Additional Mount Points” on page 80.

Storing Clips in a Different Directory

If you are going to store files in a directory that is not a subdirectory of the main base path, you will need to create a separate mount point for those files. The mount point functions as a shortcut for the path information. Use the new mount point in links to that content, in addition to any other appropriate mount points.

Tip

Choose a name for the mount point that reflects the type of content streamed from this location or its subdirectories.

► To stream on-demand files from a different directory:

1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.

2. Click **Add New**.

A generic mount point name appears in the Mount Points list and in the Edit Mount Point box.

3. Type the new mount point name in the **Edit Mount Point** box.

4. Click **Edit**.

5. Type a description in the **Description** box.

6. Give the location of the content in the **Base Path** box.

7. Click **Edit**.

8. Click **Apply**.

When you create a link for the content in the new mount point's base path, use the new mount point. If the content is in a subdirectory of the mount point's base path, include the mount point and the subdirectory in the link.

Creating Additional Mount Points

Uses for additional mount points include:

- Even if your content is all stored in the Content directory, an additional mount point can provide conceptual organization in links.

- If presentations are stored in obscure directories, a mount point can be a brief and sensible name in the link.

In the following example, assume a mount point has been defined as /music/, and that it refers to the actual Concerts directory:

Example Additional Mount Points

Directory Structure	(main directory) Content Speeches Concerts Classical bach.rm debussy.rm vivaldi.rm
Values	Mount Points: /music/ Base Path: C:\Program Files\Real\RealServer\Concerts (Windows), usr/RealServer/Concerts (UNIX)

A file named debussy.rm, located in the Classical subdirectory, would have the following link in a Web page:

<http://realserver.example.com:8080/ramgen/music/Classical/debussy.rm>

It would use the following URL if typed directly in a Ram file, SMIL file, or in RealPlayer's Open Location dialog box:

<rtsp://realserver.example.com:554/music/Classical/debussy.rm>

The full path to the file is not included. Instead, only the portion relative to the base path is shown.

► To create additional mount points:

1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.

2. Click **Add New**.

A generic mount point name appears in the Mount Points list and in the Edit Mount Point box.

3. Type the new mount point name in the **Edit Mount Point** box.

4. Click **Edit**.

5. Type a description in the **Description** box.

6. Give the location of the content in the **Base Path** box.
7. Click **Apply**.

When you create a link for the content in the new mount point's base path, use the new mount point. If the content is in a subdirectory of the base path, include the subdirectory in the link.

Recognizing Clips in a Different System

RealServer can stream files from any location that your operating system can recognize.

► To add a mount point for a different drive:

1. Use your operating system to identify the other drive or system.
 - In Windows systems, map a drive letter to the other drive.
 - In UNIX systems, mount the other system to a mount point.
2. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.
3. Click **Add New**.

A generic mount point name appears in the Mount Points list and in the Edit Mount Point box.
4. Type the new mount point name in the **Edit Mount Point** box.
5. Click **Edit**.
6. Type a description in the **Description** box.
7. Give the location of the content in the **Base Path** box, using the appropriate naming method for your operating system. (On Windows-based systems, if you are mapping to a drive letter, omit the trailing slash letter. For example, type G:, not G:\.)
8. Click **Apply**.

When you create a link for the content in the new mount point's base path, use the new mount point. If the content is in a subdirectory of the base path, include the subdirectory in the link.

Authenticated Clips

For files that will be authenticated (the user will be asked for a name and password—and possibly for other information— before being given access), the files must be placed in a completely different directory, one which is not a

subdirectory of Content. It's necessary to isolate secure material so that the RealServer authentication feature can perform the security checks before granting access.

The directory or directories that contain the secure material must be at the same level as Content, or at a higher level, or on a different system.

Example Secure Directory Structure

Directory Structure	(main directory) Content Speeches Concerts Secure TopSecret MembersOnly PayPerView
Values	Mount Points: /secure/ Base Path: C:\Program Files\RealRealServer\Secure (Windows), usr/RealServer/Secure (UNIX)

For instructions on how to set up authentication and the appropriate directories and mount points, refer to Chapter 15, "Authenticating RealServer Users".

Where to Put Live Clips

Live clips, created from a production tool such as an encoder, aren't physically stored anywhere, so their links don't usually include a path to an actual directory. The link to a live event may include a virtual path, as typed in the production tool. It may or may not correspond to any actual directories. For

live material, the path always begins with the /encoder/ mount point. See “Virtual Paths” on page 47 for an in-depth discussion.

Example Directory Structure

Directory Structure	(main directory) Content Speeches Concerts
Values	Mount Points: /encoder/

For example, a content creator encodes a live event and names it Speeches/Famous/Lincoln.rm. The Speeches directory is an actual directory in this case, but it has no Famous subdirectory. The virtual directory is Famous.

In a Web page, the link to the live clip would use the following format:

```
http://realserver.example.com:8080/ramgen/encoder/Speeches/Famous/Lincoln.rm
```

The link to the live clip would have the following format:

```
rtsp://realserver.example.com:554/encoder/Speeches/Famous/Lincoln.rm
```

Chapter 6

STARTING AND STOPPING REALSERVER

This chapter gives information on starting and stopping RealServer on both Windows and UNIX platforms. As soon as you start RealServer, it is ready to begin streaming.

Windows

In Windows NT, RealServer is automatically installed as a service, named RMsServer, unless you cleared that option during setup. As a service, RealServer is always running in the background.

Starting RealServer Manually

You can start RealServer from the **Start** menu or from a command line.

If RealServer is already running as a service, do not try to start it a second time. If you want multiple instances of RealServer, use the instructions in “Running Multiple Servers on One System Under Windows NT”.

- To start RealServer from the Start menu:

On the **Start** menu, click **Programs>Real>RealServer 8**. This starts the `rmserver.exe` program.

If this is the first time you have run RealServer, it loads the default configuration file.

- To start RealServer from a command line:

Move to the main RealServer directory and type the following at a command line and then press Enter:

```
Bin\rmserver  rmserver.cfg
```

Setting Up RealServer as a Service Under Windows NT

RealServer on Windows NT can be run as a service; an option during setup configures this automatically. Instructions in this section describe how to add RealServer to the services list if you did not instruct setup to do so.

► To install RealServer as a service:

1. At a command prompt, move to the RealServer Bin directory.
2. Import the configuration file you want to use into a specific key in the registry by typing the following:

```
rmserver.exe -import[:key] configuration_file
```

where:

key is the Registry key name you want to use. If you omit it, the default name Config is substituted.

configuration_file is the path and configuration file you want to import.

Note

The configuration file you use must contain absolute paths for variables such as BasePath. Relative paths will not be recognized by RealServer when it is run as a service.

For example, the following command:

```
rmserver.exe -import:Server1 ../rmserver.cfg
```

imports all the values in the rmserver.cfg file into the following key of the Windows NT registry:

```
HKEY_CLASSES_ROOT\Software\RealNetworks\RealMedia Server\7.0\Server1
```

Note

You must supply the path to the configuration file. If RealServer cannot find the configuration file, it may not start.

Tip

You can now start RealServer using this configuration by typing the following at a command line:

```
rmserver.exe registry:Server1
```

3. Install the service by typing the following command at a command prompt:

```
rmserver.exe -install[:ServiceName] "parameters"
```

where:

ServiceName is the name that will appear in the Services dialog box. If you omit *ServiceName*, RMServer is substituted.

parameters is either the name of the configuration file, or the registry and key name, as entered in Step 2. The format of the registry and key name is `registry:key`. Any command line parameters can be used.

Note

The quotation marks surrounding *parameters* are required.

The next time you start RealServer from the Services dialog box, it will use the settings specified in *parameters*, and will be configured to start automatically.

For example, the following command:

```
rmserver.exe -install:RMInternet "Server1"
```

installs RealServer with the service name “RMInternet” and uses the settings in the Server1 key.

4. Start the service. In the Services control panel, select the name you used for *ServiceName*, and click Start.

- To remove any RealServer from the services list:

At a command prompt, type the following:

```
rmserver.exe -remove[:ServiceName]
```

where *ServiceName* is the optional name of the service. If you omitted a service name when you installed the service, you can omit it here, and RealServer will use RMServer.

Additional Options for Windows NT

Under Windows NT, the following option is available:

- Installing multiple, separate instances of RealServer

Running Multiple Servers on One System Under Windows NT

You can load different configuration files into different Windows NT registry keys, and connect them to different instances of RealServer running as separate services. Multiple services of RealServer can be useful if you want to switch between a production and a test configuration file, for example.

- To import a configuration file into a specific key in the registry:
 1. Follow the instructions in Step 2 of “Setting Up RealServer as a Service Under Windows NT” to import a particular configuration file into a specific registry key.

2. Start RealServer by typing the following:

```
rmserver.exe registry:key
```

where:

key is name you want to use for the configuration. RealServer places the configuration information in

```
HKEY_CLASSES_ROOT\Software\RealNetworks\RealMedia Server\7.0\Key.
```

In the example from Step 2 of “Setting Up RealServer as a Service Under Windows NT”, in which the configuration settings are loaded into the “Server1” key, the full key name would be

```
HKEY_CLASSES_ROOT\Software\RealNetworks\RealMedia Server\7.0\Server1.
```

Stopping RealServer on Windows NT

If RealServer was started from the Start menu or the command prompt, switch to the command window and press **CTRL+C**.

In Windows NT, if RealServer was started as a service, stop RealServer through the Services control panel. Click **Start>Settings>Control Panel**. Double-click **Services**. Locate RMsServer on the list (your service name may be different), highlight it, and click **Stop**.

UNIX

Instructions in this section describe how to start and stop RealServer running under UNIX.

Starting RealServer on UNIX

Start RealServer initially with the default configuration file; later, you can create other configuration files and start RealServer using those.

RealServer includes one default port setting that is lower than 1000 (port 554 for the RTSP Port). Because the use of ports lower than 1000 requires that the person starting RealServer have root privileges, you must log in as root before you can start RealServer.

To run RealServer as a specific user, configure the User and Group variables with the appropriate User and Group names. Although your RealServer will start as root, it will automatically be switched to use the User and Group names you indicated. Refer to “UNIX-Only Features” on page 115.

- ▶ To start RealServer under UNIX:

Move to the main RealServer directory and type the following:

```
Bin/rmservice rmservice.cfg
```

If you do not start from the Bin directory, RealServer cannot understand the relative paths in the configuration file.

- ▶ To start RealServer in the background:

Type the following from the main RealServer directory:

```
Bin/rmservice rmservice.cfg &
```

If you have other configuration files, you can substitute their names for `rmservice.cfg` and RealServer will use the settings in the file you name.

- ▶ To limit the amount of memory that RealServer uses:

Start RealServer with the `-m` parameter:

```
Bin/rmservice rmservice.cfg -m 32
```

where the number after `-m` can be any amount of memory in megabytes, 32 or greater.

Stopping RealServer on UNIX

To stop RealServer under UNIX, obtain the parent process identification number, and then issue the `kill` command with that process number. The process ID is stored in the `rmservice.pid` file, which is usually kept in the `Logs` directory. The `PIDPath` variable specifies this location.

You can perform both actions with one command. Move to the directory that contains the RealServer PID file, and type the following:

```
kill `cat pidfile`
```

where `pidfile` is the name of the RealServer PID file, as shown in the `PIDPath` variable.

Chapter 7

CUSTOMIZING REALSERVER FEATURES

RealServer settings are customized through the RealSystem Administrator. This chapter describes how to use RealSystem Administrator as well as the basic settings used by all RealServers.

Overview

To monitor and modify your RealServer, use RealSystem Administrator. Any adjustments you make are stored in a configuration file. There are a few specific features that can only be adjusted by editing the configuration file directly. These are highlighted in “Features Available Only Through Direct Editing” on page 433.

Customizing RealServer Using RealSystem Administrator

RealSystem Administrator is the Web-based console for customizing RealServer features. It can be run from any browser on your network. It is password-protected when first installed, and you can create additional user names and passwords for any other people who will be helping you administer your RealServer.

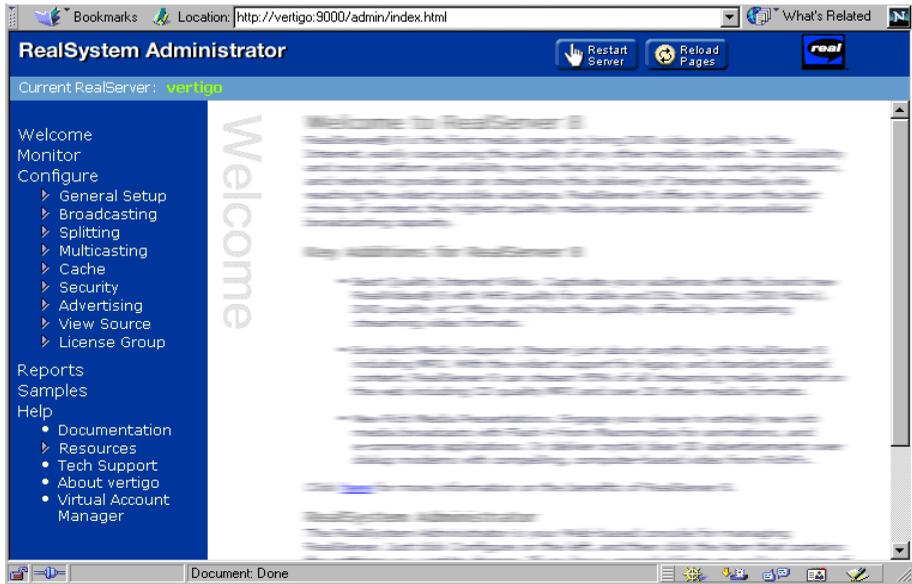
When the RealServer installation program completes, it asks if you want to start RealServer and run RealSystem Administrator. If you choose Yes, RealSystem Administrator asks you for a name and password, then it starts.

To make changes to any feature, click on the appropriate category listed under **Configure**. Make the changes and click **Apply**. A confirmation page appears to let you know that the changes have been made. You may be required to restart RealServer; a message to that effect will appear if it is necessary.

If your Web browser is set to permit cookies, RealSystem Administrator “remembers” the page that was open in the right-hand frame the last time you used it or when you click the refresh button. In Netscape Navigator,

RealSystem Administrator will reload with the main Welcome page when you resize the browser window unless cookies are enabled.

RealSystem Administrator Welcome Page



Starting RealSystem Administrator

You can view the configuration of your RealServer from nearly any browser on your network. Compatible browsers are Netscape Navigator version 4.06 or higher and Microsoft Internet Explorer version 4.0 or higher.

► To start RealSystem Administrator:

1. Start RealServer. See Chapter 6, “Starting and Stopping RealServer”.
2. In a browser, type the following address:

`http://address:AdminPort/admin/index.html`

where:

address is the IP address or host name of the machine on which RealServer is installed.

AdminPort is the port that RealSystem Administrator uses to connect to RealServer. You were asked for a port number during setup. Use that port number here.

The following URL will start RealSystem Administrator if it is typed in the browser on the same computer as RealServer (be sure to substitute your port number for *AdminPort*):

`http://127.0.0.1:AdminPort/admin/index.html`

The following command also works on the same computer:

`http://localhost:AdminPort/admin/index.html`

3. You are prompted for your user name and password; these will match the values you entered during setup. (To change these values, see Chapter 15, “Authenticating RealServer Users”.) Click **OK**.

RealSystem Administrator appears.

Tip

Bookmark this location so that you can easily return here at any time.

Using RealSystem Administrator

Once you have started RealServer and RealSystem Administrator, you can change RealServer features with the instructions below:

► To customize RealServer settings:

1. In RealSystem Administrator’s left-hand frame, click the appropriate category below **Configure**.
2. Change the values in the page on the right.
3. When you have finished changing values, click **Apply**.

If you made changes that require the Server to be restarted, the **Pending Changes** button at the top of RealSystem Administrator changes to red, and a red asterisk appears next to the field name that requires the restart.

In addition, the **Restart Server** button, located at the top of the RealSystem Administrator window, turns red if you need to restart RealServer for your changes to take effect. When you see the Restart Server button change its color to red, you should click it as soon as it is convenient.

Restricting Access to RealSystem Administrator

To ensure that only certain people can use RealSystem Administrator to make changes to RealServer, you can authenticate all connections to RealSystem

Administrator. Instructions are given in “Authenticating RealSystem Administrator Users” on page 228.

Configuration File

Changes made with RealSystem Administrator are stored in the configuration file, named `rmserver.cfg`. It is a text file formatted with tags that are based on XML (Extensible Markup Language). This language introduces great flexibility to the configuration file format and enables third-parties to use this file and add to its functionality. Syntax of this file is given in Appendix C, “Configuration File Contents”.

Be sure that your configuration file is stored where only authorized users can make changes to it.

Tip

Keep a backup copy of the configuration file. You may need it if you make changes to this file that you later want to undo or if you accidentally delete the working copy.

Editing the Configuration File with a Text Editor

A few specialized elements can only be changed by editing the file directly; they are noted in the text where they appear. In addition, third-party plug-ins may require their own parameters and variables, which cannot be added or modified through RealSystem Administrator; use a text editor to add them to the configuration file.

Additional Information

Appendix C, “Configuration File Contents” gives instructions on using a text editor to modify the configuration file directly.

Common Settings

Regardless of which features are in use, certain settings are used by every feature and apply to every RealServer. They are described in this section.

Port Numbers

Port settings tell RealServer where to listen for requests. Ports are described in detail in Chapter 3, “Overview of RealServer”.

If your RealServer and Web server are on the same machine, you may need to modify the HTTP Port setting. See “Running Web Servers and RealServer on the Same System” on page 112 for additional information.

Otherwise, you will probably not need to make any changes to the port settings.

Note

If you change the port numbers for **RTSP Port**, **PNA Port** and **HTTP Port** from their default values, you will need to tell your users so that they can include the new ports in their links. (If a link does not include a port number, RealPlayer uses default values for contacting the RealServer. But if RealServer is no longer listening on those ports, it will not receive the request.)

RealServer uses the following settings to determine where to listen for requests sent via a particular protocol (you can view the settings from RealSystem Administrator by clicking **General Setup>Ports**):

- **PNA Port**—the port where RealServer listens for material requested via PNA (such requests begin with `pnm://`). The default value is 7070. Earlier versions of RealSystem used this protocol.
- **HTTP Port**—the default value for this setting is 8080.
- **RTSP Port**—where RealServer listens for RTSP requests (these begin with `rtsp://`). At installation, the value is 554.

Note

To use a port lower than 1024 on a UNIX system, you must be logged on as super-user.

- **Monitor Port**—port used by Java Monitor. The default value is 9090.

- **Admin Port**—port number to which RealSystem Administrator connection requests are directed. The value for this setting is selected at random during setup to ensure security, and can be overridden by the user during setup.

Mount Points

Mount points on this page refer to on-demand clips. For a complete description of mount points, see “Mount Point” on page 71. Mount points in other sections, such as for live material, are described in their respective chapters.

You do not need to change this setting unless you want to keep your media clips somewhere other than the Content directory or its subdirectories.

RealServer uses the following mount points when it is first installed:

- **Main mount point** (represented by /)—This refers to all content streamed by this RealServer
 - **secure**—Content that will be authenticated is identified with this mount point
- To change the base path of the main mount point:
1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.
 2. In the list on the left, select /.
 3. To change the base path, type the new path in the **Base Path** box.
 4. Click **Apply**.
- To add another mount point:
1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.
 2. Click **Add New**.
A generic mount point name appears in the Edit Mount Point box.
 3. Type the new name in the **Edit Mount Point** name box. It must be unique.
 4. Click **Edit**.
 5. Give a description for this new mount point in the **Description** box.
 6. Identify the location of the content by typing the full path in the **Base Path** box.
 7. Click **Apply**.

MIME Types

MIME Types on a Web Server

RealServer works with any Web server that supports configurable MIME types. Make sure that your Web server has the RealNetworks MIME types defined.

Refer to the instructions accompanying your Web server to define the following MIME types on your Web server. Of the items on this list, only MIME types for the extensions `.ram` and `.rpm` are required. Other applications may use other types shown in the table.

Web Server MIME Types and Extensions

MIME Types	Extensions
audio/x-pn-realaudio	ra, rm, or ram
audio/x-pn-realaudio-plugin	rpm
application/x-pn-realmedia	rp
application/smil	smi or smil
application/sdp	sdp
image/gif	gif
image/jpg	jpg, jpeg
text/html	html, htm

MIME Types on RealServer

In addition, RealServer acts as a Web server for certain features. To this end, RealServer has its own MIME types section. You should only modify the list of MIME Types if you will be streaming a data type via HTTP that is not on the list. The following table shows RealServer's initial settings.

RealServer MIME Types and Extensions

MIME Types	Extensions
audio/x-pn-realaudio	ram
image/gif	gif
image/jpg	jpg, jpeg
text/html	html, htm

Chapter 8

ADVANCED FEATURES

This chapter covers features that are specific to the operating system, such as allowing users to view the source code of SMIL files and media clips, as well as reserving IP addresses for RealServer's use, running RealServer on the same system as a Web server, and working with firewalls.

Displaying Source Code for SMIL Files and Media Clips

Just as users can right-click an HTML file and view the HTML code that was used to create a Web presentation, RealServer contains a feature that enables users of RealPlayer 7 and later to view the source code for SMIL presentations or information about media clips. When a user right-clicks on a presentation or selects **View>Clip Source**, RealServer sends a Web page that contains the text of the SMIL file, or data about the clip, to the user's browser.

Users can then “learn by example” and understand how to create their own SMIL files. Content creators will find this feature useful when they troubleshoot SMIL files.

View Source on SMIL Files

The HTML page that displays the text of the SMIL file also links to information about each clip referenced by the SMIL file. The information shown on these pages describes the contents of the entire file, including information such as file size, buffer time, and bit rate.

RealServer sends the text of the SMIL file to the browser using an automatically generated Web page. In the browser's address box, the URL shown is:

`http://realserver.example.com:8080/viewsource/template.html?ABcdlkj293847`

By default, the name and path of the SMIL file are not shown; random numbers and letters are displayed instead. (You can make the SMIL path and

file name appear; use the instructions in “Allowing Users to See Complete Paths in SMIL Files” on page 103.)

Security

Within the text of the SMIL file in the browser, all references to other files are shown as hyperlinks. Clicking these links displays another Web page, with detailed information about each referenced file.

This feature is especially helpful for content creators, as it also enables them to see detailed information about the components of the SMIL file.

To protect the location of your content, this feature is initially configured to omit the full path of the clips referenced in the SMIL file, showing an ellipses (...) instead. You can also disable this feature for some or all paths.

Note

For content on your local computer, paths are always shown. They cannot be hidden.

Listing All On-Demand Content

The content browsing feature creates a Web-based directory with links to all on-demand content available on to your RealServer. By clicking **View Source>Browse Content Now** in RealSystem Administrator, you generate the index. Only other administrators who know the correct URL, user name, and password for RealSystem Administrator can view on-demand content available to this RealServer.

A Note About Web Servers

Although it is not the preferred delivery method, some content creators serve SMIL and media from Web servers. When a user selects View Source for content delivered by a Web server, the paths that appear in the SMIL file are hidden. Source information is available for all other media clips delivered by the Web server. Since RealServer has no control of Web servers, settings used for the View Source feature in RealServer have no effect on how the source code is displayed for content served by Web servers.

View Source Used with Other Features

The view source feature interacts with other RealServer features.

Streaming, Unicasting, and View Source

The view source feature applies to both on-demand and live content.

The browse content feature applies only to on-demand content.

Archiving and View Source

The browse content feature is a good way to take inventory of your Archive directory.

G2SLTA and View Source

On-demand files that are converted to live files through the use of **G2SLTA** show the same information as any other live files.

Splitting and View Source

View source is disabled for users who are receiving a broadcast through a backup push splitting source.

Access Control, Authentication, and View Source

The view source feature is automatically disabled for all secure content.

Reporting and View Source

A record is created in the access log when a user makes a view source request. See the “Summary of GET Statements” table on page 300.

Changing View Source Settings

In RealSystem Administrator, click **View Source**, then click **Source Access**. At installation, the settings are:

- **View Source**—Set to Yes, for the main mount point (/)
- **Hide Paths**—Set to Yes (paths will be hidden)

Optional View Source Features

The view source feature has these options that you can customize:

- Displaying source code only for certain streams

- Allowing users to see complete paths in SMIL files
- Temporarily overriding individual path settings

Displaying Source Code Only for Certain Streams

You can choose to enable the view source feature for a limited number of paths, or, conversely, to enable it for most paths but disable it for only a few.

► To enable view source for a few specific streams:

1. In RealSystem Administrator, click **General Setup**. Click **Source Access**.
2. In the **Paths** list, select the single forward slash (/).
3. From the **View Source** list, select No.
4. Click **Add New**.

A generic path name appears in the **Edit Path** box.

5. In the **Edit Path** box, replace the generic path with the name of a path for which you want to enable the view source feature.
6. Click **Edit**.
7. In the **View Source** area, select Yes.
8. Repeat Step 4 through Step 7 for each path you want to enable.
9. In the **Master Settings** area, from the **View Source** list, make sure Use Settings Above is selected.
10. Click **Apply**.

► To enable view source for most streams, but disable it for a few:

1. In RealSystem Administrator, click **General Setup**. Click **Source Access**.
2. In the **Paths** list, select the single forward slash (/).
3. From the **View Source** list, select Yes.
4. Click **Add New**.

A generic path name appears in the **Edit Path** box.

5. In the **Edit Path** box, replace the generic path with the name of a path for which you want to disable the view source feature.
6. Click **Edit**.
7. In the **View Source** area, select No.

8. Repeat Step 4 through Step 7 for each path you want to enable.
9. In the **Master Settings** area, from the **View Source** list, make sure Use Settings Above is selected.
10. Click **Apply**.

Temporarily Overriding Individual Path Settings

In the Master Settings area, the settings for View Source and Hide Paths are normally Use Settings Above. By selecting one of the other two options, Disable View Source or Enable View Source, or Show All Paths or Hide All Paths, you can leave the settings of the individual paths intact, but supersede them with the new values. This can be useful for temporary troubleshooting, or for disabling the feature quickly and globally.

Of course, these settings do not have to be temporary. They stay in effect until you change them.

Allowing Users to See Complete Paths in SMIL Files

At installation, view source is configured to “hide” the paths of the clips referenced in SMIL files. This protects the privacy of the content creators, and enables the user to focus on the syntax used within the SMIL file.

For example, the following tag within a SMIL file:

```
<video src="rtsp://realserver.example.com/presentation/presentation.rm"
region="VideoRegion">
```

appears as the following:

```
<video src="rtsp://.../presentation.rm" region="VideoRegion">
```

Identifying information is removed, and only the protocol and file name are shown.

► To include full path information:

1. In RealSystem Administrator, click **View Source**. Click **Source Access**.
2. In the **View Paths** list, select the path for which you want to allow users to see the source.
3. In the **Hide Paths** list, select No.
4. Click **Apply**.

Browsing Your Content

The content browsing feature creates a Web-based list of all on-demand content that your RealServer can stream.

If the clips are stored on your hard drive, full paths are always shown.

- ▶ To browse the on-demand content:

In RealSystem Administrator, click **View Source**. Click **Browse Content Now**.

A new browser window appears, containing two columns. Along the left column, labelled Info, the word “Directory”, “MountPoint”, or a file size appears. The next column shows a file name. The third column contains a link to the file.

Changing Content Browsing Settings

In RealSystem Administrator, click **View Source**, then click **Content Access**. At installation, the settings are:

- **Mount Points to Browse**—Set to /, for the main mount point (/)
- **Extensions to Browse**—Set to * (all file types will be browsable)

Optional Content Browsing Settings

- ▶ To browse more mount points:

1. In RealSystem Administrator, click **View Source**. Click **Content Access**.
2. Click **Add New**.
A generic mount point name appears in the Edit Path box.
3. In the **Edit Path** box, type the name of a mount point that you want to be able to browse.
4. Click **Edit**.
5. If you want to limit the file types that are included in the generated list, type them in the **Extensions to Browse** box. For example, you might want to include just the extensions smi and smil.
6. Click **Apply**.

RealServer Caching Features

RealProxy is software that stores streamed media. While it is not part of RealServer 8, it can work with RealServer to share the distribution load,

thereby conserving bandwidth over an intranet and allowing RealServers to send streams to a wider audience. It is generally installed on an intranet or on a large Internet Service Provider (ISP). When a client on the intranet or hosted by the ISP requests a streamed media file, RealProxy intercepts the request and sends it on behalf of the client. RealProxy then stores the requested media and streams it to any other clients who subsequently request the same material.

RealServer is designed to work with RealProxy. RealServer is configured at installation to allow all content to be cached by RealProxy. This ensures that clients whose requests are sent via RealProxy will be able to view your content. Also, because more than one RealProxy is now rebroadcasting some of your content, your RealServer now has more connections available.

Only on-demand content can be cached by RealProxy.

Caching used with Other Features and with RealProxy

This section describes how RealServer interacts with RealProxy software.

Streaming On-Demand Clips and RealProxy

All on-demand clips are automatically available to media caching software. If there is content served by your RealServer that you do not want to be cached by a media cache, you can mark it as non-cacheable, on a per-file or per-folder basis.

Unicasting, Splitting, Multicasting and RealProxy

Live clips are not available to media caching software; RealProxy will still proxy the live broadcasts for clients. RealServer acts as a source for pull splitting, and RealProxy acts as a receiver.

Access Control and RealProxy

RealServer does not see the IP addresses of the individual clients that request content; instead, it sees the IP address of the RealProxy. You can prevent specific RealProxys from requesting material on behalf of clients (see “Preventing Certain RealProxys from Accessing Your RealServer” on page 108). But if you do, you will also prevent all those clients from accessing your clips.

Authentication and RealProxy

Before allowing clips to be cached, RealServer verifies whether the client's IP address is valid. If the requested material is marked as secured, it then performs any necessary authentication checks.

Authenticated material can be stored in a RealProxy cache, but the client will be authenticated with the source RealServer every time it tries to access the stored clip.

ISP Hosting and RealProxy

All on-demand material served on behalf of ISP-hosted customers can be cached, unless you mark those directories as non-cacheable (see "Preventing Some Paths and Files from Being Cached" on page 107).

Monitoring and RealProxy

The Java Monitor will show the IP address of the caching software as it plays a clip. The caching software is not identified as such; rather, it appears to be a client.

Reporting and RealProxy

All client requests for streaming media are recorded in RealServer's access log, as if they were made directly by clients and not sent through RealProxy. In addition, a separate log file, `cache.log`, records all clips that were accessed by RealProxy. The `cache.log` file can give you an idea of which content is most requested by media caches.

The access log will show a record for the request made by the cache software, and for every client request.

The access log and the cache log are independent of each other.

Additional Information

The cache log is explained in Chapter 19, "Reporting RealServer Activity".

Ad Streaming and RealProxy

All material served through the ad streaming feature is cacheable, unless you mark those directories as non-cacheable (see "Preventing Some Paths and Files from Being Cached" on page 107).

Changing Cache Settings

On the **Cache** page (located by clicking **General Setup**), the **Cache Port** number 7802 is shown. RealProxys will send their requests to this RealServer port.

If you change this value, requests by RealProxy will not be accepted by RealServer, and therefore will not be cached unless you share the new port number with the administrators of all RealProxys that are accessing your streams.

Optional Caching Features

Unless you specify otherwise, all material on your RealServer is available to RealProxy. RealServer has these options for restricting which RealProxys can cache streams:

- Preventing some paths and files from being cached
- Preventing certain RealProxys from accessing your RealServer
- Preventing all caching

Preventing Some Paths and Files from Being Cached

You can restrict the paths and files RealProxys can store. If RealServer receives a request for material included in the **No-Cache Paths** list, it streams the file directly to the client rather than allowing it to be cached and re-transmitted. As always, RealServer records the transaction in the access log, and reports a download size of 0 bytes in the cached requests log file.

For example, you might choose to prevent material in authenticated content locations from being cached. Or, you might put the path to time-sensitive clips on this list so that it cannot be stored by RealProxy.

Note

Media caching software makes more streams available on your RealServer. If you limit which clips can be cached, you also limit how many clients you can serve.

- To prevent RealProxy from caching material on your RealServer:
 1. In RealSystem Administrator, click **Cache**. Click again on the word **Cache** that appears below it.
 2. In the **No-Cache Paths** section, click the **Add New** button.
A generic path name appears in the Edit No-Cache Paths box.

3. In the **Edit No-Cache Paths** box, type the name of the path or file whose content you want to restrict.

For example, if a subdirectory of the Content directory contained a directory named News, you would add /News to the No-Cache Directory box. If you only wanted to prevent the late-breaking news clip from being cached, you would add that to this list instead: /News/breaking.rm.

4. Click **Edit**.
5. Repeat Step 2 through Step 4 for each path or file name that you do not want cached.
6. Click **Apply**.

Preventing Certain RealProxys from Accessing Your RealServer

You can indicate that certain RealProxys are not allowed to cache any of your material. To do this, you must know the IP address of the machine on which RealProxy is installed.

Tip

Look in the cache.log file to find the IP addresses of cache software that is accessing your content.

- To prevent certain RealProxys from making requests:

Create an access rule for the RealProxy you want to restrict. In addition to specifying the IP address, indicate the port number to which access should be denied (usually 7802).

Additional Information

To learn about limiting access to your RealServer according to the IP address of any other computer, see “Limiting Access with the IP Address” on page 214.

Preventing All Caching

- To prevent all caching of all material from all clients and RealProxys:
 1. In RealSystem Administrator, click on **Cache**. Click again on the word **Cache** that appears below it.
 2. In the **Cache Requests** list, select **Disabled**.
 3. Click **Apply**.

Distributed Licensing

The distributed licensing feature enables you to purchase a single license to be used by multiple RealServers on your network, that share a pool of streams to be used for specific features. The authentication and ad streaming features can both use this feature.

The RealServers that share a license are called a *license group*. Clients are not denied service unless the entire pool of connections in the group is in use.

When you use this feature, you place the main license file on the main RealServer (called the *publisher*). You configure the publisher RealServer, then configure corresponding RealServers (called *subscribers*). The subscribers can be set up to look for a primary publisher RealServer, and if that RealServer is not available or has too few connections available for use, a secondary RealServer publisher can be called upon.

Note

Within a given network or organization, you can have multiple groups, but you should not use this feature outside of a network or across a firewall.

Each RealServer maintains its own configuration file, and can be configured independently using RealSystem Administrator. The features available to each subscriber depend on the features permitted by the shared license, but can be configured independently to use different values.

Setting Up Distributed Licensing

To use this feature for the first time requires three steps:

1. Setting up the publisher RealServer.
2. Setting up the subscriber RealServers.
3. Starting the license group.

Once you have configured both publisher and subscribers, you must remember to start the publisher before starting the subscribers. If you start the subscribers first, they will operate with minimum settings until the publisher comes online. For a list of these settings, see “Minimum Licensed Features” on page 39.

Setting Up the Publisher

In addition to the steps shown below, you must tell the person who is setting up the subscriber what value you are using for Admin Port.

► To set up the publisher:

1. On the publisher, start RealSystem Administrator.
2. Look up the value of **Admin Port**:
 - a. In RealSystem Administrator, click **General Setup**. Click **Ports**.
 - b. Look at the value for **Admin Port** and make a note of it. You'll use it in configuring the publisher.
3. In RealSystem Administrator, click **License Group**. Click **Configure**.
4. List all license publishers for this group, including this publisher:
 - a. In the License Publishers area, click **Add New**.
A generic IP address appears in the Edit Publisher IP Address box.
 - b. In the **Edit Publisher IP Address** box, type the IP address of the other license publisher.
 - c. Click **Edit**.
 - d. In the **Admin Port** box, type the value of the other publisher's Admin Port.
5. Repeat Step 4 to add each additional publisher in the group.
6. List the subscribers that will be sharing this RealServer's license (repeat this step for each subscriber):
 - a. In the License Subscriber area, click **Add New**.
A generic description appears in the Edit Subscriber box.
 - b. Type a descriptive name in the **Edit Subscriber** box, to distinguish this subscriber from others.
 - c. Click **Edit**.
 - d. In the **Subscriber Address** box, type the subscriber's IP address. To indicate a range of addresses, type the subscriber's subnet mask (usually 255.255.255.255) in the **Subscriber Netmask** box.
7. Click **Apply**.
8. Repeat these instructions for each publisher in the license group.

Setting Up Subscribers

You'll need to know the value for each publisher's Admin Port, as well as each publisher's IP address.

► To set up the subscribers:

1. In RealSystem Administrator of the subscriber, click **License Group**. Click **Configure**.
2. List the license publishers for this subscriber:
 - a. In the License Publishers area, click **Add New**.
A generic IP address appears in the Edit Publisher IP Address box.
 - b. In the **Edit Publisher IP Address** box, type the IP address of the license publisher host.
 - c. Click **Edit**.
 - d. In the **Admin Port** box, type the value of publisher's Admin Port.
3. Click **Apply**.
4. Repeat these steps for each subscriber in the license group.
5. Restart the publisher.
6. Restart the subscribers.

Distributed licensing is now active; RealServers can pool their stream count and feature availability.

Monitoring Distributed Licensing

On the publisher, you can use a license group monitor to view the number of connections currently in use by the subscribers.

► To monitor license activity:

On the publisher, in RealSystem Administrator, click **License Group**. Click **Monitor**.

The Monitor page appears, and shows the number of publishers, subscribers, and connections currently in use.

Reserving IP Addresses for RealServer's Use

When RealServer starts, it uses the IP address assigned to the machine's host name.

You can configure RealServer to always use a specific IP addresses by setting up the IP Binding list. Within this list, you cite individual addresses to use, or you can bind to all the IP addresses available on the RealServer machine.

► To reserve IP addresses for RealServer:

1. In RealSystem Administrator, click **General Setup**. Click **IP Binding**.
2. Click the **Add New** button.

A generic address appears in the Edit Address box.

3. In the **Edit Address** box, type the IP address that you want RealServer to use.

To capture all addresses for RealServer's use, add the IP address of 0.0.0.0, and delete any other addresses. RealServer will automatically bind to all addresses and to localhost (127.0.0.1).

Tip

Binding to all addresses, by using 0.0.0.0, is recommended for most administrators.

If you type a specific address, RealServer will bind to the specified address only; it will not bind to localhost.

Warning

Use either 0.0.0.0 or a specific address, but not both. If you use both, RealServer will not start.

4. Click **Edit**.
5. Repeat Step 2 through Step 4 for each address on this machine that you want RealServer to use.
6. Click **Apply**.

If you leave the **IP Address** box blank, RealServer binds to the host IP address and localhost. It does not bind to any other addresses.

Running Web Servers and RealServer on the Same System

If you install RealServer on the same system as your Web server, you may need to complete additional steps. Most Web servers use port 80 for HTTP requests. At installation, RealServer's default HTTP Port is 8080, but if you configure

RealServer to use port 80 (the same port as the Web server), problems may ensue. You may have to perform the following steps:

- Choose a different port for RealServer to use for HTTP requests and change links that point to HTTP pages
- Reserve an IP address for RealServer

Change the HTTP Port Value

Because RealServer can serve requests for HTML pages sent via HTTP (such as RealSystem Administrator), if RealServer is on the same system as a Web server, requests that begin with `http://` may be misdirected. When a user clicks a link that begins with `http://` and it does not contain a port number, the client supplies a port number of 80. When the Web server and RealServer are on the same machine, the Web server will attempt to serve the file. If the link points to what's meant to be a RealSystem presentation, the Web server will not find the file and will display the error message "File not found".

To prevent this problem from occurring, make sure the HTTP Port value is not the same as the port number your Web server is using. The default value is 8080. Most Web servers use port 80. Be sure that you include RealServer's HTTP Port number in the URL.

Set IP Binding List

You may need to reserve at least one IP address for RealServer's use. Assign RealServer and the Web server to individual addresses, so that they can both use port 80. See the "Distributed Licensing" section earlier in this chapter.

Port Hinting

Client software (such as RealPlayer) always tries to receive RealServer content via the RTSP or PNM protocols. But if it cannot use either of these (for example, if a firewall does not permit those types of traffic), it will signal RealServer to send the data by wrapping it in the HTTP protocol. This method is known as "HTTP cloaking". While this method does not provide the best possible user experience, it does allow users to receive streamed media who would otherwise be blocked.

The new port hinting feature enables the content creator to specify ports to use when the client requests a cloaked stream, or to automatically send the port information when `ramgen` is included in the link. Thus a content creator can add any of the following lines to an URL:

```
?cloakport="port1"  
?cloakport="port1 port2 port3"  
?cloakport="port1, port2, port3"
```

where *port1*, *port2*, and *port3* are ports that your RealServer uses (generally 554, 7070, and 8080). The ports will be tried in the order they're listed.

Add the cloakport switch to the full URL:

```
http://www.example.com:8080/ramgen/clip.rm?cloakport="554, 7070, 8080"
```

A user who clicks this link will now be able to receive a cloaked stream (should one be needed) more quickly than in previous versions of RealServer.

Or, the content creator can omit this information, and RealServer will include it automatically if ramgen is used in the link.

This feature is used only by RealPlayer 8.

Setting Up Port Hinting

RealServer is configured for port hinting by default. To inspect the settings it uses, click **General Setup>Ports**. The following setting is used:

- **Enable Ramgen Port Hinting URLs**—this is set to Yes by default.

A single variable controls this feature in the configuration file:

```
<Var CloakingHint="1"/>
```

where 1 means enabled, and 0 means disabled.

Features Specific to the Operating System

While RealServer functions nearly identically on both Windows NT and UNIX platforms, there are a few differences that allow you to take advantage of unique characteristics of each operating system. These features and settings are optional.

Windows NT-Only Features

This section describes features unique to RealServer running on a Windows NT system.

Windows NT Service

When you install RealServer, you have the option to install it as a service. You can also configure this later. Several RealServers can be run from the same machine, with different configuration files.

Additional Information

See “Setting Up RealServer as a Service Under Windows NT” on page 85.

Windows NT Performance Monitor

RealServer comes with a file to use with the Windows NT Performance Monitor, so that you can use the Windows NT method of monitoring RealServer performance.

Additional Information

See “Optional Java Monitor Features” on page 275.

Windows NT Event Viewer

RealServer information and errors are displayed in the Windows NT Event Viewer.

UNIX-Only Features

This section describes features unique to RealServer running on a UNIX system.

User and Group Variables

The User setting indicates the user name under which RealServer runs. The user name must exist on the computer on which RealServer is running; otherwise, RealServer will not start.

If you do not specify a user name when installing RealServer, the user name defaults to the user name of the user who first logs in and starts RealServer; this is accomplished with the default value of `%-1`.

The Group variable gives the group name under which RealServer runs. The group name must already exist on the computer on which RealServer is running; otherwise, RealServer will not start.

If you do not specify a group name, this variable defaults to the group name of the user who first starts RealServer.

Note

Be sure that the user or group name you assign has write permissions for the Logs and Secure directories.

► To change the group or user names:

1. In RealSystem Administrator, click **General Setup**. Click **User/Group Name**.
2. Type the correct user name or ID number in the **User Name or ID** box. The default is %-1, which means RealServer uses the user name of the user who logged in and started RealServer.
3. Type the correct user name or i.d. number in the **Group Name or ID** box. The default is %-1, which means RealServer uses the group name of the user who logged in and started RealServer.
4. Click **Apply**.

Process ID (PID)

RealServer creates a text file that stores the current value of the process ID of the parent RealServer process, `rmserver`. The file is stored in the directory indicated by the `PidPath` variable, and is named `rmserver.pid` at installation. If `PidPath` is omitted from the configuration file, RealServer stores the information in the directory specified by the `LogPath` variable.

SIGHUP

Some changes that you make to RealServer require that RealServer re-read the changes while still running. Other changes require that RealServer be restarted. If you use RealSystem Administrator to change settings, it will either force RealServer to re-read the configuration file while RealServer is still running (thus preserving all connections), or it will display a message instructing you to restart the Server at your convenience.

If you make changes to the configuration file manually, you will need to instruct RealServer to re-read the configuration file. This is possible for RealServer running on a UNIX platform with the **SIGHUP** command. Use the following command at a command prompt:

```
kill -HUP processID
```

where *processID* is the RealServer process number, as shown in the `Logs/rmserver.pid` file.

Chapter 9

FIREWALLS AND REALSERVER

Firewalls can inadvertently or deliberately block streaming media presentations, so familiarity with your network's firewalls will help you stream successfully. This chapter may also help you answer questions from users who are experiencing difficulties due to firewall issues.

Overview

A firewall is a software program or device that monitors, and sometimes controls, all transmissions between an organization's internal network and the Internet. Whether the network consists of a company's local area networks, wide area networks, and the Internet, or an Internet Service Provider, a firewall is typically deployed to prevent inappropriate access to the files, data, or machines of its customers. The firewall's role is to ensure that all communication, in both directions, conforms to the organization's security policies.

Often, firewalls permit one-way outbound access to the Internet. Because RealServer and the client need to establish two-way communication to stream and receive media content, firewalls may reject a client's attempt to establish this connection, and the client's request for a clip will be rejected by the firewall.

This chapter explains why you cannot serve any content to users on the Internet if your RealServer is behind a firewall, and shows where to move RealServer so that it can serve content while staying within a perimeter network of protected machines.

Who Should Read This Chapter

The next sections discuss the different possible firewall arrangements and illustrate how RealServer works with them. This information will be of interest to anyone who wants to know:

- where to place RealServer in relation to a network firewall

- why a firewall that allows RTSP and PNA traffic provides the best user experience
- why some clients are unable to receive your clips, or receive them at poor bit rates
- what some of the issues are in working with encoders or receivers on opposite sides of a firewall

More information on firewalls is available from the RealNetworks Web site at <http://service.real.com/firewall>.

For information on configuring a specific firewall product, consult the firewall software's documentation.

Highlights of This Chapter

If a Server is behind a firewall, it can only stream content to other users behind the firewall. It cannot stream over the Internet to users on the other side of the firewall.

Additional Information

Read "How Firewalls Can Affect the User Experience" on page 121.

For a Server that is streaming or broadcasting over the Internet, the best location is in the perimeter network, sometimes known as the de-militarized zone (DMZ).

Additional Information

See "Locating RealServer Near the Firewall" on page 132.

The firewall that provides the best user experience is one that allows RTSP and PNA application-layer traffic, and that allows use of the UDP transport protocol.

Additional Information

Refer to "Summary of Firewall Types and Characteristics" on page 131.

Firewalls and Their Interaction with RealServer Features

Streaming content to a client—whether via on-demand streaming or any of the live delivery methods— is straightforward, as described in this section.

On-Demand Streaming and Firewalls

Issues that clients may have in connecting to on-demand streams are described in “Communicating with Clients” on page 123.

Live Unicasting and Firewalls

Clients connect to live broadcasts in the same way they connect to on-demand streams. However, the encoder that supplies RealServer with its live data may not be able to connect to RealServer if a firewall exists between the encoder and RealServer. See “Communicating with Encoders” on page 126.

Splitting and Firewalls

Working with receivers that are located on the other side of a firewall requires special consideration, described in Chapter 12, “Splitting Live Presentations”.

Multicasting and Firewalls

Multicasts usually take place within an intranet, where broadcasts are not travelling outside a firewall. If a multicast is occurring through a firewall, the firewall must be specially configured to allow multicast traffic.

Caching and Firewalls

RealProxy connects to your RealServer just as any other client would. In addition, it uses two TCP connections to store media in the cache.

Access Control, Reporting, and Firewalls

When a firewall exists between a client and RealServer, the IP address that appears in the access log’s *client_IP_address* field may not be the true client address, and you might not get an accurate idea of exactly which clients are viewing material streamed by your RealServer. See the “Streaming Media Over the Firewall Types” table on page 131 for a list of which firewalls replace the client’s address with their own.

Authentication and Firewalls

Requests by the Server for authentication information (either from the user or the client software) is delivered over the control channel. If a firewall prevents

the control channel connection, RealServer cannot authenticate the request and therefore will not deliver it.

ISP Hosting and Firewalls

If there is a firewall between users and the location where they are to store their content for hosting, they may not be able to send their clips to the Server.

Protocols Used by RealServer

RealServer uses two connections, known as *channels*, to communicate with clients: one for communication with the client, and one for actual data. The communication channel is known as the *control channel*, since it is over this line that RealServer requests and receives passwords, and the client sends instructions such as play, pause, and stop. Media is actually streamed over a separate *data channel*.

RealServer uses two sets of protocols in transmitting its data.

- For the control connection, RealServer uses the two-way Transmission Control Protocol (TCP) protocol.

The TCP protocol guarantees delivery of packets, which is important for control information and error-checking. It has built-in congestion control, but it is slow to respond to changing network conditions. Because TCP is a two-way connection protocol, the client and RealServer can communicate about passwords; the user can press pause or fast-forward and the information is sent over the TCP connection. However, verification that each set of instructions reached its intended destination consumes some overhead.

- For the data connection, RealServer uses the one-way User Datagram Protocol (UDP) protocol by default. If the firewall blocks UDP traffic, RealServer will use TCP instead.

UDP packets are sent in one direction only. Because the transport does not perform error checking, it can deliver the packets faster than TCP does.

The characteristics of TCP that make it suitable for control information also make it less appropriate for continuous data delivery. The overhead used in TCP is not optimized for the delivery of streaming media.

The quality of the stream received by a client is related to the transport protocol in use.

RealServer uses two main application-layer protocols to communicate with clients: RTSP (Real Time Streaming Protocol) and PNA (Progressive Networks Audio). These protocols work with the two-way TCP connection to send commands from the client such as “start” and “pause,” and from RealServer to clients for information such as the clips’ titles. A third protocol, HTTP, is used in sending other types of data.

- RTSP is a client/server protocol designed specifically for serving multimedia presentations. It is an open standard, one that is very useful for large-scale broadcasting. Only RTSP can deliver SureStream™ files, which use multiple bandwidth encoding, and automatically choose the best available presentation for the user’s available bandwidth.
- PNA is the proprietary client/server protocol designed and used in earlier software versions. The ability to serve via PNA is supported in RealServer 8 for compatibility with older versions of RealPlayer.
- HTTP is the protocol used for metafiles (or dynamically generated metafiles) that point to RealServer content, and for the HTML pages served by RealServer (such as the Web-based RealSystem Administrator). It may also be used in delivering clips to clients that are located behind firewalls.

Control and Data Channel Protocols

Control Channel Protocol	Data Channel Protocol
RTSP	TCP and UDP, or TCP only
PNA	TCP and UDP, or TCP only
HTTP	TCP only

As we will see later in this chapter, the single TCP protocol may be used if a firewall does not permit UDP connections that originated outside the firewall.

How Firewalls Can Affect the User Experience

Firewall security policy stops all traffic, and allows only those services that the firewall administrator has specifically designated. Typically, HTTP traffic and TCP-based traffic that initiates inside the organization is allowed to pass through the firewall.

A client (such as RealPlayer) that tries to request streamed media through such a firewall will initially be rejected by the firewall, because it is attempting to use UDP, which is not allowed by this type of firewall.

The client is aware that it isn't able to establish a connection with the Server using UDP as the default transport protocol. At this point, the client will try alternate protocols for data delivery, such as TCP or HTTP.

Use of HTTP through the firewall is likely to succeed, since most firewalls are configured to allow HTTP traffic.

RealServer can deliver streaming media over HTTP in two unique ways: by wrapping the RTSP or PNA protocol stream with the HTTP protocol, or by downloading the presentation via HTTP. However, neither of these methods provides the best possible user experience.

Potential Problems with Firewalls

If a firewall separates any of the RealSystem component software packages that communicate with each other—such as encoders, Servers, or clients—the delivery of data may not be at an optimal rate.

- Firewalls configured to only allow TCP traffic may cause the user to see frequent buffering of clips.
- User experience of the presentation is compromised; greater latency and startup times affect the time needed to view the clip, and delivery of the clip requires more total bandwidth.
- A RealServer behind a firewall can only serve content to users who are also behind the firewall.
- If an encoder is behind a firewall and attempts to send content to your RealServer, the Server may not be able to receive its data.
- If there is a firewall between your RealServer and a receiver, RealServer may not be able to send data.

Communicating with Software Behind Firewalls

Information in this section applies to administrators of the Server who are interested in the nature of the connection between RealServer and the following RealSystem software:

Component	Your Control Over Connection Type
Clients (such as RealPlayer)	If your RealServer is placed correctly in relation to your firewall (if any), there is not much you can do to enhance the user experience if the client software is behind a restrictive firewall.
Encoders (such as RealProducer Plus) Receivers (such as RealServers) Proxies (such as RealProxy)	The information in these sections will provide useful background information for discussions you may have with administrators of these last three types of software.

Communicating with Clients

When no firewall exists between RealServer and the client (such as when they are both in the same internal network), the component software first tries to establish a two-way TCP control connection to RealServer. The Server uses this connection initially as a means of sending information to the client about the requested media, such as the name, length, and copyright of the clip. The client uses the connection to send commands to RealServer when features such as the Play and Stop buttons are activated.

Initial Connection Between RealServer and Client



After the initial connection is established, RealServer then establishes a data channel back to the client. The actual media is sent along this channel, which uses UDP.

Data Channel From RealServer to Client**How Clients Communicate with a RealServer from Behind a Firewall**

This section explains the logic used within the client software as it tries to contact your RealServer.

To optimize playback quality, clients are designed to automatically try different methods of connecting to RealServer to work through common firewall configurations.

The list below shows how the client software determines what protocol it will ask RealServer to use in sending the streamed media over the data channel.

1. The client attempts to open a control connection, using TCP. It uses port 554 for the RTSP protocol, or port 7070 for the PNA protocol.
 - If the firewall does not allow TCP on 554 (or port 7070), the client tries to establish a connection using HTTP on port 80. RealServer first tries to communicate by sending responses wrapped in the HTTP protocol. This is known as HTTP cloaking.
 - If HTTP cloaking doesn't work, the client attempts to download the media clip from RealServer using HTTP downloading.

By default, HTTP download is not allowed, because clients obtain the whole clip, rather than a streamed version. To enable HTTP download, add the clip's path to the HTTP Delivery list.
 - If the firewall permits the TCP connection, the client goes to Step 2.
2. Now that a TCP control connection has been established, the client attempts to set up the data channel.

If the request is for on-demand content, the client tries these methods:

- a. First, it tries UDP, in the range of port 6970 through 32000. (Earlier versions of RealPlayer used a smaller range. Consult the "Ports Used by RealPlayer" table.)
- b. If UDP is not allowed, it requests that the data be sent via TCP on port 554, using the established control channel.

If the request is for live content, the client tries three connection methods:

- a. First, it tries to use multicast. This is a specialized option not available on many networks. Multicast uses the UDP transport protocol and may use either the RTSP or PNA application-level protocol. Firewalls must be specially configured to allow multicast traffic.
- b. If multicast is not available, the client requests that the material be sent via UDP on ports 6970 through 6999.
- c. If UDP cannot pass through the firewall, the client requests delivery via TCP (also on port 554 or 7070).

Users can configure RealPlayer to always use a particular protocol and port as directed by their firewall administrator.

Additional Information

Refer to *RealPlayer Plus Manual* for instructions on setting preferences in the client. See

<http://service.real.com/help/library/index.html>.

Improving User Experience for Clients Behind a Firewall

If the client is behind a firewall, and is having difficulty accessing your content, there are steps you can take that may improve service.

Once you have determined that your RealServer is in the correct position in relation to your firewall (see “Best Firewall Configurations” on page 131), these steps will help:

- Make sure your RealServer is using 80 for the HTTP Port (see the instructions in “To ensure that HTTP traffic can get through to your RealServer:” on page 125).
 - Refer users or their firewall administrators to the RealNetworks firewall information page at **<http://service.real.com/firewall>**, where they can find information that explains how to make firewalls compatible with streaming media.
- To ensure that HTTP traffic can get through to your RealServer:
- RealPlayer includes an option to request that all streams be sent in HTTP format. If you do not have a Web server installed on the same computer or IP address as RealServer, you can receive these clients’ requests by setting the HTTP Port value to 80. If RealServer is installed on the same computer as a Web server, you have the following options:

- If you have one IP address, do not use 80 for HTTP Port. You must use different values, and these different values must be included in any RealServer links that begin with HTTP.
- If you have two IP addresses, use the IP Binding feature to instruct RealServer to use a particular IP address, on which it can listen for HTTP requests.

Communicating with Encoders

RealServer can communicate with encoders that are behind firewalls, as long as RealServer is located in its network's perimeter network.

When RealServer and the encoding software are on the same side of a firewall, there are no communication difficulties between them.

Different encoder versions use different protocols to connect to RealServer:

- Encoders developed for use with RealServer version 8 have the option to use UDP or TCP (you can choose which transport method they use in RealProducer Plus Preferences dialog box)
- Encoders developed for use with RealSystem version 6 use UDP
- Encoders developed for use with RealServer versions 3 through 5 use TCP

Additional Information

To see the port numbers used by the different encoder versions, refer to "Port Numbers Used by Encoders" on page 135.

As in communicating with clients, UDP is the preferred option for communicating with the Server.

UDP is a more robust protocol for real-time communications, and is therefore the preferred method for encoder-to-Server connections. The TCP protocol can be affected by network congestion and disrupt the live session. UDP is more resilient in periods of brief network congestion.

Encoding connections cannot be proxied. Therefore, if the encoder is behind a firewall, you must do one of the following (listed in order of preference):

- Move the encoding tools to the DMZ
- If you are using RealProducer Plus version 6.1, use TCP

- ▶ To use TCP for communicating with RealProducer Plus 6.1:
 1. In RealProducer Plus, click **Options>Preferences**.
 2. Select the **Live Broadcast** tab.
 3. Select **Connect to Server Using TCP**.
 4. Click **OK**.

Communicating with Receivers

By default, receivers and RealServers use UDP to communicate. An option is available for them to use TCP instead.

Splitting connections cannot be proxied. Therefore, if the receiver is behind a firewall, you must do one of the following (listed in order of preference):

- Move the receiver to the perimeter network (see “Locating RealServer Near the Firewall” earlier in this chapter)
- Change the transport protocol used for transmitter-to-receiver communication to TCP as described in Chapter 12.

Communicating with RealProxys

A RealProxy located behind a firewall is a common scenario. In this respect, a RealServer-to-RealProxy connection behaves like a RealServer-to-client connection, but only two connection types are available:

1. It tries to connect with RTSP with UDP for data transport.
2. If that fails (the firewall prohibits UDP connections), RealProxy tries RTSP with TCP for data transport.

Options for HTTP delivery, which other component software may use, are not available.

If the firewall prohibits TCP, RealProxy will not be able to proxy streams on behalf of clients.

Additional Information

Refer to *RealProxy Administration Guide* for information on configuring RealProxy. See

<http://service.real.com/help/library/index.html>.

Firewall Configurations (for Firewall Administrators)

Firewalls can be categorized into roughly six types. A particular firewall vendor may combine more than one type into a particular product. The type of firewall in use by your organization will affect the method that RealServer uses to stream content to clients.

- Application-level proxy
- Transparent proxy
- Packet filter
- Stateful packet filtering
- SOCKS
- Network address translation

The address that appears in the access log of the source RealServer depends on the client's type of firewall.

A firewall monitors every type of transmission between client software and the Internet, but this discussion looks only at the firewalls' effects on streaming media.

Application-Level Proxy Firewall

Application-level firewalls first determine if a requested connection between a computer on the internal network and one on the outside is permitted. If the connection is authorized, the firewall mimics the requesting software and sets up the necessary communication links between the two computers. As an intermediary, the firewall can monitor the communication between the two networks and suppress any unauthorized activity.

Because an application-level firewall acts as an intermediary between RealPlayer and RealServer, the firewall itself must know how to handle the RealPlayer protocols (RTSP and PNA).

The user must configure the client software to contact a proxy or firewall machine. (In RealPlayer, this setting is located under **Options>Preferences>Proxy**.)

The source RealServer's access log shows the IP address of the firewall machine, instead of the client's address.

Transparent Proxy Firewall

A network administrator configures the firewall to intercept requests for streaming media.

The source RealServer's access log shows the IP address of the firewall machine, instead of the client's address.

Packet Filter Firewall

Rather than impersonating an application, network-level firewalls examine the packets of information sent at the transport level to determine whether a particular packet should be blocked. Each packet is either forwarded or blocked based on a set of rules defined by the firewall administrator.

A common configuration for network-level-filtering firewalls is to allow all connections initiated by machines inside the firewall, and to restrict or prohibit all connections made by machines outside the firewall. For most programs, this works well since they usually only establish a single outbound TCP connection.

However, RealPlayer and RealServer maintain two simultaneous connections: a TCP connection for sending commands and a UDP connection to stream the actual media according to the instructions received via TCP. The TCP connection initiated by the Player for controlling the connection will work through a packet filter firewall. Since network-level filters block UDP as a matter of course, the UDP stream sent by the RealServer will be deflected off the firewall and never reach the Player that made the request.

RealServer's access log displays the address of the client.

Stateful Packet Filtering Firewall

A stateful packet filtering firewall monitors the communication between the client and the Internet to ensure that inbound packets are being sent at the request of a client inside the firewall. Similar to packet filters, it may include additional options that allow more sophisticated actions to be taken with individual packets.

These firewalls should be configured to permit RTSP and PNA traffic.

RealServer's access log displays the address of the client.

SOCKS Firewall

Only software with built-in SOCKS support, that must additionally be configured by the user, can send data through a SOCKS firewall; RealPlayer does not include SOCKS support.

In some cases, a user can install a Winsock.dll that supports SOCKS, and configure it to point to the SOCKS firewall.

Network Address Translation Firewall

A network address translation firewall converts the client's internal address to an external address before it forwards the client's requests to RealServer. Once it receives a request, RealServer will send its UDP packets directly to the firewall, rather than to the client, and the firewall may not know which client requested the packets.

Summary of Firewall Types and Characteristics

The table below summarizes the six most common firewall types and any special configuration information.

Streaming Media Over the Firewall Types

	Client configuration required?	IP address seen by the client	IP address seen by the Server (in access log)	Valid inside addresses required?	RTSP support required to get UDP?	RTSP support required to get TCP?
Application-level proxy	Yes	Firewall's address	Firewall's address	No *	Yes	Yes
Transparent proxy	No	Server	Firewall	No*	Yes	No**
Packet filter	No	Server	Client	Yes	No	No
Stateful packet filtering	No	Server	Client	Yes	No	No
SOCKS	Yes	Firewall	Firewall	No*	No***	No
Address translation	No	Server	Firewall	No*	Yes	No

* Usually requires compliance with RFC 1597 Address Allocation for Private Internets (<http://www.ietf.org/rfc/rfc1597.txt>)

** May require special configuration

*** Requires SOCKS version 5

Some firewalls are actually a mix of the firewall types described in the preceding section. For example, many packet filtering firewalls also allow network address translation, so the IP shown in the RealServer access log is the firewall's, rather than the client's.

Best Firewall Configurations

The firewall that provides the best experience for RealSystem software users is one that allows streaming media, by allowing a TCP control channel on port 554, and a UDP data channel on a range of ports.

- Several firewall vendors already include this type of streaming media support. View the RealNetworks firewall page at <http://service.real.com/firewall> to find a vendor.

- You can modify your existing firewall with the help of the free RealNetworks Firewall Administrator's Proxy kit.

The next best option is a firewall that allows a TCP control channel and a TCP data channel both on port 554. Your firewall administrator can easily make this change to the firewall. However, the quality of the connections will not be as good with this configuration.

Finally, nearly all firewalls allow HTTP traffic, which will work for RealServer-to-client connections. However, encoders and receivers will not be able to communicate with RealServer over HTTP.

Locating RealServer Near the Firewall

If your RealServer is behind a firewall streaming content to clients on the other side of the firewall, reconsider its location. A RealServer behind a firewall does not make much sense, for the following reasons: RealServer needs to open TCP connections based on client requests, and most firewalls permit TCP connections only when they are initiated inside the firewall. Also, RealServer needs to open UDP channels on a variety of ports. Here again, most firewalls permit few, if any, UDP connections.

Your content will be completely inaccessible to clients on the Internet if your RealServer is behind a firewall.

The solution is to move the firewall to a perimeter network, sometimes known as a De-Militarized Zone (DMZ). A perimeter network is outside the main internal network, but still secured by the firewall. Client requests for TCP and UDP connections do not pose the security risk here that they do when the RealServer is behind a firewall. Machines in the perimeter network can be set up with a different, more liberal set of security features than is suitable for those on the internal network.

Ports Used in Streaming and Unicasting

Information in these tables will help you decide which ports to open on your firewall. For more detailed information, especially if you do not want to explicitly open all the ports listed, refer to the documentation on the RealNetworks Web site, at <http://service.real.com/firewall>.

These tables do not cover use of port numbers in multicasting.

Port Numbers Used by RealServer

Normally, the client software chooses UDP for the data channel, and indicates a port number between 6970 and 6999 on which it will receive the data. RealServer receives the request on port 554 (if requested via RTSP) or port 7070 (if requested via PNA), and directs the data to the port number specified by the client.

If the client software chooses TCP for the data channel, RealServer uses the same port number for both the control channel and the data channel. If the clip was requested using RTSP, both channels will use port 554. If the clip was requested using PNA, both channels will use port 7070.

This table shows the typical values used by RealServer.

Ports Used by RealServer

Listen On or Send To	Port Number	Protocol	Purpose
Communicating with RealPlayer			
Listen on	554	TCP	Control channel for RTSP requests (data channel also, if TCP was requested)
Listen on	7070	TCP	Control channel for PNA requests (data channel also, if TCP was requested)
Listen on	8080	TCP	HTTP requests
Send to	6970-6999	UDP	Data channel (port numbers are not configurable)
Communicating with RealSystem Administrator			
Listen on	Admin Port	TCP	Connections to RealSystem Administrator
Listen on	9090	TCP	Java Monitor traffic
Communicating with Receivers			
Listen on	2030	TCP or UDP	Data channel for pull splitting requests
Send to	11001	TCP or UDP	Push receiver requests
Communicating with Encoders			
Listen on	4040	TCP	Control channel for RealProducer Plus version 6 and 6.1 connections
Listen on	6970-32000	UDP	Data channel for RealProducer Plus 6 and 6.1
Listen on	4040	TCP	Data channel for RealProducer Plus 6.1 connections, if TCP was selected
Listen on	5050	TCP	Control channel for pre-G2 encoder connections

(Table Page 1 of 2)

Ports Used by RealServer (continued)

Listen On or

Send To Port Number Protocol Purpose

Communicating with RealProxy

Listen on	3030	TCP or UDP	RealProxy live requests
Listen on	7802	TCP	RealProxy requests
Listen on	7878	TCP	RealProxy requests

(Table Page 2 of 2)

Port Numbers Used by Receivers

In addition to the values shown in the table, if receiver is also serving its own content (separate from splitting), it will use all the values in the “Ports Used by RealServer” table on page 133.

Ports Used by Receivers

Listen On or

Send To Port Number Protocol Purpose

Communicating with RealServer

Listen on	554	TCP	RTSP requests from RealPlayer
Send to	2030	TCP or UDP	Pull splitting requests
Listen on	11001	TCP or UDP	Push splitting requests

Port Numbers Used by RealProxy

Ports used by RealProxy are shown below.

Ports Used by RealProxy

Listen On or

Send To Port Number Protocol Purpose

Communicating with RealPlayer

Listen on	1090	TCP	PNA proxy requests
Listen on	1091	TCP	RTSP proxy requests
Send to	6970-6999	UDP	Data channel (port numbers are not configurable)

Communicating with RealServer

Send to	554	TCP	Control channel for RTSP requests to RealServer
---------	-----	-----	---

(Table Page 1 of 2)

Ports Used by RealProxy (continued)

Listen On or Send To	Port Number	Protocol	Purpose
Send to	3030	TCP or UDP	Data and control channel for pull splitting requests
Send to	7070	TCP	Control channel for PNA requests to RealServer
Send to	7802	TCP	Cache requests to RealServer
Send to	7878	TCP	Cache requests to RealServer
Communicating with RealSystem Administrator			
Send to	9090	TCP	Java Monitor traffic
Listen on	Admin Port	TCP	RealSystem Administrator

(Table Page 2 of 2)

Port Numbers Used by Encoders

In RealProducer Plus versions 6 and later, you can instruct RealProducer Plus to use TCP for the data connection. UDP is the preferred method, as it results in a better user experience, but TCP may be necessary if there is a firewall between the encoder and the Server. If you do opt to use TCP, port 4040 will be used for both the control channel and the data channel.

Ports Used by Encoders

Listen On or Send To	Port Number	Protocol	Purpose
RealProducer Plus versions 6 and later, communicating with RealServer			
Send to	4040	TCP	Control channel.
Send to	6970-32000	UDP	Data channel, if UDP is selected for the Server Connection. (The actual port number is not configurable)
Send to	4040	TCP	Data channel, if TCP is selected for the Server Connection.
RealProducer Plus versions 5 and earlier, communicating with RealServer			
Send to	5050	TCP	Control and data channel

Port Numbers Used by RealPlayer

In addition to the settings shown below, RealPlayer inherits proxy settings (if they exist) from the default browser. If they change in the browser, the changes are reflected in RealPlayer. Users can turn off this feature from the RealPlayer Preferences menu.

Ports Used by RealPlayer

Listen On or

Send To Port Number Protocol Purpose

RealPlayer versions 6 and later, communicating with RealServer or RealProxy

Send to	554	TCP	Control channel for RTSP requests. Data channel, if TCP was requested.
Send to	7070	TCP	Control channel for PNA requests. Data channel, if TCP was requested.
Send to	80	TCP	Control channel. Data channel, if HTTP cloaking or HTTP streaming is used.
Send to	8080	TCP	Control channel. Data channel, if HTTP cloaking or HTTP streaming is used.
Listen on	6970 - 32000	UDP	Data channel

RealPlayer versions 3 through 5, communicating with RealServer or RealProxy

Send to	554	TCP	Control channel for RTSP requests. Data channel, if TCP was requested.
Send to	7070	TCP	Control channel for RTSP requests. Data channel, if TCP was requested.
Send to	80	TCP	Control channel (and data channel, if HTTP cloaking or HTTP streaming is used)
Send to	8080	TCP	Control channel (and data channel, if HTTP cloaking or HTTP streaming is used)
Listen on	6970 - 6999	UDP	Data channel (not configurable)

Chapter 10

STREAMING ON-DEMAND PRESENTATIONS

This chapter describes how RealServer streams on-demand, or prerecorded, presentations.

Overview

RealServer is ready to stream content when you first install it, and will stream on-demand presentations and files that you place in the Content directory. When a user clicks a link to an on-demand presentation, RealServer streams the presentation from the beginning of the file. In live broadcasting, the streaming begins at a specified time, and anyone who clicks the link later than that misses the beginning.

Summary of On-Demand Streaming Versus Live Broadcasting

On-Demand Streaming	Live Broadcasting
Can access presentations anytime	Can only access presentations while they're in-progress.
Files are stored on disk	Presentations don't exist as files
Presentations always begin streaming at the beginning of the file	Everyone sees the same part of the presentation at the same time—latecomers join in the middle
Like a movie on videotape	Like a movie premiered on network television

When to Use Streaming

On-demand streaming is suitable for any content that is not time-sensitive. The following are examples of good uses for on-demand streaming:

- Archives of live broadcasts
- Information that is not time-sensitive
- SMIL presentations

Use live broadcasting for events that you want to broadcast as they are encoded. Refer to Chapter 11, “Unicasting Live Presentations”.

On-Demand Streaming Used with Other Features

This section describes the ways in which on-demand streaming works together with other features.

Live Delivery Methods: Unicasting, Simulated Live Broadcasting, Splitting, Multicasting, and On-Demand Streaming

Streaming and live broadcasting methods are mutually exclusive methods, with one exception: you can use the Simulated Live tool (**G2SLTA**) to broadcast on-demand files as if they were live. See “Creating a Live Source with G2SLTA” on page 48 for detailed information on using the **G2SLTA** tool.

Live Archiving and On-Demand Streaming

If you have used the live archiving feature to convert live streams to on-demand files, you can then create links to these individual files and deliver them via on-demand streaming.

You can also use the archived files to re-create a live presentation using the **G2SLTA** program.

RealProxy and On-Demand Streaming

RealServer is configured to automatically allow RealProxy to store all live and on-demand content streamed by your RealServer. To prevent certain on-demand files from being cached by RealProxy software, see the “To prevent RealProxy from caching material on your RealServer:” instructions on page 107.

Firewalls and On-Demand Streaming

As long as your RealServer is correctly located outside a firewall, it can deliver on-demand content to clients on the Internet. Refer to Chapter 9, “Firewalls and RealServer” for more information.

Access Control, Authentication, and On-Demand Streaming

Any access control or authentication rules you set up for your RealServer are automatically used when users request on-demand content.

Monitoring and On-Demand Streaming

You can view which presentations are being streamed at any time with the Java Monitor. Click the **Connections** tab or the **Files** tab to see which files are in use.

Logging and On-Demand Streaming

All presentations streamed from your RealServer, whether on-demand or live, will be listed in the access log.

Storing On-Demand Clips

On-demand files are made by content creators. The administrator of RealServer and the content creator may be the same person, but they are discussed here as two separate people. The content creator may encode into RealAudio or RealVideo files, or assemble RealPix presentations, or create any other type of file that RealServer can stream.

Place clips in the RealServer Content subdirectory.

If you do not want to use the Content directory, do one of the following:

- Place the files in a subdirectory of Content and include the subdirectory name in the link.

Additional Information

See “Storing Clips in a Subdirectory of the Content Directory” on page 79.

- Place the files in a different location (not a subdirectory of Content) and create another mount point to use for this content.

Additional Information

Refer to “Storing Clips in a Different Directory” on page 80.

Streaming On-Demand Clips

RealServer needs three things in order to stream on-demand clips:

1. The on-demand clips themselves. For more information on creating clips, see Chapter 4, “Sources of Content”.
2. Correct settings on RealServer. These are pre-configured, but read “RealServer Settings” if you want to verify what they are.

3. Links to the clips.

RealServer Settings

When RealServer is installed, it is configured to stream content found in the Content subdirectory of the main RealServer directory. Subdirectories of Content may also contain content. RealServer is ready to stream your on-demand content when you first install it.

RealServer uses these settings to stream on-demand files:

- **Main Mount Point**—The main mount point is named with a forward slash (/). To view this setting, click **Mount Points** under **General Setup**.
- **Base path**—the base path of the main mount point gives the location of the Content directory.
- **HTTP Port, PNA Port and RTSP Port**—these are the port numbers where RealServer expects to receive requests.

Linking to On-Demand Clips

A link to an on-demand file has the following format:

`http://address:HTTPPort/ramgen/path/file`

RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming. Always use <code>http</code> in Web pages.
<code>address</code>	Machine and domain name of RealServer. IP address may be substituted.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<code>ramgen</code>	Ram file generator mount point.
<code>path</code>	Optional; the virtual directory is any actual directory, relative to the base path of the mount point. If the file is located in the base path itself, omit <code>path</code> .
<code>filename</code>	The file name itself, including the extension.

For example, a link to a file named `video.rm`, if placed in the Content directory, would look like this:

`http://realserver.example.com:8080/ramgen/video.rm`

To create a link for it in a Web page, use the anchor tags:

```
<a href="http://realserver.example.com:8080/ramgen/video.rm">Click here to
watch the latest video.</a>
```

A link used in RealPlayer has a slightly different format:

`rtsp://address:RTSPPort/path/file`

For example,

`rtsp://realserver.example.com:554/video.rm`

Working with SureStream Clips

Because of differences in software, equipment, and data transmission speeds, users will view your content at different bandwidth volumes. When a client requests a clip, it sends its bandwidth capabilities to the RealServer. RealAudio and RealVideo files encoded with the RealSystem encoding tools (versions 6

and later) record media at different rates, and store them in a single file, called a SureStream file. A RealServer that receives a request for a media file from a client will note the client's bandwidth, locate the correct portion of the file, and will stream the highest portion of the stream that matches the request. In this way, visitors to your site will receive the highest possible quality transmission, the person who encodes the file need encode only once, and you the administrator need keep track of only one file. In addition, as available bandwidth can vary over time, RealServer switches between streams automatically.

If the file does not contain an encoded portion that matches the client's requested bandwidth, RealServer sends a message to the client indicating that no matching bandwidth is available.

Files Created with Previous Encoder Versions

Bandwidth negotiation of RealAudio and RealVideo was handled in previous versions of RealNetworks products by creating one file for each compression algorithm, and putting all the files in a directory whose name ended with `.rm`. Files were named according to the compression algorithm with which they were encoded.

If you still have these files, you don't need to re-encode them. RealServer reads the old directory structure and can perform the bandwidth negotiation automatically. Bandwidth negotiation is always active; only in those directories ending with `.rm` will RealServer perform old-style bandwidth negotiation.

All Other Data Types

Audio and video data types are the only types that contain multiple compression rates within one files. If you are streaming another data type, such as text, bandwidth negotiation is handled via a SMIL file. Instructions on doing this are available in *RealSystem Production Guide*.

Chapter 11

UNICASTING LIVE PRESENTATIONS

Concerts, presentations, speeches, can all be encoded and broadcast to clients almost instantaneously. Live presentations can be archived for later reference or later broadcast. For example, you can archive an event that happens in one time zone and then play it later for viewers in a later time zone with the **G2SLTA** tool.

Overview

In live broadcasting, a content creator sets up a device, such as RealProducer Plus or RealProducer Pro, to capture a real-time event, that will create the digital media for RealServer to stream. A Web page includes a link to the live event, and everyone who clicks that link sees or hears the same event at the same time.

Visitors who click a link to a live broadcast join the event as it happens, and everyone sees the same content at the same time.

Streaming live content is much the same as streaming static content, with the following differences:

- The encoder sends encoded material immediately to your RealServer. On-demand files are completely recorded or created and are then copied to RealServer.
- Everyone who clicks the link to a live stream sees the same material at the same time. On-demand files always begin playing at the beginning.
- Live streams don't exist as files. They're streamed while they're encoded, but they don't exist as files. The content creator gives the live broadcast a file name, and you use this in the link for the live broadcast.

Tip

If you don't want to broadcast live events yourself, RealBroadcast Network™ (RBN) provides full services

for encoding and broadcasting events to a few or a few thousand viewers. See <http://www.realnworks.com/rbn> for details.

When to Use Live Unicasting

The following are factors in deciding whether to use this feature:

- You want all users to play the same content at the same time.
- You have the equipment, or the content creator has the equipment, to encode a live event.
- The number of users whom you anticipate for a particular event is not enough to warrant a multicast event.
- Users who typically connect to your events are located nearby, and using a receiver is not necessary.

Live unicasting is typically used for audiences of light-to-medium volume live events. Splitting and multicasting may be more appropriate for events with a larger number of viewers.

Live unicasts are often used for such events as:

- live radio broadcasts
- executive speeches

Live Unicasting Used with Other Features

Live broadcasting works with all other RealServer features. There are a few special considerations for each feature, however.

On-Demand Streaming and Live Unicasting

Streaming and live broadcasting methods are mutually exclusive methods, with one exception: you can use the Simulated Live tool (G2SLTA) to broadcast on-demand files as if they were live. See “Creating a Live Source with G2SLTA” on page 48 for detailed information on using the G2SLTA tool.

Live Archiving and Live Unicasting

The live archiving feature can create on-demand files of any live broadcast. These files can be used for historical purposes or for later rebroadcast.

Simulated Live (G2SLTA) and Live Unicasting

The simulated live tool (**G2SLTA**) is a command-line tool that creates a live presentation from an on-demand file.

Splitting and Live Unicasting

Unicasting and splitting (both push and pull splitting) are often used together, without any special configuration. This happens when your RealServer is configured as a source RealServer, and there are links that allow clients to connect directly to your RealServer.

In the figure titled “Illustration of Splitting” on page 160, the RealServer in Japan is acting as a source to the receivers in Australia and North America, and is also unicasting to the three RealPlayers at the right.

Multicasting and Live Unicasting

Unicasting and back-channel multicasting are also often used together; links to back-channel multicasts do not require that the client be multicast-enabled, so if the client is not able to connect in multicast mode, it will be able to receive the stream via unicast instead.

Scalable multicasting includes a feature that enables a client to receive a broadcast via unicast if the special multicast presentation is not available for some reason.

RealProxy and Live Unicasting

RealProxy cannot cache live broadcasts, because there is no actual file to cache. But RealProxy includes an ability to “share” live streams among clients, and thus reduce the bandwidth required from a transmitter. They communicate through pull splitting; RealServer is pre-configured to act as a transmitter, and RealProxy is automatically set up to act as a receiver.

Firewalls and Live Unicasting

Standard protocols are used in delivering live broadcasts. These are covered in Chapter 9, “Firewalls and RealServer”.

Access Control, Authentication, and Live Unicasting

Before any client is allowed to receive any broadcast, RealServer checks the client’s IP address to see whether the client is allowed to receive a broadcast. If the address is acceptable, RealServer looks at the location of the file to see if it

is in a secure location. If so, RealServer challenges the user (or Player) for identification. Once the client passes the tests, RealServer connects the client to the live broadcast.

ISP Hosting and Live Unicasting

RealServer's ISP hosting feature can host on-demand content only; it does not host any live content.

Monitoring and Live Unicasting

All streams and connections to broadcasts currently in use can be viewed by using the Java Monitor in RealSystem Administrator.

Logging and Live Unicasting

All client connections to live broadcasts are recorded in the access log file. Any errors encountered by clients are recorded in the error log.

Unicasting Live Clips

Setting up RealServer to broadcast live events consists of three steps:

1. Configure RealServer.
2. Create the link to the unicast.
3. Start encoding the live event. This is described briefly in Chapter 4, "Sources of Content".

Configuring RealServer for Live Unicasting

RealServer is ready to stream live events when you first install it. This section describes the settings that RealServer uses in live unicasting.

Encoders

If the encoder is a RealSystem version 6 encoder, such as RealProducer Plus 6, RealServer uses the settings on the **Encoder** page (located by clicking **Broadcasting** in the left-hand frame of RealSystem Administrator):

- **Mount Point**—typically /encoder/. All links to live material will include this mount point.
- **Port**—the port number to which the encoder will send its live data. For RealSystem version 6 encoders, a typical value is 4040. If you change this

value, be sure to give the new port number to content creators so they can direct their encoded feeds to the right RealServer port.

- **Authentication**—Content creators using RealSystem version 6 encoders can have individual user names and passwords. If you will be requiring user names and passwords from encoder connections, select the name of the appropriate realm from the **Authentication** box. A typical realm for encoders is `EncoderRealm`. Select `None` if you do not want to require user names and passwords from encoders.

Additional Information

Realms and authentication are described in Chapter 15, “Authenticating RealServer Users”.

- **PNA Port and RTSP Port**—because links may include port numbers, make sure that the settings for **PNA Port** and **RTSP Port** are correct. View these settings by clicking **General Setup>Ports**.

Pre-G2 Encoders

View these settings by clicking **Broadcasting>Pre-G2 Encoder**.

- **Mount Point**—typically `/live/` for links to material created by encoding software that was released prior to RealSystem G2, such as RealEncoder. All links to live material will include this mount point.
- **Port**—the port number to which the encoder will send its live data. For encoders earlier than version 6, a typical value is 5050. If you change this value, be sure to give the new port number to content creators so that their encoders can still connect to RealServer.
- **Password**—encoders developed before RealSystem G2 are able to supply passwords, but no user name. The value for Password is established during setup. If you change the password using RealSystem Administrator, be sure to tell content creators what password to use. All pre-G2 encoders will use this same password.
- **PNA Port and RTSP Port**—because links may include port numbers, make sure that the settings for **PNA Port** and **RTSP Port** are correct. View these settings by clicking **General Setup>Ports**.

Creating the Link to the Live Unicast

Links to live events are similar to links for on-demand clips, with the addition of the encoder mount point.

Use the format of linking to an individual file, and use the live file mount point, usually `/encoder/` (if the material is arriving from a version 6 or later encoder).

- To link the live file from a Web page:

The link to a live file has the following format:

```
http://address:HTTPPort/ramgen/encoder/path/file
```

RealServer Live Content URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming. Always use <code>http</code> in Web pages.
<code>realserver.example.com</code>	Machine and domain name of RealServer. IP address may be substituted.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<code>ramgen</code>	Ram file generator mount point.
<code>encoder</code>	Live events encoded with version 6 and later software use <code>/encoder/</code> as their mount point. If the live event is created using a pre-G2 encoder, use the <code>/live/</code> mount point instead.
<code>path</code>	Optional; any actual directory, relative to the base path of the main mount point. If you typed any subdirectory in the source software (other than the <code>/encoder/</code> mount point), include it here.
<code>file</code>	The file name itself, including the extension.

For example, a link to a live event being encoded as `concert.rm` would look like this:

```
http://realserver.example.com:8080/ramgen/encoder/concert.rm
```

Links typed directly in RealPlayer or used in a Ram or SMIL file use the following format:

```
rtsp://address:RTSPPort/encoder/path/file
```

The format is nearly the same as the link used in the Web page; the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

Optional Live Unicasting Features

With live unicasting, the following optional features are available:

- Playing A “Please Stand By...” Message

Playing A “Please Stand By...” Message

If a live unicast is interrupted, you can still send information to clients, displaying a message that says “Currently experiencing technical difficulties” when a live broadcast is interrupted. This is possible by making a file that contains the message you want to display, and placing it in a subdirectory with the same name as the live mount point.

► To create a “Please Stand By...” Message:

1. Create an actual subdirectory with the same name as the live mount point (encoder), and place it under the main base path (usually Content).

If you are working with pre-G2 encoders, use live instead (or use both encoder and live).

Your directory structure will now look something like this:

```
RealServer
  Content
    encoder
```

2. Create a file, in the same format as your broadcast (such as RealAudio, RealVideo, or SMIL) that contains the error message you want to stream in place of the live file. You may want to include information about when the visitor should check back (keep in mind the different time zones which viewers may be in).
3. Place this new file in the encoder subdirectory.

If a live stream fails to arrive at RealServer, RealServer will search for an actual directory that matches the URL. In this case, it will find the subdirectory with the error file and will stream the error file.

Archiving Live Broadcasts

You can save (or *archive*) a live broadcast for historical purposes or for later playback. For information on playing saved files as if they were live, see “Creating a Live Source with G2SLTA” on page 48.

RealNetworks’ encoding products include an option to save a copy of a file while encoding. This setting is different from, and independent of the archiving feature in RealServer. Typically, there is more storage space on the RealServer system than there is on the content creator’s computer.

Choosing the Size of the Archived Files

For each live broadcast that you want to save, you can choose to create one large file that contains everything in the original broadcast or several small files. These small files can be based on length of recording or file size. For example, RealServer can archive a continuous live feed into files each containing thirty minutes of the broadcast, or can start a new archive file each time a certain size is met.

Live Archiving Options

Method of Archiving	Suggested Use
A single large file	<ul style="list-style-type: none"> • Corporate presentations • Concerts
Small files, based on elapsed time	<ul style="list-style-type: none"> • Ongoing news broadcasts • Event coverage
Small files, based on file size	<ul style="list-style-type: none"> • Ongoing events • Where disk space is a concern

Large Files

Large files are appropriate when you want to save an entire event in one file. If RealServer archives a live broadcast with the same destination path and file name as an existing file, RealServer automatically renames the file by appending a unique number to the end. For example, if RealServer encountered a file named `concert.rm` in the archive directory, it would rename it as `concert.rm.86400`. The new file gets the `concert.rm` name. The number that RealServer chooses is related to a timestamp; larger numbers indicate newer files. In this way, one directory can be used to store the latest version of a broadcast and the previous versions as well.

Reusing the same output file name can simplify Web page maintenance, because the links for a recurring event remain the same.

Small Files

Small files based on elapsed time are saved with the following method: as soon as the initial value indicated in the configuration file is reached, the archived file will be named *filename0.rm*. When the second archived file maximum size is reached, it is named *filename1.rm* where *filename* is the name of the live file stream.

File names for files based on size are named with the same method as for files archived according to elapsed time.

If RealServer tries to archive a stream for which an archived file already exists, it uses the same method as described in “Large Files”.

Temporary Files in the Archive Directory

Because several bit rates are present in SureStream files, RealServer creates several temporary files as it archives the streams. When the desired file time or size is reached, RealServer merges the temporary files into the final SureStream format.

The time required to merge the files is usually as long as or longer than the presentation itself. Thus a one-hour clip takes a little more than an hour to merge.

Warning

Do not stop RealServer before it merges the temporary files, or it will not create the archive file.

When to Use Live Archiving

Use live archiving when:

- the files will be large when archived, and there is not enough storage space on the encoding computer
- you want records of your live broadcasts for later reference

Live Archiving Used with Other Features

Live archiving works with all other RealServer live broadcasting features. There are a few special considerations for each feature, however.

On-Demand Streaming and Live Archiving

Although you can use the Simulated Live tool (**G2SLTA**) to broadcast on-demand files as if they were live, and then archive them, it does not make much sense to archive an existing file.

Splitting, Multicasting, and Live Archiving

Both transmitters and receivers can archive broadcasts.

Setting Up Live Archiving

When you set up live archiving, you create settings that apply to all live broadcasts performed by your RealServer, or indicate that only broadcasts associated with certain source paths will be archived. In both cases, you can specify any location for the resulting archived files.

The instructions given here will archive a live broadcast into a single large file. If the resulting file will be too large, or if you want to divide the archived file into smaller files automatically, see “Creating Small Files Limited by Size or Time” on page 155.

Note

These instructions describe only the steps required to set up this feature. For more options, see “Optional Live Archiving Features” on page 154.

► To set up live archiving:

1. In RealSystem Administrator, click **Broadcasting**. Click **Live Archiving**.
2. Select the asterisk (*) in the **Source Paths** list. This archives all live streams broadcast by this RealServer.

Additional Information

Instead of archiving all live streams, you can be selective. See “Archiving Only Certain Streams” on page 154.

3. From the **Archiving** list, select Enabled.
4. In the **Destination Path** box, name the virtual path where RealServer should store the files. Archived files will be created and stored here automatically. The default location for archived files is the Archive subdirectory of the RealServer Content directory.

5. Click **Apply**.

Disabling Live Archiving

To turn off live file archiving, select the virtual directory for which you want to turn off live archiving. Select Disabled from the **Archiving** list.

Linking to Archived Files

An archived file is a live broadcast that has been converted to an on-demand file. Links to archived files use the on-demand style; the default location is the Archive subdirectory of the Content directory, so include that in the URL.

The actual file name which the link will reference depends on the archiving method you used:

- If you archived the broadcast in a single large file, link to the single file.
- If you archived in small files, you will need to create a link for each small file.

Links in a Web page use this format:

`http://address:HTTPPort/ramgen/Archive/file`

RealServer Archived File URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming. Always use <code>http</code> in Web pages.
<code>address</code>	Machine and domain name of RealServer. IP address may be substituted.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<code>ramgen</code>	Ram file generator mount point.
<code>Archive</code>	The virtual directory of the archived files.
<code>file</code>	The file name itself, including the extension.

For example, a link to a single large file would look like:

`http://realserver.example.com:8080/ramgen/Archive/concert.rm`

If the same event were archived by file time, a link to an individual small file would look like:

<http://realserver.example.com:8080/ramgen/Archive/concert42.rm>

Optional Live Archiving Features

Live archiving has these additional features:

- Archiving only certain streams
- Creating small files limited by size or time
- Archiving to another disk drive
- Using bandwidth negotiation

Archiving Only Certain Streams

You can choose to archive a limited number of broadcasts, or to archive all but a few broadcasts.

► To archive only specific broadcasts:

1. In RealSystem Administrator, click **Broadcasting**. Click **Live Archiving**.
2. Select the asterisk (*) in the **Source Paths** list.
3. From the **Archiving** list, select Disabled.

This step disables archiving for all broadcasts. The next steps will turn on archiving for specific streams.

4. Click **Add New**.

A generic path name appears in the Edit Source Path box.

5. Type the name of the path or virtual path that you want to archive in the **Edit Source Path** box. This is the same text that the content creator will type in the encoder (excluding the file name).

6. Click **Edit**.

7. From the **Archiving** list, select Enabled.

8. List the directory where RealServer should store the files in the **Destination Path** box.

9. Repeat Step 4 through Step 8 for each location where files are being broadcast.

10. Click **Apply**.

- To archive most broadcasts, but prevent certain broadcasts from being archived:
 1. In RealSystem Administrator, click **Broadcasting**. Click **Live Archiving**.
 2. Select the asterisk (*) in the **Source Paths** list.
 3. From the **Archiving** list, select Enabled.

This step turns on archiving for all broadcasts. The next steps will disable archiving for specific streams.
 4. Click **Add New**.

A generic path name appears in the Edit Source Path box.
 5. In the **Edit Source Path** box, type the virtual path to which live files are being sent.
 6. Click **Edit**.
 7. From the **Archiving** list, select Disabled.
 8. Repeat Step 4 through Step 7 for each location where files are being broadcast.
 9. Click **Apply**.

Creating Small Files Limited by Size or Time

Limiting by file size is shown in “Choosing the Size of the Archived Files” on page 150.

- To limit the files by their size, type the maximum desired size (in megabytes) in the **Limit Archive Files By Size** box.
- To limit the size of the archived files by time, type in the appropriate box in the **Limit Archive Files By Time** boxes.

If you give values in both the **Limit Archive Files By Size** or **Limit Archive Files By Time** boxes, RealServer will use the first, or lower, limit.

To save entire broadcasts without limiting the file size (create a large file), omit values for both areas.

Archiving to Another Disk Drive

If the machine on which RealServer is installed does not have enough disk space for archiving, you can set up a mount point for another machine.

These instructions apply to archiving all streams to one location, but you can combine the instructions in “Archiving Only Certain Streams” with the

instructions in this section to direct individual streams to separate archive storage locations.

► To archive to another disk drive:

1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.
2. Click **Add New**.
A generic mount point name appears in the **Edit Mount Point** box.
3. Type the name of the new mount point, such as “FileStorage”.
4. Click **Edit**.
5. Type a description of this mount point in the **Description** box, such as “Archive Directory”.
6. In the **Base Path** box, type the complete path to the directory where you want RealServer to create and store archive files.
7. Click **Apply**.
8. Click **Broadcasting>Live Archiving**.
9. From the **Source Paths** list, select the directory of incoming live events that you want to archive. To archive all incoming broadcasts, select the asterisk (*).
10. In the **Archiving** list, select Enabled.
11. In the **Destination Path** box, type the new mount point you created in Step 11, such as /filestorage/.
12. Select any additional features for live archiving, and click **Apply**.
RealServer will now archive live broadcasts in the new location.

Using Bandwidth Negotiation

If RealServer 5-style bandwidth is in use, select 0n from the **Bandwidth Negotiation** list. When this option is set to 0n, and RealServer receives streams from encoders with the .rm extension, RealServer creates a directory named after the filename, including the extension. All streamed files are created in this subdirectory. For more information on version 5-style bandwidth negotiation, see “Files Created with Previous Encoder Versions” on page 142.

Note

If you are using bandwidth negotiation to create the files and the **Bandwidth Negotiation** option is set to Off, RealServer makes a choice as to which file it stores in the archive directory. It will store the first stream that arrives, as *file.rm* (not as a directory), and the other bandwidth-encoded files will not be available. Be sure to specify the *.rm* extension when setting up the encoder to encode the live stream.

Chapter 12

SPLITTING LIVE PRESENTATIONS

Splitting is a method of sending live broadcasts to other RealServers, rather than to clients. These other RealServers, configured as receivers, re-broadcast the streams to clients. By replicating streams close to users, clients receive high quality content, bandwidth usage is minimized, and audience size is maximized.

Overview

The RealServer where the live content originates, called the *transmitter*, makes its live broadcasts available to other RealServers, called *receivers*. Links on Web pages point to the receiver, rather than to the transmitter. When a user clicks the link, the receiver recognizes the special URL and relays the stream from the transmitter to the client.

As soon as the transmitter begins streaming a live source, it sends the broadcasts to all receivers. When a client requests a broadcast from the receiver, a connection has already been established between the receiver and the transmitter, and the broadcast is delivered to the client immediately.

Note

There is another type of splitting in which live events are only sent at the request of receivers. Read about this method in “Pull Splitting” on page 176.

For example, a concert from Japan can be broadcast over the Internet to RealServers in Australia and North America. Users in those cities connect to the closest RealServer, thereby getting better media quality and using less network bandwidth.

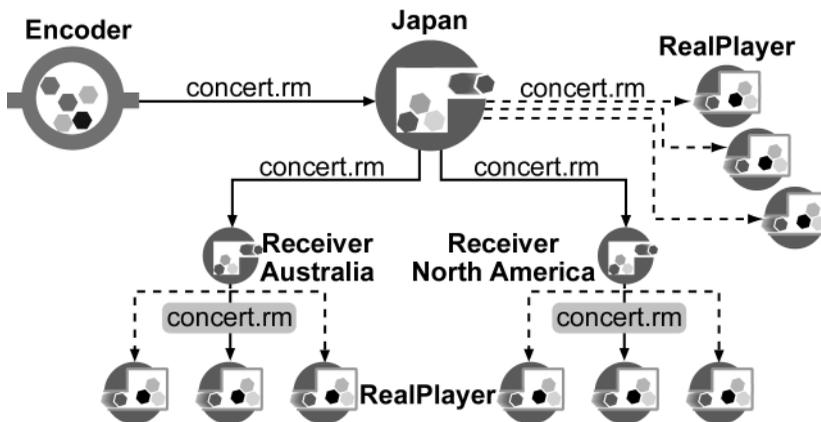
While serving content that originated on another computer, a RealServer—whether a transmitter or a receiver—can still stream its own content. And

because it's no longer serving to all the clients, it has more connections available for streaming its own content.

Example Splitting Scenario

Throughout this chapter, we'll use the example of a transmitter located in Japan that is broadcasting to receivers in Australia and North America.

Illustration of Splitting



Web pages listing the event have different links for different locations:

Sample Web Page, Linked to a Push Split Broadcast



Splitting Used with Other Features

This section describes the ways in which splitting works together with other features.

On-Demand Streaming and Splitting

Neither splitting method works with on-demand streaming; splitting only delivers live broadcasts. You can use **G2SLTA** to convert on-demand clips to live broadcasts. See “G2SLTA and Splitting” later in this section.

Live Unicasting and Splitting

Unicasting can happen automatically from a transmitter or a receiver—when you create a link that points directly to the stream on the receiver, you’re using unicasting. In the diagram titled “Illustration of Splitting”, the three clients shown in the upper right are receiving unicasts from the transmitter in Japan.

Multicasting and Splitting

There are two ways you can use these features together to make the most efficient use of bandwidth.

- For multicast communications between transmitter and receiver, use multicast for the protocol. Select this option as shown in “Step 1: Setting Up the Transmitter” on page 165.
- For multicast communications between receiver and clients, see “Splitting and Multicasting” on page 187 for information and an illustration.

Live Archiving and Splitting

Both transmitters and receivers can archive live streams.

G2SLTA and Splitting

The transmitter can broadcast either a live event from an encoder or an on-demand clip broadcast by **G2SLTA**. Using **G2SLTA** can be a good way to test your receiver configurations before you broadcast the real event.

RealProxy and Splitting

RealProxy cannot cache live broadcasts, because there is no actual file to cache. But RealProxy includes an ability to “share” live streams among clients, and thus reduce the bandwidth required from a transmitter. They communicate through pull splitting; RealServer is pre-configured to act as a transmitter, and RealProxy is automatically set up to act as a receiver.

Firewalls and Splitting

In their default configuration, the transmitter and the receiver use UDP for fast, efficient distribution, but some firewalls are not configured to permit UDP traffic to traverse network segments. If this is the case with your firewall, modify your firewall to allow UDP traffic between the transmitter and the receiver. You can customize the port numbers that the transmitter and receiver use to communicate, and match them to port numbers on your firewall.

The TCP option from the **Protocol** list is not recommended unless the network is highly reliable and resend requests are unlikely. The TCP transport should not be used in lieu of properly enabling UDP routes between transmitters and receivers.

Additional Information

Firewalls, and the best arrangements of transmitters and receivers, are described in “Communicating with Receivers” on page 127.

Access Control and Splitting

If you use the access control feature to block access to your transmitter, receivers can still obtain split broadcasts, but they cannot make resend requests, regardless of whether the **Honor Resend Requests** feature is enabled on the transmitter. Furthermore, receivers using the pull splitting feature will not receive any content at all.

Authentication and Splitting

If you are sending a stream to a RealServer that is acting as a receiver, you must put copies of all the databases that store authentication information on the receiver. This distributes the authentication load.

Calculating Latency and Bandwidth

Stream Acquisition Latency

To determine how long it will take a receiver to acquire a live split stream from a transmitter, consider the following:

- The degree of network latency that exists between the transmitter and receiver

- The configurable interval used to send session announcements.

The receiver does not begin processing the incoming data until it receives the session description from the transmitter. If the session announcement-forwarding interval is set to say, 30 seconds, the receiver will require a maximum of 30 seconds before being able to begin processing live broadcast packets. This time delay does not include any network latency that may exist between the transmitter and the receiver.

As soon as the live distribution stream arrives at the receiver, the receiver constructs a local buffer before streaming data to the connected clients. This introduced receiver latency occurs only if a client connects to the receiver before the session announcement information arrives.

In cases where a pull splitting request is initiated by the receiver, the receiver stores only a 3-second buffer before sending the stream to the client.

Bandwidth Considerations

Calculating the bandwidth used by a particular presentation depends on whether the broadcast is single-rate or SureStream.

Single-rate broadcasts consume bandwidth according to:

- The encoded bit rate of the live stream.
- The percentage of overhead (approximately 10 percent) for session announcement information and TCP/IP headers.
- The percentage of overhead for the configurable error correction rate.

Additional Information

Instructions on setting the error correction rate can be found in “Optional Transmitter Features” on page 167.

For example, a single-rate 100-Kbps stream with an error correction rate set to 10 percent will consume approximately 120 Kbps of bandwidth.

Determining the encoding bit rates for SureStream broadcasts is a bit more complex. As a rule, the bandwidth consumed by a SureStream broadcast equals the combined bit rates of all the audiences plus the overhead percentages mentioned a few paragraphs earlier.

Compatibility with Earlier Versions of RealServer

Because of the numerous changes to the splitting feature, both the transmitter and the receiver must be running RealServer 8.

Links use a different format than in previous versions of RealServer.

Earlier versions of RealProxy can still receive pull splitting from RealServer.

If you used splitting in earlier versions of RealServer, you'll want to note two areas of changes.

One is a change in terminology; what used to be called *push splitting* is now called simply *splitting*. Pull splitting is called out as a separate case, as shown in "Pull Splitting" on page 176. A *source* is now called a *transmitter*, and a *splitter* is now called a *receiver*.

Technical changes consist of the following differences:

- The transmitter and receiver can now communicate via multicast, thus saving bandwidth. For more information on the multicast option, see "Step 1: Setting Up the Transmitter" on page 165.
- Splitting is now a "connectionless" broadcast method; it only requires a one-way connection from transmitter to receiver. Unless the receiver is configured to request resends, the transmitter doesn't hear from the receiver. For more information on requesting resends, see "Optional Receiver Features" on page 168.
- Data that was previously transmitted via the "back-channel" of the two-way communication is now sent in the data stream using the Error Correction and Metadata options. For more information on these options, see "Optional Transmitter Features" on page 167.
- Splitting now uses a password and security method to render data unusable to any RealServer that does not have the correct password. For additional information, see "Step 1: Setting Up the Transmitter" on page 165.
- There are now only two pages in RealSystem Administrator for setting up this feature.
- All variables used in configuration files are completely different than in earlier versions. For more information, see Appendix C, "Configuration File Contents".

Setting Up Splitting

There are three main steps to setting up splitting; the settings you use on the transmitter must match those used on the receiver.

1. Configure the transmitter. Use the steps in “Step 1: Setting Up the Transmitter” on page 165.
2. Configure the receiver. Follow the steps in “Step 2: Setting Up the Receiver” on page 167.
3. Create the links. Use the format described in “Step 3: Linking to Split Content” on page 169.

Step 1: Setting Up the Transmitter

When you install RealServer, the setup program automatically supplies some information to the splitting feature, based on information it detects on your system. Assuming these settings are correct, you need configure only a few areas, as described in the steps below. To determine whether the supplied values are correct, look at “Optional Transmitter Features” on page 167 for information.

► To configure the transmitter:

1. In RealSystem Administrator, click **Splitting**. Click **Transmitter**.
2. In the **Source Name** box, type a name to use in the links to the split broadcast. Use only letters and numbers; do not use spaces. For the examples in this chapter, we'll use Japan.
3. In the **Broadcast Receivers** area, click **Add New**.
A generic receiver name appears in the Edit Receiver Name box.
4. In the **Edit Receiver Name** box, type a name or description for the receiver RealServer to which this live content will be broadcast.
5. Click **Edit**.
6. In the **Receiver IP Address or Hostname** box, type the address of a receiver that is allowed to receive this content.
7. From the **Transport** list, select the type of transportation method to use in transmitter-to-receiver communications. You must choose the same transportation setting on both the transmitter and the receiver. The options, listed in order of recommended use, are:

- **UDP/Unicast**—The option most commonly selected.
- **UDP/Multicast**—The most bandwidth efficient option, this requires a multicast-enabled network between the transmitter and the receiver. If you select this option, you must also type an appropriate class D multicast address in the **Receiver IP or Hostname** box, and type a number in the **Multicast Time to Live (TTL)** box. For a list of possible TTL values, refer to the “Time to Live (TTL) Values” table on page 197.
- **TCP**—Use this option if you are splitting over a very reliable network connection. This choice will produce a connection that can be broken, disturbing the broadcast; it also takes more overhead.

Note

If you select the TCP transport option, you should also use an additional method. Refer to “Broadcasting Over Multiple Network Paths” on page 171.

8. Either accept the supplied values in the **Port Range** box, or type your own. This is the set of ports on receivers to which this broadcast will be sent. If the default ports will not work with your system, be sure to specify a range of at least 20 ports. Be sure that the ports are not in use for any other purpose.
9. To secure these transmissions, select Basic from the **Security Type** list, and type a password in the **Password** box. The receiver will pass this word to verify the identity of the transmitter as it receives broadcasts. To turn off security, choose None from the **Security Type** list.
10. Accept the settings for all the other values, or use the information in “Optional Transmitter Features” on page 167.
11. Click **Apply**.
12. Repeat the previous steps to add more receivers.
13. The person who will be setting up the receiver will need to know the values you chose in these steps, so that she can make the same selections (or enter the correct values in RealSystem Administrator). Specifically, the receiver administrator needs to know:

- | | |
|------------------------------|-------------------------------------|
| • Source Name | • Port Range |
| • Transmitter Address | • Security Type and Password |
| • Transport | • Receiver IP or Hostname |

Optional Transmitter Features

This section describes the additional settings that you may customize for your transmitter.

- **Error Correction Rate**—The percent of corrective packets sent. A higher number results in more reliable delivery of data, but consumes more network bandwidth. The default value for use over the Internet is 20. If you're using splitting within a local-area network, you can set this value to a lower value or to zero, since you are less likely to need error correction.
- **Metadata Transmit Rate**—The frequency (in seconds) at which special packets are sent from the transmitter to the receiver. These special packets describe the stream being split, and are required for the receiver to process the incoming stream.

A lower number may result in lower startup latency on a new receiver, but consumes more network bandwidth. Use a number from 1 to 60. See “Calculating Latency and Bandwidth” on page 162 for more information.

- **Honor Resend Requests**—To improve quality of service, you can choose to have receivers re-request packets that arrived in poor quality or not at all. Select **Yes** to turn on this feature. However, this option requires a back-channel from the receiver to the transmitter, produces a small increase in bandwidth to support the resend request traffic.

Step 2: Setting Up the Receiver

When you install RealServer, the setup program automatically supplies some information to the splitting feature, based on information it detects on your system. Assuming these settings are correct, you need configure only a few areas, as described in the steps below. To determine whether the supplied values are correct, look at “Optional Receiver Features” on page 168 for information.

Using the information you received from the administrator of the transmitter (see Step 13 on page 166), use the steps below to set up the receiver.

- ▶ To configure the receiver:
 1. In RealSystem Administrator, click **Splitting**. Click **Receiver**.
 2. In the **Mount Point** box, confirm the name of the mount point. This is the name you will use in the links.
 3. In the **Broadcast Transmitters** area, click **Add New**.

A generic transmitter name appears in the Edit Transmitter Name box.

4. In the **Edit Transmitter Name** box, type a name for the transmitter (or set of transmitters) where the live broadcast originates.
5. In the **Transmitter Address Space** box, type the host name, IP address, or IP address range of the transmitter. To indicate a range of addresses, type the netmask in the **Transmitter Netmask**.
6. In the **Protocol** box, select the same option that's in use on the transmitter.

If you selected udp/multicast, you must also add the **Multicast Address** field with the class D multicast address.

Note

If you are using udp/multicast, the address you type in the **Multicast Address** field must match the value in the **Receiver IP or Hostname** box on the transmitter.

7. Accept the settings for all the other values, or use the information in “Optional Receiver Features” in the next section.
8. Click **Apply**.

Optional Receiver Features

This section describes the additional settings that you may customize for your receiver.

- **Resend Requests**—To re-request packets that did not arrive at the receiver, select Yes. This option provides greater quality, but may take up more network bandwidth. It can also add latency as the receiver waits for the resent data to arrive.

Note

If you chose Yes for **Resend Request**, the administrator of the transmitter will need to set **Honor Resend Requests** to Yes. If she does not make this change, the transmitter will not re-transmit any lost data even though your receiver is requesting it.

- **Port Range**—Choose the same option that's in use on the transmitter.

- **Security Type and Password**—Choose the same options that are in use on the transmitter.

Step 3: Linking to Split Content

This section describes the format of links to split content.

- To create the Web page URL:

The link to the split content looks like this:

```
http://Receiver:HTTPPort/ramgen/broadcast/TransmitterSourceName/encoder/
path/file
```

Notice that the first part of the link refers to settings on the receiver, and the second part refers to settings on the transmitter.

URL Components in Web Page

Component	Meaning
Receiver Information	
http	The protocol used to initiate streaming.
<i>Receiver</i>	The receiver's host name or IP address.
<i>HTTPPort</i>	Optional; include only if the port setting has been changed from its default value of 8080.
ramgen	Required when you link in a Web page.
broadcast	The mount point used on the receiver, usually /broadcast/.
Transmitter Information	
<i>TransmitterSourceName</i>	The transmitter's Source Name value. If you are using backup transmitters (see "Using Backup Transmitters" on page 172), use an asterisk (*).
encoder	The live mount point.
<i>path</i>	Optional. The virtual path (if any) defined by the source software.
<i>file</i>	The name of the presentation, including the extension.

- To create the direct link URL:

The link that you would type directly in the Open Location dialog box of RealPlayer has the following format. The format is nearly the same as the link used in the Web page: the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

```
rtsp://Receiver:RTSPPort/broadcast/TransmitterSourceName/encoder/path/file
```

Example Links

Consider the example shown at the beginning of this chapter, in which a transmitter in Japan sends its broadcasts to receivers in Australia and North America.

Note

The direct link to the Japan RealServer uses a regular live broadcast link, rather than the splitting format used in the other links.

This example shows the text you would place in a Web page.

...we hope you enjoy the concert! Choose the link nearest you:

```
<a href="http://Japan.example.com.jp:8080/ramgen/encoder/concert.rm">
Japan<a>
<a href="http://Australia.example.com.au:8080/ramgen/broadcast/Japan
/encoder/concert.rm">Australia</a>
<a href="http://NorthAmerica.example.com:8080/ramgen/broadcast/Japan
/encoder/concert.rm">North America</a>
```

The same links, when typed directly in the Open Location dialog box of RealPlayer, would have the following formats:

```
rtsp://Australia.example.com.au:554/broadcast/Japan/encoder/concert.rm
rtsp://NorthAmerica.example.com:554/broadcast/Japan/encoder/concert.rm
```

Optional Configurations

This section describes some ways in which you can use splitting to create more sophisticated delivery of live broadcasts.

- Broadcasting Over Multiple Network Paths
- Using Receivers as Transmitters
- Using Backup Transmitters
- Chaining Receivers
- Pull Splitting

Broadcasting Over Multiple Network Paths

For broadcasting over sophisticated networks, you can combine RealServer splitting features to provide even more robust delivery of data between transmitter and receiver.

By setting up multiple delivery methods for each broadcast, data is sent over different network paths; the receiver re-assembles packets in order, regardless of the transport. For instance, if a packet is late or missing from the data sent via UDP/Unicast, the receiver may get the right packet from those arriving via UDP/Multicast.

► To broadcast over multiple network paths:

1. Use the instructions in “Step 1: Setting Up the Transmitter” on page 165 to set up a rule for this broadcast.
 - In the **Edit Transmitter Name** box, you might identify that this is one of three delivery methods, with a name such as JapanUDP.
 - From the **Protocol** list, select UDP/Unicast.
2. Create a second rule that uses multicast.
 - In the **Edit Transmitter Name** box, you might identify that this is one of three delivery methods, with a name such as “Japan Concert (Multicast)”.
 - From the **Protocol** list, select Multicast.

Remember, for multicast you must also add the multicast address to the **Transmitter Address** box.
3. On the receiver, create matching Broadcast Transmitter names. Again, give them similar names to those in the previous steps. Match the protocols to the transmitter settings.

Using Receivers as Transmitters

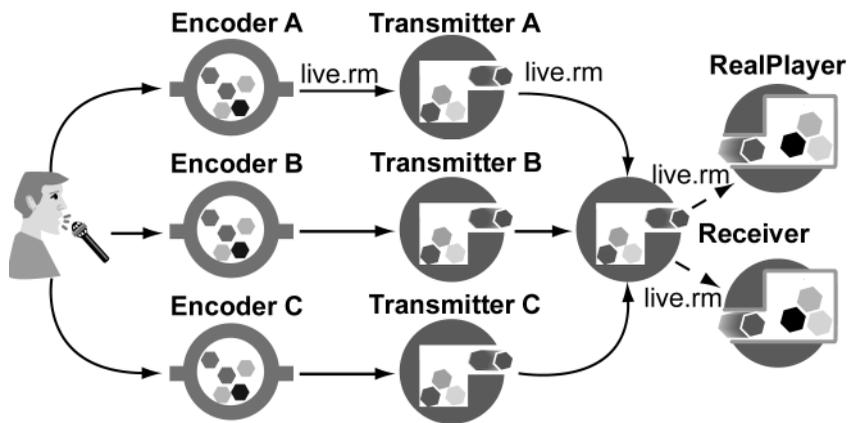
While serving as a receiver for material originating on a transmitter, a receiver can also serve its own content, using the standard broadcasting and on-demand streaming methods. To set up a receiver to also act as a transmitter, first set up the receiver portion, using the steps in “Step 2: Setting Up the Receiver” on page 167. Then set it up as a transmitter, using the instructions in “Step 1: Setting Up the Transmitter” on page 165.

You may also be interested in the daisy-chaining feature, in which each receiver acts as both receiver and transmitter for the same material. Refer to “Chaining Receivers” on page 173 for information.

Using Backup Transmitters

If you are broadcasting a live event that is being served from several transmitters, you can create a special link so that if one transmitter becomes unavailable, clients will still be able to connect the event using the single link. The link to this type of broadcast uses an asterisk (*) instead of the source transmitter’s name; this has the added advantage of obscuring the source of the broadcast, since the link does not include the name of the source. Should the transmitter become unavailable, the receiver will automatically choose the next available transmitter for new connections.

Backup Transmitters



For example, a client connects to the Receiver, which receives the broadcast from Transmitter A. If Transmitter A goes down, the client receives an error message. Meanwhile, the Receiver switches to Transmitter B. The user can re-click the link in the Web page or click the Play button in RealPlayer, and will receive live.rm again.

Using Backup Transmitters versus Redundant Sources

The backup transmitters feature provides parallel stream sources to a given receiver. The redundant sources feature delivers parallel sources to a transmitter. Use can use both methods at the same time for the greatest

reliability. For information about redundant sources, see “Working with Redundant Sources” on page 61.

► To set up backup transmitters:

1. Add the receiver information to each transmitter’s **Broadcast Receivers** list.
2. Configure the receiver to get broadcasts from each transmitter, by adding all transmitters to the receiver’s **Broadcast Transmitters** list.
3. Create the special URL for this broadcast: instead of typing the *TransmitterHostName* in the URL, type an asterisk (*).

Linking to Backup Transmitters

To create the link that will allow the stream to come from multiple RealServers, use the same format as for splitting (refer to “Step 3: Linking to Split Content” on page 169), but substitute an asterisk (*) for the *TransmitterHostName* value.

The following uses the example in the illustration above:

```
<a href="http://receiver.example.com:8080/ramgen/broadcast/*/live.rm">
```

Within RealPlayer, this link appears as the following:

```
rtsp://receiver.example.com:554/broadcast/*/live.rm
```

Tip

You can use the asterisk in an URL even when backup transmitters are not in use. This link format allows you to change the name of the RealServer shown in *TransmitterHostName* without having to also update all the links that reference it.

Chaining Receivers

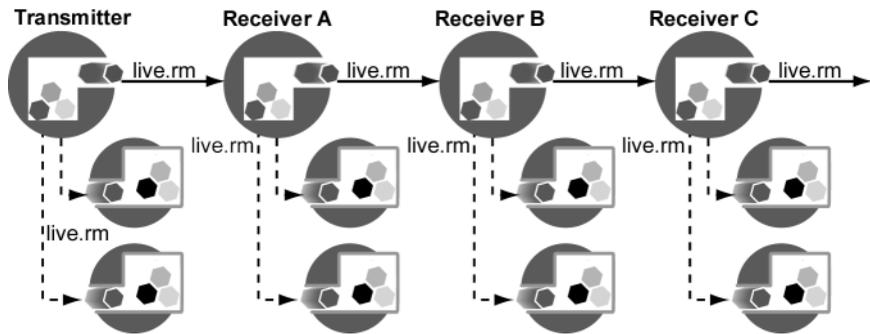
A receiver can act as a transmitter for another receiver. Clients connecting to the second receiver receive the broadcasts originating at the transmitter.

In the illustration below, a stream that originates at the transmitter is passed to Receiver A, then to Receiver B, and finally to Receiver C. A client can receive the live stream from any receiver.

Advantages to this method include the option to use different settings at each connection for error correction, metadata, and passwords.

Tip

If your network is multicast-enabled, consider using UDP/Multicast to transmit the stream instead of using chaining. Multicast has the same reliability provides greater reliability and scalability than chaining.

Daisy-Chain Splitting

In addition to being configured as a receiver, each receiver in the chain (except the last one) is also configured as a transmitter.

The configuration on each receiver points to the previous receiver in the chain. Each RealServer only knows about the RealServer on either side in the chain; it doesn't know about the entire chain.

Setting Up Chain Splitting

To set up this feature, you configure each RealServer in the chain, and then create the appropriate links.

► To set up chained splitting:

1. Configure the first RealServer:
 - a. Set up this RealServer as a transmitter.
 - b. Add Receiver A to the **Broadcast Receivers** list.
 - c. From the **Relay Live Broadcasts** list, select No.
2. Configure Receiver A:
 - a. Configure this RealServer as a receiver, and add the first Transmitter to the **Broadcast Transmitters** list.
 - b. From the **Relay Live Broadcasts** list, select Yes.

- c. Configure this RealServer as a transmitter, and add Receiver B to the **Broadcast Receivers** list.
 - d. In the **Source Path** box on the Transmitter page, change the default value from /encoder/ to /broadcast/ (to match the transmitter).
3. Configure Receiver B:
 - a. Configure this RealServer as a receiver, and add Receiver A to the **Broadcast Transmitters** list.
 - b. From the **Relay Live Broadcasts** list, select Yes.
 - c. Configure this RealServer as a transmitter, and add Receiver C to the **Broadcast Receivers** list.
 - d. In the **Source Path** box on the Transmitter page, change the default value from /encoder/ to /broadcast/ (to match the transmitter).
 4. Configure Receiver C:
 - a. Configure this RealServer as a receiver, and add Receiver B to the **Broadcast Transmitters** list.
- To create the links for chained push splitting:

The link for each receiver reference the receiver and the previous receiver in the chain, as shown in the following table. Notice that each link starts with the name of the RealServer that appears to be hosting the content, and also references the transmitter where the material originates.

Example Links for Web Pages

RealServer	URL Used for Receiver
Transmitter	http:// <i>HostName:port</i> /ramgen/encoder/live.rm
Receiver A	http:// <i>ReceiverA:port</i> /ramgen/broadcast/Transmitter/encoder/live.rm
Receiver B	http:// <i>ReceiverB:port</i> /ramgen/broadcast/Transmitter/encoder/live.rm
Receiver C	http:// <i>ReceiverC:port</i> /ramgen/broadcast/Transmitter/encoder/live.rm

A link typed within a Player, Ram file, or SMIL file would appear the same, but would use rtsp for the protocol and would omit ramgen.

Pull Splitting

In this type of splitting, the transmitter doesn't transmit any streams to the receiver until the first client makes a request.

In contrast to the constant communication in regular splitting, the connection between the transmitter and receiver in pull splitting stays quiet until a client makes a request. When the receiver gets a request for the live broadcast, it requests a pull splitting session. The transmitter RealServer sends the broadcast to the receiver, which in turn sends it to the client.

Using Push Splitting and Pull Splitting Together

You can combine these methods to make the best use of your bandwidth. For example, divide your use of splitting methods according to time zones. Use push splitting to send an event to receivers in the same or nearby time zones, and make pull splitting links available to users in time zones on the other side of the world, where potential viewers are likely to be asleep.

Setting Up the Transmitter for Pull Splitting

When you install RealServer, the setup program automatically supplies some information to the splitting feature, based on information it detects on your system. Assuming these settings are correct, you need configure only a few areas, as described in the steps below.

- ▶ To configure the transmitter for pull splitting:
 1. In RealSystem Administrator, click **Splitting**. Click **Transmitter**.
 2. In the **Pull Splitting Sources** area, click **Add New**.
A generic receiver name appears in the Edit Source Name box.
 3. In the **Edit Source Name** box, type a name for the stream.
 4. Click **Edit**.
 5. In the **Source Path** box, type a name of a path through which live streams are being sent, typically /encoder or /redundant.
 6. Type a password in the **Password** box. The receiver will use this word to verify the identity of the transmitter as it receives broadcasts. To turn off this feature, choose None from the **Security** list.

7. In the **Listen Port** box, type a port number on which this transmitter should listen for pull splitting requests. If you leave this field blank, RealServer uses the value 2030. Type here to override.
8. Click **Apply**.
9. The administrator of the receiver will need to know the values you chose for the following settings:
 - **Transport**
 - **Security Type and Password**
 - **Listen Port**
 -

Setting Up the Receiver for Pull Splitting

When you install RealServer, the setup program automatically supplies some information to the splitting feature, based on information it detects on your system.

► To set up the receiver for pull splitting:

1. In RealSystem Administrator, click **Splitting**. Click **Receiver**.
2. In the **Mount Point** box, confirm the name of the mount point. This is the name you will use in the links.
3. In the **Broadcast Transmitters** area, click **Add New**.
A generic transmitter name appears in the Edit Transmitter Name box.
4. In the **Edit Transmitter Name** box, type a name for the transmitter.
5. In the **Transmitter Address Space** box, type the host name, IP address, or IP address range of the transmitter. To indicate a range of addresses, type the netmask in the **Transmitter Netmask**.
6. In the **Port Ranges** boxes, type the range of ports that will receive the transmission. Be sure that the ports are not in use for any other purpose. If the default ports will not work with your system, be sure to specify a range of at least 20 ports.
7. From the **Enable Pull Splitting Requests** list, select Yes.
8. In the **Pull Splitting Source Path** box, type a value to identify this pull splitting stream in the URL. In the examples in this chapter, we'll use /pull/.
9. Accept the settings for all the other values, or use the information in "Optional Pull Splitting Receiver Features" in the next section.

10. Click **Apply**.**Optional Pull Splitting Receiver Features**

This section describes the additional settings that you may customize for your receiver.

- **Error Correction Rate**—The amount of corrective packets sent. The number shown here is the percent of error-correcting packets sent. A higher number results in better quality, but consumes more network bandwidth.
- **Metadata Transmit Rate**—The rate at which packets describing the session are sent. A higher number results in better quality, higher startup latency on the receiver, but consumes less network bandwidth. Use a number from 1 to 60. The default value is 30, which sends the metadata once every 30 seconds, adding 30 seconds of latency to the transmission.
- **Use TCP for Pull Splitting Backchannel**—Use this option if you are splitting over a very reliable network connection. This choice will produce a connection that can be broken, disturbing the broadcast; it also takes more overhead.

Linking to Pull Split Content

This section describes the format of the link to pull split broadcasts.

- To link to pull split content from a Web page:

The link to the split content looks like this:

```
http://Receiver:HTTPPort/ramgen/broadcast/SourcePath/Transmitter:ListenPort/encoder/path/file
```

Notice that the first part of the link refers to settings on the receiver, and the second part refers to settings on the transmitter.

Pull Splitting URL Components for Web Page

Component	Meaning
Receiver Information	
http	The protocol used to initiate streaming.
<i>Receiver</i>	Host name or IP address of the receiver.
<i>HTTPPort</i>	Optional; include only if the port setting has been changed from its default value of 8080.
ramgen	Used to create the link shown in “To create the direct link URL for pull splitting.”

(Table Page 1 of 2)

Pull Splitting URL Components for Web Page (continued)

Component	Meaning
<code>broadcast</code>	The receiver's mount point, usually <code>/broadcast/</code> .
<code>SourcePath</code>	The value in the Pull Splitting Source Path box.
Transmitter Information	
<code>Transmitter</code>	Host name or IP address of the transmitter.
<code>ListenPort</code>	The transmitter's Listen Port value. Default default value is 2030.
<code>encoder</code>	The live mount point.
<code>path</code>	Optional. The virtual path (if any) defined by the source software.
<code>filename</code>	Name of the file being split.

(Table Page 2 of 2)

- To create the direct link URL for pull splitting:

The link to the split content, as created by the transmitter or as typed directly in the Open Location dialog box of RealPlayer, has the following format. The format is nearly the same as the link used in the Web page: the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

```
rtsp://Receiver:RTSPPort/broadcast/SourcePath/Transmitter:PullListenPort/encoder/file
```

Example Pull Splitting Link

Consider the example shown at the beginning of this chapter, in which a transmitter in Japan sends its broadcasts to receivers in Australia and North America.

Note that the direct link to the Japan RealServer uses a regular live broadcast link (rather than the special pull splitting format).

The links used in the Web page have the following format:

...we hope you enjoy the concert! Choose the link nearest you:

```
<a href="http://Japan.example.com.jp:8080/ramgen/encoder/concert.rm">Japan</a>
```

```
<a href="http://Australia.example.com.au:8080/ramgen/broadcast/pull
```

`/Japan.example.com.jp:2030/encoder/concert.rm">Australia`

**`<a href="http://NorthAmerica.example.com:8080/ramgen/broadcast/pull
/Japan.example.com.jp:2030/encoder/concert.rm">North America`**

Links typed in RealPlayer, or used in SMIL or Ram files, have the following format:

`rtsp://Japan.example.com.jp:554/encoder/concert.rm`

`rtsp://Australia.example.com.au:554/broadcast/pull/Japan.example.com.jp:2030
/encoderconcert.rm`

`rtsp://NorthAmerica.example.com:554/broadcast/pull/Japan.example.com.jp:2030
/encoder/concert.rm`

Chapter 13

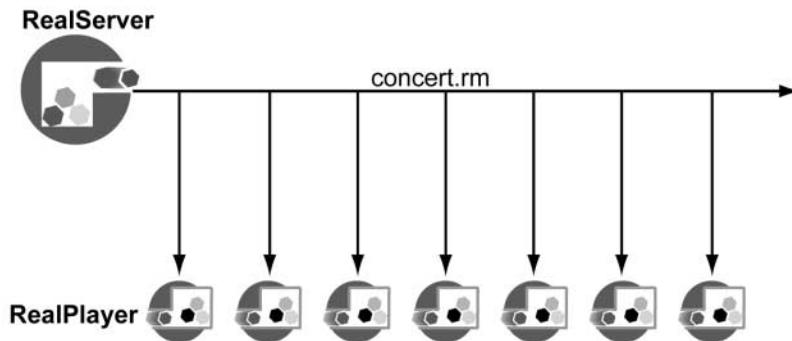
MULTICASTING LIVE PRESENTATIONS

Multicasting is another way of reducing the number of streams in use. It requires a specially configured network.

Overview

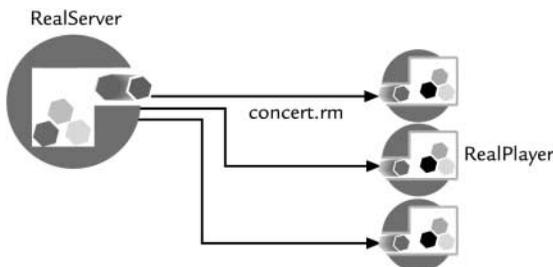
Multicasting is a way of sending a single live stream to multiple clients, rather than sending a stream to every single client. Clients connect to the stream, rather than to the RealServer.

Multicasting



In contrast, regular unicasting transmission sends a stream to each client that requests it:

Unicasting



To take advantage of multicasting, both RealServer and clients, as well as the routers, switches, and other devices between them, must be multicast-enabled. For this reason, multicasting is mostly used with intranets where network devices can be configured for multicasts. However, multicast delivery can be done over the Internet where intermediary network devices have been multicast-enabled.

The live material being multicast can be a live event encoded by an encoder such as RealProducer Plus, or it can be a prerecorded live event which is broadcast by **G2SLTA**. Only live or simulated live content can be delivered with multicasting. Refer to Chapter 11, “Unicasting Live Presentations” for information on configuring the live source.

When to Use Multicasting

The following are factors in deciding whether to use multicasting:

- You want to conserve bandwidth
- You know that most or all of the clients who will be connecting to your broadcasts are multicast-enabled
- You know how to set up the network for multicasting, or can work with another administrator who does

RealServer Multicasting Methods

RealServer includes two methods of multicasting: back-channel multicast, and scalable multicast. You can use both methods at once.

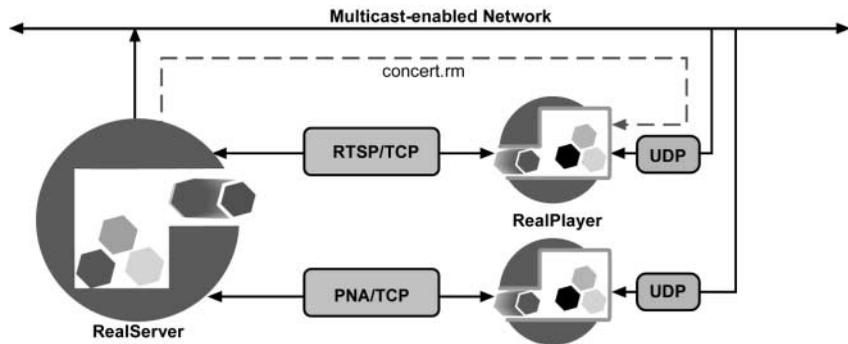
Back-Channel Multicasting

Back-channel multicast maintains an accounting control channel between the client and RealServer. RealServer uses this channel to provide information about the presentation and to query the client for a user name and password, if authentication is in use. The client uses the control channel to send password information and commands such as “play” and “stop”. With this information, RealServer can track how many clients are viewing a presentation. Monitoring tools such as the Java Monitor in RealSystem Administrator show client activity.

Back-channel multicast can be accessed using the RTSP or PNA protocols. In this type of multicast, authenticated material, client statistics, and quality of

service information can be sent because the exchange between the client and RealServer is bi-directional.

Back-Channel Multicasting



RTSP Multicast

This method of multicasting uses the RTSP protocol to send control information over a TCP channel. RealServer maintains a control connection for each client. The data channel is multicast to all clients using RDT, RealNetworks data transport.

RTSP multicast provides these features:

- **Authentication**—user name and password for secure content are sent using the RealServer authentication feature.
- **Connection statistics**—RealServer can receive client connection information.
- **SureStream**—these multiply-encoded files are supported. However, clients cannot shift between rates in the clips during playback.

Note

RTSP multicasting works only with RealSystem 6 clients.

PNA Multicast

PNA multicast uses the PNA protocol over a TCP connection to exchange information between the client and RealServer.

PNA multicast is used when transmitting to pre-G2 clients. RealServer maintains a control connection for each client. The data channel is multicast to all clients.

PNA multicast provides these features:

- **Authentication**—user name and password for secure content are sent using the RealServer authentication feature.
- **Connection statistics**—RealServer can receive client connection information.

SureStream is not supported in PNA multicast.

Scalable Multicasting

Unlike back-channel multicasting, scalable multicasting does not use a control channel; thus it takes up far less bandwidth and administrative overhead and system resources on RealServer. Monitoring tools such as Java Monitor will not track client activity. Client statistics can be sent, but only at the end of the multicast or when the user stops the presentation or the RealPlayer.

Scalable multicasting enables you to transmit to an unlimited number of clients because the transmission from the RealServer is completely one-way; there is no connection from each client to RealServer at all. All data is multicast on the network once. Each client connected to this multicast receives all data packets.

It is thus suitable for situations that would otherwise consume much bandwidth.

Scalable multicasting uses a different URL format than either RTSP multicast or PNA multicast.

Note

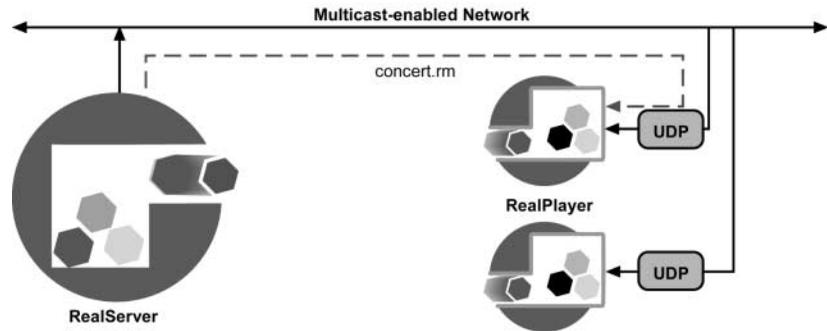
This method supports version 6 and later clients only; those clients version 5 and older will not receive any presentations, and will receive an error message instead.

Scalable multicast provides these features:

- **Scalability**—RealServer can support any number of clients.
- **Authentication**—the SDP file, indicated in the special URL format, is authenticated, but the actual content is not.

- **Connection statistics**—RealServer or a Web server can receive client connection information.
- **SureStream**—these multiply-encoded files are supported. However, clients cannot shift between rates in the clips during playback.

Scalable Multicasting



Users connect to scalable multicasts by clicking a link to a Session Description Protocol (SDP) file. This file is automatically generated by RealServer when a user clicks the scalable multicast link.

Choosing the Method of Multicasting

It's a good idea to enable back-channel multicast, because it applies to all streams broadcast by your Server. All client software is pre-configured to automatically try to connect in multicast mode if possible, so enabling back-channel multicast ensures that those clients that can use multicast will do so, leaving more bandwidth available for other clients.

Scalable multicast makes sense when you are broadcasting a high-bandwidth presentation, or if a large number of multicast-enabled clients will be viewing the presentation.

The following table summarizes the benefits of each multicast method.

Feature	Multicast Methods		
	Back-Channel Multicast		Scalable Multicast
	RTSP	PNA	
Control channel maintained with RealServer	•	•	
Authentication of users	•	•	•
Client statistics	•	•	•
Minimal RealServer resource use			•
RTP-enabled			•
SureStream support (without shifting capabilities)	•		•
Requires special URL format			•

The following table shows the multicasting methods as they apply to clients.

Feature	Multicast Methods		
	Back-Channel Multicast		Scalable Multicast
	RTSP	PNA	
RealSystem version 6 clients only	•	•	•
Older clients		•	
RealSystem 6 clients or any RTP-enabled clients			•

You can only take advantage of multicasting (either type) on networks that are multicast-enabled. If you are not certain whether your network is set up for multicasting, contact your network administrator.

Multicasting Used with Other Features

This section describes the ways in which multicasting works together with other features.

On-Demand Streaming and Multicasting

Multicasting only broadcasts live events; on-demand files cannot be multicast.

Live Unicasting and Multicasting

Both types of multicasting allow clients to switch to unicasting if the multicast becomes unavailable for some reason. In back-channel multicasting, this happens automatically; for scalable multicasts, you must set up this feature explicitly. Refer to “Using Unicasting as a Backup Method” on page 205 for instructions.

Live Archiving and Multicasting

As with all live broadcasts, you can configure RealServer to automatically create on-demand archived files of live multicasts.

Simulated Live (G2SLTA) and Multicasting

A multicast uses a live encoded event as its source; this can be an in-process encoded event or it can be a simulated live event created by **G2SLTA** using on-demand clips.

Splitting and Multicasting

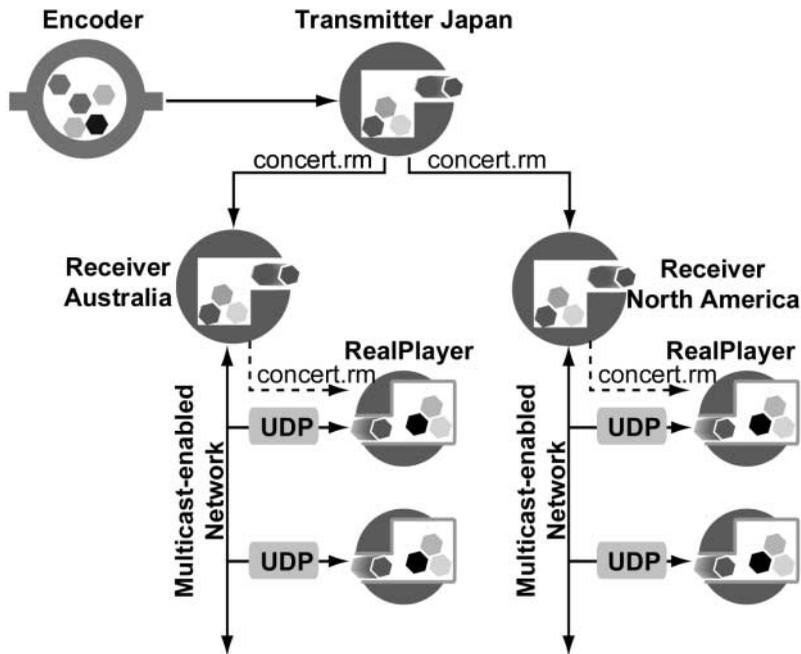
Use receivers to send data across the Internet to intranet sites, wide area networks, or local area networks, and then configure the receivers to multicast the streams to clients within the intranet. Clients will receive a multicast from the receiver.

Receivers cannot receive the source material using multicast, but when they re-transmit the material to clients, they can use multicast.

Additional Information

Read Chapter 12, “Splitting Live Presentations” for information on setting up splitting. Push splitting and pull splitting can be used with both back-channel and scalable multicasting.

Combining Splitting and Multicasting



There are four stages to configuring a combination of splitting and multicasting:

1. Configure the source RealServer for splitting.
2. Set up the receivers.
3. Configure the receivers for multicasting.
4. Create multicast URLs for clients in the intranet.

RealProxy and Multicasting

Depending on how the network is configured and the streams are listed in RealServer, clients whose requests are forwarded by a RealProxy may receive different results.

RealProxy cannot join a multicast. Instead, it will try to receive the multicast using pull splitting. If pull splitting is enabled on the source RealServer, the RealProxy will use that broadcast, instead of connecting to the multicast. The client will receive the broadcast in unicast mode. If there is a multicast-enabled

network between the RealProxy and the client, the RealProxy can be configured to re-send its pull split stream via multicast instead.

Type of Broadcast Received by Clients Using RealProxy

Is RealServer Configured to Split the Content?	Is RealProxy Configured to Multicast?	Result
Yes	Yes	Client receives the pull split unicast, not the multicast. However, RealProxy can be configured to multicast the pull split stream it receives.
	No	Client receives pull split unicast, rather than multicast.
No	Either Yes or No	Client receives multicast (RealProxy uses passthrough mode).

Additional Information

Refer to *RealProxy Administration Guide* for information on configuring RealProxy. See

<http://service.real.com/help/library/index.html>.

Firewalls and Multicasting

Multicasts usually take place within an intranet, where broadcasts are not travelling outside a firewall. If a multicast is occurring through a firewall, the firewall must be specially configured to allow multicast traffic.

Access Control, Authentication, and Multicasting

As with all delivery methods, RealServer verifies that the client requesting a broadcast is allowed to receive it before proceeding to send the multicast.

In scalable multicasting, the user connects to the multicast via an automatically created SDP file. If you include the authentication mount point in the link (*/secure/*), RealServer will verify the user's identity when the user clicks the link to the SDP file.

However, if the user saves the SDP file, she will not go through the authentication process if she later uses the saved SDP file to connect to the multicast.

Monitoring and Multicasting

In back-channel multicasts, you can see the clients that are connected, just as with regular unicasts.

Clients who are viewing a presentation via scalable multicast will not appear in the Java Monitor.

Reporting and Multicasting

Material served via back-channel multicast appears in the access log just like unicast material. The access log shows which method was used to transmit the stream.

Scalable multicasts can be identified in the access log by their mount point in the GET statement. If RealServer is configured for requesting client statistics (see “Controlling Client Statistics” on page 206), the log file will also contain statistics for each client.

Additional Resources

RealNetworks’ implementation of multicasting is based on open industry standards. You may be interested in the following resources:

General Information about Multicasting

- **Addresses available for multicast use**—“Assigned Numbers,” RFC 1700, available at <http://www.ietf.org/rfc/rfc1700.txt>.

Scalable Multicasting

- **RTP**—“RTP: A Transport Protocol for Real-Time Applications,” RFC 1889, available at <http://www.ietf.org/rfc/rfc1889.txt>.
- **RTP**—“RTP Profile for Audio and Video Conferences with Minimal Control,” RFC 1890, available at <http://www.ietf.org/rfc/rfc1890.txt>.
- **SDP (Session Description Protocol)**—“SDP: Session Description Protocol,” RFC 2327, available at <http://www.ietf.org/rfc/rfc2327.txt>.
- **SAP (Session Announcement Protocol)**—“Session Announcement Protocol: Version 2,” available at <http://search.ietf.org/internet-drafts/draft-ietf-mmusic-sap-v2-00.txt>.

Setting Up Both Types of Multicasting

Before you set up either type of multicasting, you need to do two things:

- Configure the network for multicasting.
- Select the addresses you'll use for your multicasts.

Once you have the network configured and have determined which addresses you'll be using, the next steps are:

1. Configure RealServer.
2. Create the link to the multicast.
3. Start encoding the live event. This is described briefly in Chapter 4, "Sources of Content".

Setting Up the Network for Multicasting

Before setting up RealServer, verify the following items with your network administrator:

- Routers and all equipment in your network are multicast-enabled.
- The system running RealServer is correctly configured for multicast support.

In addition to network settings, for clients to take full advantage of multicast transmissions, they must be configured to request multicast transmission of live material. Consult the client software's user guide for information on configuring the client.

As noted earlier, both RealServer and clients, as well as the routers and all other network infrastructure between them, must be multicast-enabled in order for you to distribute presentations using the multicast features. This section describes only what is required to enable RealServer for multicast broadcasting.

Allocating Addresses and Ports in RealServer

There are two factors to take into account when establishing the addresses and port numbers that RealServer will use for multicasting:

- Select addresses from a legal range of available addresses. Valid ranges are between 224.0.0.0 and 239.255.255.255. The network administrator should know which multicast addresses are available on the intranet. On

the Internet, certain ranges such as the addresses between 224.0.0.0 and 224.0.0.255 are reserved for other uses; see RFC 1700, “Assigned Numbers” for a complete list of restricted addresses.

- You must select enough addresses for the type of file you are multicasting. See “Determining the Number of Required Addresses and Ports” for information on selecting the appropriate number. For SureStream clips, you’ll need to know how many bit rates are included in each file that you are multicasting, and set aside the appropriate number.
- If your multicast streams are referenced in SMIL files, you will need to provide addresses for each stream.

Although the information in this document will help you calculate the number of addresses and ports you’ll need for scalable multicasting, you’ll still need to consult with your network administrator regarding the actual IP addresses you’ll use.

Determining the Number of Required Addresses and Ports

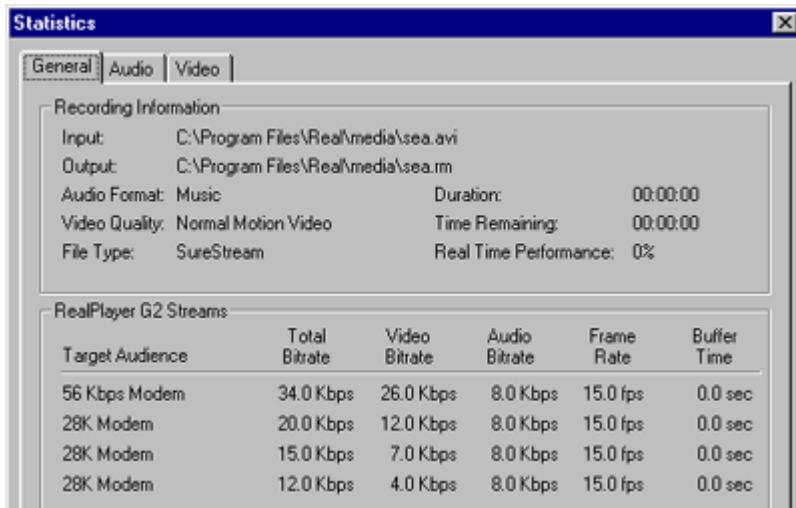
For each clip that you are transmitting via multicast, you must calculate the number of addresses you’ll need. The number of addresses is based on the number of bit rates in the SureStream clip.

In scalable multicasting, you’ll also need to set aside port numbers; these are based on the number of streams per bit rate.

For single rate RealVideo files, figuring the number of addresses and port numbers is relatively simple. SureStream clips are more complex, as they can contain several bit rates, each with its own number of streams.

If another person is supplying the file to be multicast, that person will need to tell you how many bit rates are in the file. For scalable multicasts, you will also need to know how many streams per bit rate are present.

You can get these numbers yourself if you are encoding the file; look in the encoding software to learn the number of bit rates and streams. For example, in RealProducer Plus, you would click **View>Statistics** to see the number of bit rates and streams being encoded.

RealProducer Plus Recording Statistics

Once you know the number of bit rates and streams in the file, refer to “Calculating Addresses for Back-Channel Multicasts” or “Calculating Addresses and Ports for Scalable Multicasts”.

If you have no way of knowing how many bit rates and streams are in the file, you’ll have to guess. A safe number is six; the maximum number of bit rates that would be present in a single SureStream file is 14, yet files prepared for multicasts are likely to include only the higher encoding rates. A non-SureStream file would have at most one bit rate and two streams.

Calculating Addresses for Back-Channel Multicasts

If you are preparing to transmit a file via back-channel multicast, you only need to know the number of bit rates in the file.

Addresses Needed for Back-Channel Multicasts

Bit Rates	Addresses
1	1
2	2
3	3
...	...
n bit rates	n

Calculating Addresses and Ports for Scalable Multicasts

Figuring the number of addresses and port numbers for scalable multicasts depends on several factors.

An audio presentation consumes one stream; a video presentation uses two streams (one for audio and one for video). For highest quality, and to match the scalable multicast RTP specification, RealServer uses one address for each stream.

RealServer includes a feature that enables you to send the audio and video streams on a single address. Use this feature if you want to conserve the number of addresses you're using. Use the dual address method if you know that clients viewing the presentation may be using a low bandwidth connection and the clients are able to select a single stream, such as just the audio portion of your multicast.

Each address must use a certain range of port numbers. The numbers you choose can begin with any number, but the first port number must be an even number, and you must use a consecutive range. (RTP is used to send the data; the RTP standard requires this format.)

In the table below, n represents the number of bit rates in the file that you'll be multicasting.

Addresses and Ports Needed for Scalable Multicast s

Bit Rates	Streams per Bit Rate	Reuse Address=False		Reuse Address=True	
		Addresses	Ports	Addresses	Ports
1	1 (audio only)	1	2	1	2
1	2 (audio and video)	2	4	1	4
2	1 (audio only)	2	4	2	4
2	2 (audio and video)	4	8	2	8
3	1 (audio only)	3	6	3	6
3	2 (audio and video)	6	12	3	12
...
n bit rates	1 (audio only)	n	$2n$	n	$2n$
n bit rates	2 (audio and video)	$2n$	$2n$	n	$4n$

Another way of calculating the number of ports needed is as follows:

- When **Reuse Address** is No, the number of ports is $2 \times \text{addresses}$.
- When **Reuse Address** is Yes, the number of ports is $2 \times \text{addresses} \times \text{streams}$.

Publicizing Your Multicasts

You can publicize your multicasts to anyone running a program that listens for the Session Announcement Protocol (SAP). These applications, such as SDR and ICAST Guide, display a list of all multicasts currently playing. RealServer creates the SAP file automatically. Programs that listen for SAP announcements will show the title, author, and copyright information encoded into the files you are multicasting.

This feature is optional; you do not need to configure it in order to use multicasting.

► To set up RealServer to create SAP files:

1. In RealSystem Administrator, click **Multicasting**. Click **SAP**.
2. In the **Host IP Address** box, type the IP address of this RealServer. The SAP announcement will include this address.
3. From the **Enable SAP Service** list, select Yes. This instructs RealServer to create and send SAP files. The default value is No.
4. From the **Listen to SAP** list, select Yes. The default value is Yes.

Tip

This option enables RealServer to collect in-use multicast addresses. RealServer consults this list when selecting a multicast address from the user-supplied address range, thus ensuring that it selects unique addresses that are not in use elsewhere on your network.

5. Click **Apply**.
6. Instruct the type of multicasting you are using to include SAP information:

For back-channel multicasts, use the following steps:

- a. Click **Multicasting**. Click **Back-Channel**.
- b. From the **Enable SAP** list, select Yes.
- c. Click **Apply**.

For scalable multicasts, use the following steps:

- a. Click **Multicasting**. Click **Scalable**.
- b. Select the **Channel** whose multicasts you want to announce.

- c. From the **Enable SAP** list, select Yes.
- d. Click **Apply**.

Multicasting with Multiple Network Interface Cards

If your machine has multiple network interface cards (NICs) and you want to ensure that RealServer always uses a particular NIC for multicasts, use your operating system to set a default address. In Windows NT, set the Bindings feature. In UNIX, use the **route** command to associate the multicast route with the appropriate NIC.

Multicasts do not use the settings in the IP Bindings list (described in “Distributed Licensing” on page 109).

Setting Up Back-Channel Multicasting

Follow the instructions below to set up back-channel multicasting. After you set it up, you will need to create the links that point to your multicasted events.

Configuring RealServer for Back-Channel Multicasting

Instructions in this section describe how to set up RealServer for back-channel multicasting.

Note

These instructions describe only the steps required to set up this feature. For more options, see “Optional Back-Channel Multicasting Features” on page 199.

- To set up back-channel multicasting:
 1. In RealSystem Administrator, click **Multicasting**. Click **Back-Channel**.
 2. Select Yes from the **Enable Multicast** to turn on this feature.
 3. In the **PNA Port** box, type the port number to which RealServer will direct its PNA multicast streams. The value in this box refers to the client’s port number. A recommended value is 7070.
 4. In the **RTSP Port** box, type the port number to which RealServer will direct its RTSP multicast streams. The value in this box refers to the client’s port number. A recommended value is 554.

5. Specify the range of addresses to which you want to multicast streams by filling in the **IP Address Range** box. RealServer uses the first available address in this range.

Additional Information

Refer to “Calculating Addresses for Back-Channel Multicasts” on page 193 to determine the exact number of addresses you’ll need.

6. Indicate how far multicast packets can travel over a network by typing a value in the **Time to Live** box. Each time a multicast data packet passes through a multicast-enabled router, its Time to Live is decreased by 1. When the value is decremented to 0, the router discards the data packet. The value for **Time to Live** can range from 0 to 255. The larger the Time to Live, the greater the distance a data packet can travel. The default value of 16 is enough to keep multicast packets within a typical internal network.

Time to Live (TTL) Values

TTL Value	Packet Range
0	Local host
1	Local network (subnet)
32	Site
64	Region
128	Continent
255	World

7. To allow missing packets to be resent to clients that request them, select True from the **Resend** list. This setting is optional. It adds some overhead to the traffic on your network; however, clients receive better quality multicasts.
8. Check to see that the **Client Access List** is using the correct values for list number 100 (this allows all clients to connect in multicast mode where possible):
 - **Client IP Address**—should contain the default value Any
 - **Client Netmask**—should be blank (blank is the default value)

Additional Information

You can customize this list to reflect the addresses of specific users or ranges of users; see “Listing Ranges of Authorized Clients” on page 199.

9. Click **Apply**.

Linking to Back-Channel Multicasts

Links to both RTSP and PNA multicast are identical to links for live unicast transmissions. This is convenient, because a single link can serve both multicast-enabled clients and unicast-only clients.

Most clients on a multicast-enabled network are configured to request material via multicast first, as this makes the most efficient use of bandwidth on your network.

- To link the live back-channel multicast from a Web page:

Just as in links to other live content, the link to a back-channel multicast has the following format:

`http://address:HTTPPort/ramgen/encoder/path/file`

RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming. Always use <code>http</code> in Web pages.
<code>address</code>	Machine and domain name, or IP address, of RealServer.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. Its default value is 8080.
<code>ramgen</code>	Ram file generator mount point.
<code>encoder</code>	Version 6 and later encoders use <code>/encoder/</code> as their mount point. If the live event is using a pre-G2 encoder, use the <code>/live/</code> mount point instead.
<code>path</code>	Optional; the virtual directory is any actual directory, relative to the base path of the mount point. If the file is located in the base path itself, omit <code>path</code> .
<code>filename</code>	The file name itself, including the extension.

Links typed directly in RealPlayer or used in a Ram or SMIL file use the following format:

`rtsp://address:RTSPPort/encoder/path/file`

The format is nearly the same as the link used in the Web page: the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

Optional Back-Channel Multicasting Features

The following features are available for back-channel multicast:

- Sending SAP Information with Your Multicasts
- Listing Ranges of Authorized Clients
- Requiring Multicasting Access Rather Than Unicasting

Sending SAP Information with Your Multicasts

Session Announcement Protocol (SAP) information can be sent over the multicast-enabled network to announce your scalable multicast. See “Publicizing Your Multicasts” on page 195 for instructions on configuring this option.

Listing Ranges of Authorized Clients

Clients whose addresses are listed in this section will use back-channel multicast to receive their clips.

Note

Access Control rules are enacted before User List rules. A client that is excluded by Access Control will not be able to connect to any multicasts, regardless of the rules you create here. (IP Access Control is described in “Limiting Access with the IP Address” on page 214.)

1. In the Client Access List area of the back-channel multicast page, click **Add New**. A generic rule number appears in the Client Access List and in the Edit Client Access List Number box.
2. In the **Edit Client Access List Number** box, type a new number or accept the default number.
3. In the **Client IP Address** box, type the address of the client that will be accessing the multicast presentation. To indicate a range of addresses,

type the netmask in the **Client Netmask** box, or leave the box blank to refer to a specific address.

4. Repeat Step 1 through Step 3 for each set of client addresses that will be allowed to view your presentations in multicast mode.
5. Click **Apply**.

Requiring Multicasting Access Rather Than Unicasting

You can require that clients connect to your broadcast in multicast mode, and make unicast unavailable.

Note

A client may be unable to connect in multicast if it is not configured to use multicast transports, or if its network is not multicast-enabled.

Normally, clients try to receive a broadcast in multicast mode, but if that is not available, the clients will use unicast mode instead. By requiring multicast, you can control bandwidth on your network.

- To require that all clients connect in multicast mode:
 1. In RealSystem Administrator, click **Multicasting**. Click **Back-Channel**.
 1. Set **Multicast Delivery Only** to Yes.
 2. Click **Apply**.

Clients that are not configured for multicast will not be able to receive the multicast, and will receive an error message instead. Use this feature when you are multicasting to a limited number of clients that you know can use multicast, or if you are multicasting a high-bandwidth presentation and do not want unicast to be an option.

Setting Up Scalable Multicasting

Just as in other live delivery methods, you indicate which live broadcasts are available for delivering in this format. In scalable multicasting, however, the settings that you configure for each live event are known as a *live channel*.

Perform the following steps for each live broadcast that you will be transmitting via scalable multicast.

1. Indicate the live channel for each scalable multicast. Each live channel has its own name, address and port numbers, and optional settings.
2. Create the link for the multicast.

In addition, you can configure some optional features:

- **Using unicast as a backup method**—configure options for what the client sees if the multicast cannot be completed as originally intended (go to a Web page or to another Server)
- **Controlling client statistics**—because scalable multicast is capable of handling thousands of client connections, all of which will send their connection statistics at the end of the presentation, RealServer enables you to redirect the client statistics to arrive at your Web server rather than RealServer.
- **Announcing multicast events**—your presentations can be announced automatically to programs that listen for multicast events. See “Publicizing Your Multicasts” earlier in this chapter.

Settings Used in Scalable Multicast

All scalable multicasts use the following setting:

- **Mount Point**— this will be included in the links to all scalable multicasts. The default value mount point is *scalable*. Locate by clicking **Multicasting>Scalable** in RealSystem Administrator.

Other settings are used, but they are different for each live channel. They are defined in the next section.

Setting Up a Live Channel

The steps in this section set up a live channel. You’ll need to know which addresses are available to you and how many to use; see “Determining the Number of Required Addresses and Ports” on page 192.

Note

These instructions describe only the steps required to set up this feature. For more options, see “Optional Scalable Multicast Features” on page 204.

- To create a live channel:

1. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.

2. Click **Add New**.
A generic channel name appears in the Edit Channel Description box.
3. Type a descriptive name for this multicast session in the **Edit Channel Description** box.
4. Click **Edit**.
5. Turn on scalable multicasting for this channel by selecting Yes from the **Enable Channel** list.
6. Type the name of the live clip in the **Path** box. The value you type here is the same as the path typed in the production tool that is encoding the live file. The information you enter here, in addition to the scalable mount point, will be included in the link for scalable multicast.

Tip

To make all live broadcasts available via scalable multicast, type an asterisk (*) here.

7. In the **Port Range** boxes, type the port numbers to which clients will listen for streams. The first port number must be an even number, and must be followed by a consecutive port number. Refer to “Calculating Addresses and Ports for Scalable Multicasts” on page 194 to determine the number of ports to use.
8. In the **IP Address Range** boxes, type the range of addresses for RealServer to use. RealServer uses the first available address in this range. To use a single address instead of a range, type the same address in each box. Refer to “Calculating Addresses and Ports for Scalable Multicasts” on page 194 to determine the number of addresses to use.
9. Indicate how far multicast packets can travel on your network in the **Time to Live** box. Use the values in the “Time to Live (TTL) Values” table on page 197.
10. From the **Reuse Address** list, select False if you want to use a separate address for each stream; select True if you want to use one address for both streams.
11. Type a value for **Timeout**. This represents the number of seconds a client will wait for multicast packets before it stops or uses the value in **Alternate URL**. See “Establishing Alternate Unicast Servers” on page 206.

Note

A video clip contains two streams: one each for audio and video. Refer to “Determining the Number of Required Addresses and Ports” on page 192 for complete information.

12. Click **Apply**.

Linking to Scalable Multicasts

Scalable multicasts use a different URL format than other material; when RealServer receives a request in this format, it sends the material differently and does not expect to establish or maintain a TCP connection. Instead, RealServer automatically creates an SDP (Session Description Protocol) file. SDP is a standard file format that contains information such as the multicast address and port, and the title, author, and copyright information.

The client receives this file when the user clicks the link to the scalable multicast. The Web browser downloads this file and sends it to the RealPlayer client software. The client software reads the contents of the file and connects to the scalable multicast.

A user can save the SDP file to disk (by right-clicking on the link in the Web page) and use it to connect later (by opening it with RealPlayer), and skip the step of downloading it from your RealServer.

Tip

The SDP file contains exact channel settings. If you will be repeating this multicast later, and still want users to connect with the same links, be sure to use the same channel settings that you used in the first multicast. The encoder settings, the addresses, and so on must be the same, or users who connect via saved SDP files will not be able to re-connect.

► To create links to scalable multicasts in a Web page

All links to scalable multicast content use the same format. Note that they always begin with `http://` and always end with the `.sdp` extension:

`http://address:HTTPPort/scalable/path/file.rm.sdp`.

Scalable Multicast RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming.
<code>address</code>	Address of RealServer; IP address or machine and domain name.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<code>scalable</code>	Scalable mount point, usually <code>/scalable/</code> .
<code>path</code>	Optional.
<code>file</code>	The file name itself.
<code>rm</code>	The file type or extension, such as <code>ra</code> , <code>rm</code> , or <code>rt</code> .
<code>sdp</code>	The <code>.sdp</code> extension is required for scalable multicast.

A link would look like the following:

```
<a href="http://realserver.example.com:8080/scalable/vivaldi.ra.sdp">
Click here to listen to today's Vivaldi selection</a>
```

The same link, if typed directly in RealPlayer or included in a SMIL file, would have the same format.

Notice that the mount point `/ramgen/` is not used.

Optional Scalable Multicast Features

The settings in this section are optional, and can be set for any live channel:

- Sending SAP Information with Your Multicasts
- Using Unicast as a Backup Method
- Controlling Client Statistics

Sending SAP Information with Your Multicasts

Session Announcement Protocol (SAP) information can be sent over the multicast-enabled network to announce your scalable multicast. See

“Publicizing Your Multicasts” on page 195 for instructions on configuring this option.

Using Unicasting as a Backup Method

When clients that are not multicast-enabled click the link to your scalable multicast presentation, they receive an error message. An error message also appears on a client screen when the multicast itself is interrupted at any point along the network. In both cases, the presentation halts.

The shift to unicast feature reconnects the client to a unicast version of the presentation automatically. You can also use this feature to customize the message those clients receive, or even redirect them to a multicast or unicast presentation. This backup presentation can be located on the same RealServer or on another RealServer.

You need only supply a single URL for your multicast if you enable this feature, rather than supplying a second URL for the unicast version.

You can enable this feature for individual live sources.

Do not enable this feature if you are multicasting a high bit-rate presentation where switching to unicast would flood your network.

► To use unicast as a backup method:

1. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.
2. In the **Channels** section, select the channel for which you want to configure this feature.
3. From the **Shift to Unicast** list, select Yes. The default value is Yes.
4. Click **Apply**.

Creating Custom Messages

In cases where clients would receive a generic error message informing them that the multicast was unavailable or available only to multicast-enabled clients, you can instruct RealServer to point the client to your own HTML page. Use your HTML page to post a custom message, such as “This presentation is only available to RealPlayers that have been configured to use multicast.”

► To create a custom message:

1. Create the Web page that you want clients to see in the event of an error.
2. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.

3. In the **Alternate URL** box, type the URL of your Web page. For example, type `http://www.example.com/mcast.html`.
4. Click **Apply**.

Establishing Alternate Unicast Servers

If you have a second RealServer which is transmitting the same broadcast, but in unicast, you can supply the source RealServer with the address of the second RealServer, and clients who are unable to receive the multicast presentation from the source RealServer will automatically be sent to the second RealServer.

Clients who cannot connect will be sent to the second RealServer, rather than receiving an error message and then halting the presentation.

Note

The steps in this section are the same as in “Creating Custom Messages”; use either an alternate URL or a custom HTML page per scalable multicast.

- To establish an alternate unicast RealServer:
 1. Create the Web page that you want clients to see in the event of an error.
 2. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.
 3. In the **Alternate URL** box, type the URL of the unicast presentation on the second RealServer. For example, type `rtsp://realserver.mycompany.com:554/encoder/vivaldi.rm`
 4. Click **Apply**.

Controlling Client Statistics

Clients such as RealPlayer have a feature that can send statistics to RealServer about the amount and quality of data they received while playing a presentation. The type of data sent by the client is controlled by RealServer’s Stats Mask setting.

As in unicast presentations, client statistics are sent to RealServer at the end of a presentation, and are stored in the access log. But scalable multicasts can serve to thousands of clients at the same time, and your RealServer may not be equipped to handle that quantity of simultaneous incoming client statistics.

RealServer includes features that help you manage two aspects of client statistics sent at the end of a scalable multicast:

- whether clients send statistics at all
- where those statistics are sent

Controlling Whether Client Statistics Are Sent

You can instruct clients not to send any data (set both Logging Style and Stats Mask to 0), but RealServer will still create a record for each connection, though it will record minimal data. (See the “Logging Style Effect on Access Log” table.) These settings will affect all material served by RealServer.

For scalable multicasts, RealServer has a feature that can instruct clients to not send any connection statistics at all. Enable this feature if your system cannot handle the volume of packets of data that would be sent simultaneously, or if you are simply not interested in these statistics.

In scalable multicasts, clients send statistics using an HTTP post.

- To control whether clients send statistics to RealServer:
 1. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.
 2. In the **Send Client Statistics** box, select Yes if you want clients to send their connection statistics at the end of a multicast presentation. Select No if you do not want clients to send any connection statistics.

Warning

If you set **Send Client Statistics** to Yes, be sure your RealServer can process the number of incoming statistics, or configure a Web server to receive the data as described in the next section.

3. In the **Web Server Address or IP Address** box, type the address of the RealServer.
4. In the **Web Server Port** box, type the port number of the RealServer.
5. Click **Apply**.
6. In RealSystem Administrator, click **General Setup** > **HTTP Delivery**. Make sure that `/scalable/` is on the list.

Controlling Where Client Statistics Are Sent

When clients initially connect to RealServer for a scalable multicast presentation, RealServer can instruct them to send connection statistics to a Web server, rather than to RealServer. Web servers may be better equipped to

handle volumes of simultaneously arriving data, since they are often configured to perform load balancing.

To use this feature, you will need to have a Web server available to you, and you will need to create a CGI script to receive the client statistics and to write them to a log file.

The statistics sent by the client to the alternate location are a subset of the statistics normally recorded in the access log. They have the following format (terms are defined in Chapter 19, “Reporting RealServer Activity”):

```
[Stat1][Stat2]#sent_time
```

The # symbol is used as a separator.

No other statistics are sent.

After the multicast event, you can look at the log file created by the cgi script and draw conclusions about the quantity of clients and quality of service.

► To control where clients statistics are sent:

1. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.
2. In the **Send Client Statistics** box, select Yes.
3. In the **Web Server Address or IP Address** box, type the address of the Web server that will receive the client statistics at the end of the multicast presentation.
4. In the **Web Server Port** box, type the port number of the Web server. If you omit this, RealServer will not multicast the event and will report an error in the error log file.
5. In the **Web Server CGI Path** box, type the path of the CGI script that will consolidate the client statistics in a log file on the Web server. For example, type `cgi-bin/client-stats/logstat`.
6. Click **Apply**.
7. In RealSystem Administrator, click **General Setup** > **HTTP Delivery**. Make sure that `/scalable/` is on the list.

Enabling RTSP-based Multicast Clients to Read SDP and SAP Files

RealServer 8 uses a slightly different file format for SDP and SAP files than earlier versions of RealServer. The new formats can be read by client software that is SDP-capable client (such as SDR). The old formats can be read by all versions of RealPlayer, but not from SDP-capable clients.

SDP Files

If you know that users will be viewing multicasts from a saved SDP file (instead of by clicking a link in a web page), and you know that they are using SDP-capable multicast clients, you will need to add a setting to your configuration file so that RealServer sends the SDP file in the new format.

In the `<List Name="Scalable Multicast">` list, and within the particular channel name, add the line `<Var DefaultSdpFileType="new"/>`. For example,

```
<List Name="Scalable Multicast">
  <Var MountPoint="/scalable/" />
  <List Name="Sources">
    <List Name="Channel1">
      <Var DefaultSDPFileType="new" />
    ...
  </List>
</List>
```

If you change this value to `old`, the SDP file can only be used by RealPlayer client software. Since you can make only one choice, there is also an option to override the configuration file setting by adding a switch to the URL. The switches are:

```
?SDPFileType=old?SDPFileType=new
```

The switch is appended to the end of the URL:

```
http://realserver.example.com:8080/scalable/concert.rm.sdp?SDPFileType=new
```

You might want to create two links, one for each SDP file type.

SAP

If you configured multicasts to be announced with automatically-generated SAP files, (as shown by the `AnnounceSAP` variable in either the scalable or backchannel multicast lists), you can also make the SAP variables include information that can be read by SAP-capable clients. Add the `<Var SDPFileType="new"/>` variable to the `<List Name="SAP">` list. For example:

```
<List Name="SAP">
  <Var SendAnnouncementEnabled="1" />
  <Var ListenAnnouncement="1" />
  <Var HostAddress="172.23.100.113" />
  <Var SDPFileType="new" />
</List>
```

The possible values for this variable are `new`, `old`, or `both`, where the value indicates which file format to send. Choosing both means that new and old

versions of the SAP file will be sent; this option results in slightly more overhead.

LIMITING ACCESS TO REALSERVER

RealServer has several methods of restricting access to content. Methods for restricting access to all material provided by RealServer include limiting the number of clients that can connect at any one time, limiting the amount of bandwidth that can be in use, requiring clients to be a certain version of the RealNetworks RealPlayer, or specifying that only multicast connections are permitted. In addition, you can restrict access based on the IP address of the client.

Overview

There are four methods which RealServer uses to block access, via connection volume or client identity. They are listed here, in the order in which they take effect:

1. Controlling access via HTTP.
2. Limiting the bandwidth or connections used.
3. Requiring a minimum player version.
4. Blocking or restricting access based on IP address of client.

Clients that do not meet the above criteria when requesting a presentation receive an error message.

Once a connection attempt is accepted, RealServer looks at the authentication information. Authentication, which can require a user name and password, is discussed in Chapter 15, “Authenticating RealServer Users”.

Controlling Access to HTTP Streams

RealServer can serve any content via HTTP, and includes a method for indicating which virtual paths contain content that can be served via HTTP. In this way you can protect your content but still serve HTML pages.

The list of virtual paths is pre-configured.

RealSystem Administrator uses the following entries in the HTTP Delivery list (you can view this list in RealSystem Administrator by clicking **General Setup > HTTP Delivery**):

- **HTTP Directories**—the default entries in this list are /admin, /httpfs, /viewsource, and /ramgen. The /scalable mount point is added automatically when you configure multicasting.

Warning

Do not add directories that contain secure material to this list, or users will not be prompted for their name and password when they view content in the secure directory.

Limiting Access by Number of Connections or Bandwidth

By using the **Maximum Client Connections** setting (the ClientConnections variable in the configuration file), you can limit the number of clients who connect simultaneously. Once this limit is reached, clients that attempt to connect receive an error message, and will not be able to connect until other clients disconnect.

Similarly, the **Maximum Bandwidth** setting limits the amount of bandwidth RealServer can use to any number of kilobits per second (Kbps).

If you establish values for both variables, RealServer will limit access when the lower threshold is reached.

► To limit access by limiting connections:

1. In RealSystem Administrator, click **General Setup**. Click **Connection Control**.
2. In the **Maximum Client Connections** box, type the number of client connections you want to allow simultaneously.

This number can be from 1 to 32767, as long as it is less than or equal to the number of streams permitted by your license. If it is 0 or blank, RealServer uses the number of streams specified by your license.

3. Click **Apply**.

► To limit access by bandwidth:

1. In RealSystem Administrator, click **General Setup**. Click **Connection Control**.

2. In the **Maximum Bandwidth** box, type the maximum number of kilobits per second (Kbps) that should be in use at once.

For example, to limit the bandwidth to one megabyte, specify maximum bandwidth usage by setting **Maximum Bandwidth** to 1024.

Tip

Your RealServer license may allow more bandwidth than is suitable for your network. Check with your network administrator to determine the right number to use.

3. When you have finished making changes, click **Apply**.

Limiting Access by RealPlayer Version

The setting for RealPlayer Plus Only means that only the RealNetworks RealPlayer Plus software can play presentations.

► To limit access to RealPlayer Plus:

1. In RealSystem Administrator, click **General Setup**. Click **Connection Control**.

2. In the **RealPlayer Plus Only** list, select **On**.

3. Click **Apply**.

Limiting Access to Back-Channel Multicast Reception

By setting **Enable Failover to Unicast** to **No** in the back-channel multicast section, you can require that clients within a certain range of IP addresses connect only in multicast mode. When this option is set to **No**, clients that are not able to connect in multicast mode receive an error message. If this option is **Yes**, clients that cannot connect in multicast mode can use unicast mode to

receive the presentation. This feature is described in “Requiring Multicasting Access Rather Than Unicasting” on page 200.

For scalable multicast, the **Shift to Unicast** feature provides the same functionality. See “Using Unicasting as a Backup Method” on page 205.

Limiting Access with the IP Address

You can block or permit access to RealServer material based on the IP address of the requesting machine and the port to which the request is made. Content is associated with specific port numbers—requests for streamed media arrive on the RTSP Port and PNA Port, HTML pages are made available via the HTTP Port. Encoders send their encoded material to the encoder ports. Server administrators use the Admin Port to access RealSystem Administrator.

The access control feature lets you associate certain client addresses with the ability or permissions to connect to certain ports.

For example, you could restrict which encoders can send encoded streams to your RealServer by restricting access to the encoding port (usually 4040). Or, you could allow only certain groups in your organization to view presentations served by your RealServer by listing their IP addresses and giving them access to your RTSP, PNA, and HTTP ports.

Clients whose IP addresses are configured with “deny” receive an error message indicating that the URL is not valid or that the connection has timed out.

A more selective form of restricting which material users can access (based on the directory or virtual directory where clips are stored) is authentication, described in Chapter 15, “Authenticating RealServer Users”.

When to Use Access Control

The following are considerations in deciding whether access control is right for your system:

- You want to restrict client access based on their IP addresses and you have a good way of figuring out their IP addresses
- You want to restrict access to certain features according to the users. For example, you only want to allow splitting from authorized RealServers.

Access Control Used with Other Features

This section describes the ways in which multicasting works together with other features.

On-Demand Streaming, Live Unicasting, G2SLTA, and Access Control

A client's address must be approved by the access control rules before being allowed to receive a stream or broadcast.

Splitting, Multicasting and Access Control

A client's address must be approved by the access control rules before being allowed to receive a stream or broadcast.

RealProxy and Access Control

If a client requests a cached stream, RealProxy will send the request to the source RealServer for permission before allowing the client to play the stream. If RealServer denies the request, RealProxy will not allow the client to receive the stream.

You can block a single RealProxy from caching the material served by your RealServer by creating an access rule that prohibits the IP address of that RealProxy from connecting to your RealServer.

Authentication and Access Control

Verification of the user's IP address takes place before authentication of user name or client ID. If a client fails the access control rules, authentication will not take place.

ISP Hosting and Access Control

Any client that tries to play a presentation hosted by an ISP-hosted customer will be compared against the access control rules before being allowed to play a clip.

Monitoring and Access Control

If a rule exists that allows access to the Monitor Port (9090) from a particular IP address, a user at that address can view and use Java Monitor.

Reporting and Access Control

The access log file shows clips served to clients that were accepted by the access control rules.

Understanding Access Control Rules

Information about each IP address or range of addresses you want to allow or restrict is stored in a rule. A rule is a set of instructions to RealServer about the address range and behavior to allow. Rules are identified by numbers which you assign, and are applied in ascending numeric order.

Before using this feature, you must make decisions about the types of rules you will create.

Each rule contains the following information:

- **Access Rule Number**—Identification number for this rule.
- **Access**—Whether the client will be allowed or denied access.
- **Client IP Address**—Client's address, or a range of addresses. This can also be an encoder's IP address.
- **Server IP Address**—RealServer's address.
- **Ports**—Port numbers to which access is specified. For general content viewing, these numbers correspond to settings on the Ports page: RTSP Port, PNA Port, and HTTP Port. For encoders, these correspond to the port numbers in the Broadcasting pages.

When a client attempts to play a RealServer presentation, or an encoder attempts to send material, RealServer compares its address and the requested port to the addresses and ports listed in the rules. You can create as many rules as you like. If the client's IP and requested port number do not match any rules, RealServer denies access.

For example, to allow a content creator to encode live material and send it to your RealServer, you would create a rule that listed the client's address and the encoder port (4040).

Deciding What Rules to Create

There are two ways you can restrict access, and these determine how you set up the rules. Create the third rule first, so that you will be able to connect to RealSystem Administrator and create the rest.

- **Specific Address Denial:** Deny a specific group of IP addresses and ports, and allow access to everyone else.
- **Specific Address Permission:** Allow a specific group of IP addresses and ports, and deny access to everyone else.

Both methods require that you set up three sets of rules:

1. The first set of rules refers to specific client addresses you are denying or allowing. There can be several rules that refer to specific addresses or ranges of addresses.
2. All clients not noted specifically in the first set of rules are allowed access (in Specific Address Denial) or denied access (in Specific Address Permission). This second set usually consists of a single rule which uses the word “Any” in the **From** box.

Warning

If you are using Specific Address Denial and you omit this step, you will deny access for everyone except those clients mentioned in the first set of rules.

If you are using Specific Address Permission, this set of rules is optional.

3. Finally, the last rule allows you to access to the RealSystem Administrator port.

As soon as you create one rule, RealServer assumes that all users not mentioned in the rule should be denied access. This is why you need to make enough rules to accommodate all conditions.

Note

Even if you are only interested in restricting access for a single client’s requests, you must still create all the rules necessary for your method.

Numbering the Rules

Rule numbers can be any length, but a number of more than one digit is recommended in case more lists are added later; with multiple digits, the new lists can be inserted between existing lists. When you create a rule, you give it a number. RealServer uses these numbers to sort the rules before it looks at a client’s request.

RealServer compares the client's IP address and requested port to the sorted rules, beginning with the lowest-numbered rule. As soon as RealServer finds a rule which matches the client's address, it allows or denies access, according to the rule's characteristics.

You do not have to create the rules in a certain order; RealServer will perform the sorting automatically.

Getting the Expected Connections

Because RealServer examines the rules in numeric order, you should make the lowest-numbered rules the most strict. Reserve high rule numbers for the most lenient rules. This is similar to schemas for firewall addresses.

Suggested Rule Schemes

Rule Set	Specific Address Denial	Specific Address Permission
Contents of Rules in Each Set		
1. Specific client addresses Suggested rule numbers: 100 - 490	Clients prevented from accessing RealServer. From setting: specific client addresses. Access setting: Deny Ports setting: <i>specific ports</i>	Clients permitted to connect to RealServer. From setting: specific client addresses. Access setting: Allow Ports setting: <i>specific ports</i>
2. All other addresses Suggested rule numbers: 500 - 990	Clients that can use your RealServer. From setting: Any Access setting: Allow Ports setting: <i>content ports</i>	Clients not permitted to use RealServer. From setting: Any Access setting: Deny Ports setting: <i>specific ports</i> This set of rules is optional.
3. Access to RealSystem Administrator Suggested rule number: 1000	All clients not listed in either of the rules above. From setting: Any Access setting: Allow Ports setting: <i>Admin Port</i>	All clients not listed in either of the rules above. From setting: Any Access setting: Allow Ports setting: <i>Admin Port</i>

Setting Up IP Access Control

There are two steps to setting up access control rules, regardless of which method you chose in "Deciding What Rules to Create":

1. Set up general rules which allow you to remain connected to RealSystem Administrator. You need only perform this set of steps once.
2. Create rules for specific IP addresses and port numbers.

Creating General Access Rules

The steps in this section create a rule that enables you to connect to RealSystem Administrator, regardless of the restrictions you create in other rules. Although it appears that you are allowing everyone to access RealSystem Administrator, the only people who will use it are other administrators who know the Admin Port number (chosen randomly at installation) and who have a user name and password specifically for RealSystem Administrator.

Warning

If you omit this initial step, you will not be able to connect to RealSystem Administrator when you restart RealServer, regardless of whether you have username-and-password permission.

Additional Information

To learn how to give access to RealSystem Administrator based on user name, see “Authenticating RealSystem Administrator Users” on page 228.

► To create the required access rule:

1. In RealSystem Administrator, click **General Setup**. Click **Ports**.
 2. Make a note of the **Admin Port** number. (This is the same number as the port number shown in your browser URL.)
 3. In RealSystem Administrator, click **Security**. Click **Access Control**.
 4. Click **Add New**.
- A generic access rule number appears in the Edit Rule Number box.
5. In the **Edit Rule Number** box, type 1000.
 6. Click **Edit**.
 7. From the Access list, select **Allow**.
 8. In the **Client IP Address** box, type Any.

For additional security, type the IP address or subnet address of users who will be permitted to use RealSystem Administrator.

If you type a subnet address, be sure to fill in the **Client Netmask** box with the appropriate subnet mask.

9. In the **Server IP Address** box, type Any.

10. In the **Ports** box, type the Admin Port number you noted in Step 2.
11. Click **Apply**.

You will now be able to access RealSystem Administrator, no matter what rules you create in the next section.

Creating Specific Access Rules

Use the steps in this section to allow or deny access to specific IP addresses or address ranges.

Warning

Be sure to first follow the steps in “Creating General Access Rules”, or you will not be able to access RealSystem Administrator after you restart RealServer.

- To limit access according to IP number:

1. Determine the port numbers in use, for Step 10:

- **Users**—If this rule will refer to users who want to play streaming media, click **General Setup>Ports**. Make a note of the values for **PNA Port** (usually 7070), **HTTP Port** (usually 8080), and **RTSP Port** (usually 554).
- **Encoders**—If this rule will refer to version 6 and later encoders that will be sending content to your RealServer, click **Broadcasting>Encoder**. Make a note of the value for **Port** (usually 4040).
- **Pre-G2 Encoders**—If this rule will refer to pre-G2 encoders that will be sending content to your RealServer, click **Broadcasting>Pre-G2 Encoder**. Make a note of the value for **Port** (usually 5050).
- **Receivers**—To allow pull splitting receiver connections, look at the port number (usually 2030) in **Splitting>Transmitter**. For push splitting, check the **HTTP Port** number (usually 8080) by clicking **General Setup>Ports**.
- **Java Monitor**—To reference Java Monitor, use the **Monitor Port** number (usually 9090), shown on **General Setup>Ports**.

2. In RealSystem Administrator, click **Security**. Click **Access Control**.
3. Click **Add New**.

A generic rule number appears in the Edit Rule Number box.

4. In the **Edit Rule Number** box, type a three-digit number for the new access rule. RealServer implements the rule numbers in numeric order.

Tip

Technically, you can type any number in this box. But because rules are sorted numerically, and because the rule that enables access to RealSystem Administrator must be the last rule on the list, use a three-digit number here so the RealSystem Administrator rule (given as rule 1000 in “Creating General Access Rules”) can be the last rule on the list.

5. Click **Edit**.
6. Indicate whether permission is being granted or refused by selecting **Allow** or **Deny** from the **Access** list.
7. In the **Client IP Address** box, type the IP address of the client machine.
8. To indicate a range of addresses, type the netmask in the **Client Netmask**.

Tip

To refer to all clients, regardless of IP address, type the word **Any** in the **Client IP Address** box, and leave the **Client Netmask** box blank.

9. In the **Server IP Address** box, type the address of the RealServer machine or network card which is hosting the requested content.

You can type a specific address, or use the word **Any** to refer to any IP address on the RealServer machine.

- If you type a specific IP address or DNS name, you must also add that address to the IP Binding list. See “Distributed Licensing” on page 109 for information.
 - Avoid using 127.0.0.1 or localhost, unless you will only be using test links which use that exact text in their links.
10. Finally, list the RealServer port numbers to which you want to restrict access. In the **Ports** box, type the port numbers, separated by commas. To restrict access to all RealServer content, the port numbers should match the other port numbers you’ve instructed RealServer to listen to;

look at the port numbers for RTSP port, PNA port, HTTP port, and the port value used by the encoder.

11. Click **Apply**.

Chapter 15

AUTHENTICATING REALSERVER USERS

RealServer authentication provides a way for you to control what or who can access your RealServer, whether it is an encoder sending a stream, a colleague perusing RealSystem Administrator, or a user viewing content for which they've paid.

Overview

Authentication verifies the identity of the users or software that make requests of RealServer. The verification can come in the form of asking for a name and password, or it can be hidden from the user.

You can require a name and password for the following RealServer areas:

- **Encoder users**—Require content creators to supply a name and password before sending their live streams to RealServer. In this way, only authorized people can send streams to your RealServer.
- **RealSystem Administrator users**—Allow certain people in your organization to use RealSystem Administrator. To protect your RealServer from changes made by unauthorized users, RealServer is installed with authentication turned on for RealSystem Administrator access.
- **Individual users**—The most popular use of all is limiting user access to individual presentations or directories of clips. An additional database can be used to list which content each user can view, and what type of access they have. Several additional options are available for this type of authentication.

The names of authorized users for each item above are stored in separate databases. One database stores the names for the authorized encoder users, another stores names of other administrators, and still another stores names of people who can view presentations. You can set up additional authentication areas and databases.

RealServer identifies requests (in the form of URLs) for secure content by the mount point. The URL must contain the mount point, and it may contain additional directory information. Encoders are an exception to this—RealServer looks at the port number at which live data arrives in deciding whether it should accept the connection.

Additional Information

To limit visitors to RealServer via bandwidth, connection volume, client version, or IP address, use the methods described in Chapter 14, “Limiting Access to RealServer”.

Compatible Client Versions

RealPlayer versions 3 and earlier do not work with authentication and may display an error message. RealPlayer 4 works with player validation only. RealPlayer 5 and later support both player validation and user authentication.

When to Use Authentication

The following are factors in deciding to use this feature:

- You are hosting material to which you want to limit access, and you want to use a more specific method than noting the client’s IP address. (The access control list enables access based on the client’s address.)
- You want to allow different types of access to different types of material.
- You want to collect data from users before they play your clips. (Collecting data is not necessarily part of authentication; it is just something you can require if you implement authentication.)
- You want clients to pay money before or after playing clips.
- You want to track how much time specific users are playing certain clips.

Authentication Used with Other Features

Authentication works with all other RealServer features. There are few special considerations for each feature, however.

On-Demand Streaming and Authentication

All on-demand files stored in the Secure directory (or in any subdirectories) are authenticated automatically, once the authentication feature has been set up.

Live Unicasting and Authentication

Once the authentication feature has been set up, live broadcasts are authenticated automatically if they include `/secure/` as part of the path when you encode the events.

Archiving and Authentication

Archived files are on-demand files; they can be authenticated if they are moved to the correct location first. They must be placed in the `Secure` directory or in a subdirectory of `Secure`, or the archiving feature must be configured to use the `Secure` directory.

G2SLTA and Authentication

Just like any other live event, broadcasts created by **G2SLTA** can be authenticated, as long as you include `/secure/` in the path you type for *livefile*. For an example, see “Unicast Live Content (from Encoders)” on page 367.

Splitting and Authentication

If you are sending a stream to a RealServer that is acting as a receiver, you must put copies of all the databases that store authentication information on the receiver. This distributes the authentication load.

Multicasting and Authentication

In both back-channel and scalable multicasts, the user or client is authenticated through the initial control channel connection. Be sure the multicast (either `/` or `/scalable/`) path is on the list of Commerce Rules.

RealProxy and Authentication

RealProxy makes requests on behalf of clients, and caches the streams it receives. Although RealProxy stores the streamed data, it requires a control channel between the requesting client and RealServer. RealServer uses the control channel to request and receive authentication information.

Firewalls and Authentication

Authentication is performed over the two-way control channel. As long as the client can establish a connection through the firewall to RealServer, all material can be authenticated for clients who are behind firewalls.

Access Control and Authentication

The access control feature, which checks the client's IP address against a list of allowed addresses, takes place before authentication. So if a client's IP address is blocked, authentication will not take place. If you have users who should be able to play secure material are receiving error messages, check the list of access rules to see if their client's address is disallowed.

ISP Hosting and Authentication

Authentication of content cannot be applied to the files of ISP-hosted customers. Their material is always available. Depending on the access needs, you may be able to apply access control rules so that customers can allow or deny certain users' access to content.

Monitoring and Authentication

You can monitor which secure presentations are in use by viewing the paths of the files in Java Monitor. Those that contain the `/secure/` mount point are authenticated.

Reporting and Authentication

Efforts to authenticate users are not included in the access log; records are created for successful serves. You can identify authenticated material in the access log by the GET statement; secure material always contains the `/secure/` mount point in the path.

In addition, connection attempts for authenticated material are stored in the `accesslog.txt` file in the Logs directory of appropriate data storage directory (if you are using the text file method), or in the `Access_log` table (if you are using the database method).

Adding User Names and Passwords

Use the following instructions to add to the list of authorized users for any type of authentication.

Note

If you are using Window NT to manage the list of users, passwords, and groups, use those tools instead of the instructions below.

- To add a user name and password:
1. In RealSystem Administrator, click **Security**. Click **Authentication**.
 2. In the **Authentication Realms** list, select the name of the realm to which you want to add a user:
 - For encoder users, select SecureEncoder.
 - For RealSystem Administrator users, select SecureAdmin.
 - For content authentication users, select SecureContent.
 - For any other category of authentication, select the name of the realm. (To learn more about realms, see “Creating a New Realm” on page 239.)
 3. Click **Add a User to Realm**.
 4. In the new window that appears, type the user’s name in the **Name** box.
 5. In the **Password** box, give the user’s password.
 6. In the **Confirm Password** box, type the password again.
 7. Click **OK**.

Authenticating Encoder Users

Encoder user authentication is configured automatically at setup; when you installed RealServer, you gave a user name and password for RealServer to use. These were added to the administrator database and the encoder database. Users of RealSystem encoders version 6 and later must supply this user name and password to connect.

Additional Information

Other authentication features that can be used in authenticating encoder users are listed in “Optional Authentication Features (for All Types of Users)” on page 238.

Encoders

RealServer uses the following settings for encoder user authentication (you can view these settings with RealSystem Administrator by clicking **Broadcasting>Encoders**):

- **Encoder Authentication Realm**—a realm to use for encoders is included with your RealServer installation, named EncoderRealm. If you want to use a realm which does not yet exist, see “Creating a New Realm” on page 239.
- To add user names and passwords for encoders:
- Add each encoder user and password with the instructions in “Adding User Names and Passwords” on page 226; use SecureEncoder for the realm.

Pre-G2 Encoders

Content creators who use older encoders need only supply a password, but the password must be the same for everyone. During installation, you were prompted for this password. If you change the password with RealSystem Administrator, be sure to tell the new password to everyone who will be connecting.

RealServer uses the following settings for encoder user authentication (you can view these settings with RealSystem Administrator by clicking

Broadcasting>Pre-G2 Encoders):

- **Password**—the password which all pre-G2 encoders must supply in order to connect to RealServer.

Authenticating RealSystem Administrator Users

At installation, RealServer is configured to prompt all RealSystem Administrator users for a user name and password. Use the user name and password you entered during Setup.

Additional Information

Other authentication features that can be used in authenticating RealSystem Administrator users are listed in “Optional Authentication Features (for All Types of Users)” on page 238.

- To add user names for RealSystem Administrator authentication:
- Use the instructions in “Adding User Names and Passwords” on page 226; use SecureAdmin for the realm.

Authenticating Content Users

There are four steps to setting up authentication for users and content:

1. Add user names and passwords for users.
2. Give those users access to specific content.
When RealServer is first installed, it is automatically configured to look for on-demand secure content in the Secure directory.
3. Create the content.
4. Create links to the secure content.

This section describes the steps necessary for authenticating users who request secured media.

Additional Information

There are two sets of optional authentication features that can be used in authenticating content users: those listed in “Optional Content User Authentication Features” on page 232, and the more general options shown in “Optional Authentication Features (for All Types of Users)” on page 238.

Step 1: Adding User Names and Passwords

Add user names and passwords by following the instructions in “Adding User Names and Passwords” on page 226; use SecureContent for the realm.

Note

A type of authentication which does not require user names and password is also available; refer to “Player Validation” on page 233.

Step 2: Giving Users Access to Content

The steps in this section assign access permission to all content stored in the default supplied directory. For live content, these steps use the default live encoding path.

You can have multiple directories that contain secure material, and they can be in different physical locations.

► To give users access to secure content:

1. In RealSystem Administrator, click **Security**. Click **Commerce**.
2. Choose the rule that matches the type of content you’re delivering:

For on-demand content:

- a. From the **Commerce Rules** list, select SecureUserContent.
- b. In the **Protected Path** box, verify that /secure is typed in the box. (This is the name of the default secured directory.) If your content is in a subdirectory, type the full path here. For example, add a subdirectory named Earnings as /secure/Earnings.

For live content:

- a. From the **Commerce Rules** list, select SecureG2LiveContent.
- b. In the **Protected Path** box, verify that /encoder/secure is typed in the box. (This is the name of the default secured directory.) If your content is in a subdirectory, type the full path here. For example, add a subdirectory named Earnings as /secure/Earnings.

Note

If you use subdirectories, be sure to also add the main secured directory as a Protected Path, or anything you put in the main directory will not be authenticated!

3. Use the supplied values for all other settings.
4. Click **Grant Permission**. A new browser window appears.
5. In the **Username** box, type the name of the first user whose account you created in “Step 1: Adding User Names and Passwords”.
6. From the **Access Type** list, select the type of access you want to give.

Permission Types

Type	Access to presentation or presentations in a directory
Event	Unlimited viewings of a given clip.
Calendar	Permission expires on a certain date and time. If the date and time of expiration arrives while the visitor plays a clip, transmission of that clip to the player is stopped, and an error message appears.
Duration	User gets a finite amount of time to view presentations. All viewing time is deducted from this amount, and when time expires, permission is revoked.
Credit	Time spent viewing presentations is noted by RealServer, and the administrator can use this information later for billing purposes. Access is granted per presentation or directory, and is unlimited.

7. Depending on the option you selected, another dialog box may appear; type the appropriate value in this box.
8. Use the default values for all other settings in this browser window.
9. Click **OK**.

Step 3: Creating the Content

For on-demand content, place the files in the Secure subdirectory of the main RealSystem Administrator.

For live content, encode it using the virtual path encoder/Secure.

Step 4: Linking to Authenticated Content

Links to individual on-demand or live streams are similar to their non-authenticated counterparts, with the addition of the /secure/ mount point.

- To link to authenticated on-demand content:

The link in a Web page to on-demand content, located in the Secure directory, has the following format:

`http://address:HTTPPort/ramgen/secure/path/file`

where:

Authenticated On-Demand Content URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming.
<code>address</code>	Address of RealServer; IP address or machine and domain name.
<code>HTTPPort</code>	Port number where RealServer listens for this type of request.
<code>ramgen</code>	Ramgen tells RealServer to create a Ram file that the client will use.
<code>secure</code>	Invokes the authentication feature.
<code>path</code>	Optional; the subdirectory, relative to the base path of the mount point, where the content is located. If the file is located in the base path itself, omit <code>path</code> .
<code>filename</code>	The name of the presentation, including the extension.

- To link to authenticated live content:

Live content which will be authenticated has this format:

`http://address:HTTPPort/ramgen/encoder/secure/path/file`

Notice that it includes the `/encoder/` mount point, which invokes the live broadcasting feature.

Optional Content User Authentication Features

There are several more options in setting up content authentication than for encoder or RealSystem Administrator user authentication:

- Allowing Users to View a Clip from More Than One Location
- Player Validation
- Registering Player Validation Users Via Links
- Using the Evaluate Permissions Feature
- Working with SMIL Files

Allowing Users to View a Clip from More Than One Location

If you want a user to be able to log in at more than one location and view the same content from more than one location, set **Allow Duplicate IDs** to Yes.

Normally, when **Allow Duplicate IDs** is set to No, a user can view a given clip from only one computer at a time. If a user tries to log in from a second computer and view the same content, he or she will receive an error message. The user must log out at the first location before being permitted to log in at the second location. Users will still be able to view different content even though they are logged in at different locations.

- To allow users to view a clip from more than one location:
 1. In RealSystem Administrator, select **Security**. Select **Commerce**.
 2. From the **Commerce Rules** list, select the rule you created in “Step 2: Giving Users Access to Content”.
 3. Set **Allow Duplicate IDs** to Yes.
 4. Click **Apply**.

Player Validation

Player validation allows or denies access to individual clients (usually one per computer), rather than to specific people, and authentication is transparent to the visitor—a dialog box warning only appears when the visitor attempts to access content for which he or she is not authorized. The user isn't asked for a user name and password. This type of authentication involves less viewer interaction than regular user authentication, but each client must be registered individually by the viewer or RealServer administrator.

Player validation requires a user name the first time the user registers. The player ID is associated with the original user name, no matter who is using the client software.

► To use player validation:

1. In RealSystem Administrator, select **Security**. Select **Commerce**.
2. From the **Commerce Rules** list, select the rule you created in “Step 2: Giving Users Access to Content” on page 229.
3. From the **Credential Type** list, select Use Player Validation.
Additional options appear in the lower right portion of the browser window.
4. In the **Player Registration Prefix** box that appears, type a word to use as a registration prefix. The word you use here must be unique; it will be used in the URL for player validation, and identifies to RealServer which Commerce Rule applies to the request.
5. The **Redirect Protected Path** option enables you to supply a clip which unregistered Players will receive if they click on a link that requires authentication. You might create a clip with a recording of instructions for registering. (Remember to use a non-authenticated clip, or unregistered users won't receive your alternate clip.) To use this option:
 - a. Click **Redirect Protected Path**.
 - b. In the new browser window that appears, type the URL for the clip in the **Destination URL** box. Do not type the full name; instead, type the URL without the protocol and server name. Thus a link that you would normally type as `rtsp://realserver.example.com:554/video/real8video.rm` would appear in the **Destination URL** box as `video/real8video.rm`.
 - c. Click **OK**.

6. Click **Apply**.

Using Player Validation with Generic Identifiers

Because of privacy concerns, users can elect to suppress the unique identifiers which RealServer and other applications use to identify RealPlayers; instead, RealPlayer will send a generic identifier (GUID) consisting of zeroes.

If users were allowed to register RealPlayers that use this particular GUID, all users with that GUID would log in using the first zero-GUID user's settings. You would not get a true picture of the number of users. To prevent this, RealServer's predefined data stores contain a single record for a "dummy" user with the all-zero GUID, and no permission is granted for this user to access any secure content. All users who have the zero-GUID option selected in their RealPlayer will be denied access to secure content.

For more information about RealPlayer and privacy, please read RealNetworks' Consumer Software Privacy Statement:

<http://www.realnetworks.com/company/privacy/software.html>.

Automating Registration

In its default state, RealServer requires that you individually add the names of users to the appropriate databases before they can receive secure content. This is feasible if you are administering RealServer for a small site. But in case you want to allow users to register themselves via a Web page, some versions of RealServer include a sample CGI program and HTML files that interact with a Web site and your RealServer so that users may register themselves. To set up self-registration, you'll need to customize two sets of supplied files: HTML pages containing forms for registration, and .cgi files that connect the .htm files with RealServer and the data storage files. Instructions are located within the Commerce subdirectory of the main RealServer directory.

Registering Player Validation Users Via Links

To quickly register individual RealPlayers when player validation is in use, you can customize the link that users click so that RealServer registers and recognizes the client software in one step.

You can also use these instructions as a basis for writing your own CGI scripts and Web pages to accomplish the same purpose automatically.

Note

If you are automating this procedure, you may omit Step 1. A realm is only required if you add users from RealSystem Administrator.

► To register players:

1. Create a realm to use for player validation:
 - a. In RealSystem Administrator, click **Security**. Click **Authentication**.
 - b. In the **Authentication Realms** area, click **Add New**.
A generic realm name appears in the Edit Realm Description box.
 - c. In the **Edit Realm Description** box, type a name for this realm. For instance, type SecurePlayer.
 - d. Click **Edit**.
 - e. In the **Realm ID** box, type a unique text string. For example, SP.
 - f. From the **Authentication Protocol** list, select RealSystem 5.0.
 - g. From the **Database** list, select the supplied PlayerContent option.
 - h. Click **Apply**.
2. Add a user name:
 - a. In RealSystem Administrator, click **Security**. Click **Authentication**.
 - b. In the **Authentication Realms** list, select SecurePlayer.
 - c. Click **Add a User to Realm**.
 - d. In the new window that appears, type the user's name in the **Name** box. (You don't need to supply a password for Player Validation). For example, type Chris.
 - e. Click **OK**.
3. Give permission to the user you just created:
 - a. In RealSystem Administrator, click **Security**. Click **Commerce**.
 - b. From the **Commerce Rules** list, select SecurePlayerContent.
 - c. Make a note of the value shown in the **Player Registration Prefix** box; you'll use this value in the self-registration link. The default value is register.
 - d. Click **Grant Permission**.

A new browser window appears.

- e. In the **Name** box, type a user name.
- f. Click **OK**.

4. Now register a RealPlayer with the special self-registration URL. (The syntax of the special URL format is shown “Web Page Self-Registration URL Format” on page 236.)

- For a Web page, use a link such as the following.

`http://realserver.example.com:8080/ramgen/registerChris!real8audio.rm`

The user clicks the link and is added to the database; she can then begin viewing secure content.

- For a link that the user can paste into the **File>Open Location** box of RealPlayer, create a link such as:

`rtsp://realserver.example.com:554/registerChris!real8audio.rm`

Web Page Self-Registration URL Format

The following shows the syntax to use in creating a link that can be used to register Players automatically, as shown in the preceding section.

`http://address:HTTPPort/ramgen/PrefixUsername!file`

RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming. Always use <code>http</code> in Web pages.
<code>address</code>	Address of RealServer; IP address or machine and domain name.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. Its default value is 8080.
<code>ramgen</code>	Ram file generator mount point.
<code>Prefix</code>	Prefix as shown in the Player Registration Prefix box (on the Security>Commerce page of RealSystem Administrator).
<code>Username</code>	User’s name as it will appear in the database; you typed this in Step e of “Add a user name.”

(Table Page 1 of 2)

RealServer URL Components (continued)

Component	Meaning
!	This delimiter separates the user name from the file name.
<i>file</i>	The name of the presentation, including the extension. Do not use a file located in a secure directory.

(Table Page 2 of 2)

For a link that users can type or paste directly in the **File>Open Location** dialog box in RealPlayer, use the following syntax:

```
rtsp://address:RTSPPort/PrefixUsername!file
```

Using the Evaluate Permissions Feature

Once RealServer has verified the identity of the user or client, an additional level of verification is available: it can allow access to all files or only to very specific files. **Evaluate Permissions** controls this; when set to No, it allows general access to all authenticated users or players. When set to Yes, RealServer performs the additional step of examining the requested URL and looking for it in the database. If the user or player who requested it has permission for that clip or content path, RealServer streams the requested file.

If you'll be looking up permissions for specific files or directories, you must also indicate the database which stores the clip permission information. This database can be the same as the database that stores user names and passwords.

- To set up access to all material or to specific material only:
 1. In RealSystem Administrator, click **Security**. Click **Commerce**.
 2. From the **Commerce Rules** list, select the rule you created in "Step 2: Giving Users Access to Content".
 3. From the **Evaluate Permissions** list, select Yes.
 4. Click **Apply**.

You can use a different setting for each rule.

If you selected this box, you must also set up the different permissions type for each user and each clip or directory to which you'll be giving them access. See the following section for a list of the different permission types.

Working with SMIL Files

Users are prompted only once per realm for name and password for SMIL files, regardless of how many files in the presentation require a name and password. When the user clicks on a link to a SMIL presentation that contains secure materials, RealServer prompts the player for security information on the first clip. The player then prompts the user for an authorized name and password. The player then stores the information and supplies it when the RealServer asks for security information on remaining clips in that realm.

Should any clip in the presentation expire sooner than the others, the entire presentation will halt. The person viewing the presentation will not be able to continue until more time is allotted by the administrator.

For this reason, it is important that all the permissions on all the files within a presentation be the same.

The best methods of organizing authentication and SMIL files are the following:

- Authenticate the SMIL file but not its contents (use if you do not need high security levels).
- If you are using duration access, use it only for the longest file in the presentation (usually the audio or video file). Apply event access for the other files.

SMIL Files and Directory-Level Duration Access

This is one case in which giving identical permission to all files (including the SMIL file) will not work as expected.

As each clip is viewed, RealServer subtracts the viewing time from the directory. If each clip is 10 minutes long, and there are three clips in the presentation, RealServer subtracts 30 minutes from the total viewing time. This means that in setting up this type of access, the time allotted must be the sum of all the clips.

Keeping track of all the clips, their lengths, and the total directory access time can be tricky. A better solution is to limit the access time only for the SMIL file.

Optional Authentication Features (for All Types of Users)

RealServer has these optional features, which apply to all types of users:

- Creating a New Realm
- Creating a New Database
- Changing RealSystem 5.0 Authentication Passwords

Creating a New Realm

A realm contains information about the type of authentication protocol and the database where the authenticated users' names will be stored.

RealServer has three authentication protocols for authenticating the identity of visitors:

- **Basic**—encodes the user's name and password with the Base64 algorithm and sends it to RealServer, which then decodes the password and verifies it. This protocol sends the user's password over the public Internet in a simple manner. Users should use a unique password for this material.
- **RealSystem 5.0**—also called *RN5*, it is RealNetworks' own authentication protocol, developed for RealServer version 5. If your material will be served to users working with RealPlayer version 5 and later, use this authentication protocol. This is a more sophisticated protocol than Basic authentication. It provides better security than Basic because it does not send the password in a manner that can be reversed.

If the clients that will be accessing content on your RealServer are RealPlayer 5 and earlier, be sure to use the RealSystem 5.0 style for content authentication.

- **Windows NT LAN Manager**—this method enables RealServer to use the existing NT database of user groups and permissions. It also allows access control of content via NTFS file permissions. The protocol uses Windows NT authentication.

This method is only available to systems using Windows NT, and requires that RealServer itself be installed on an NT Server. For authenticating content, it also requires Microsoft Internet Explorer and RealNetworks RealPlayer.

► To create a realm:

1. In RealSystem Administrator, click **Security**. Click **Authentication**.
2. Click **Add New**.

A generic realm name appears in the Edit Realm Description box.

3. In the **Edit Realm Description** box, type a name for this realm.
4. Click **Edit**.
5. In the **Realm ID** box, type a name. You will use this name in other areas of RealSystem Administrator, so make a name that is meaningful to you. The Realm name may also appear to users as part of the name and password prompt.
6. In the **Authentication Protocol** list, select the authentication method you want to use for this realm:
 - **Basic**—use for authenticating encoders and RealSystem Administrator. You will also need to select a database in which the names and passwords of authenticated users will be stored; refer to “Creating a New Database” on page 240.
 - **RealSystem 5.0**—use for authenticating encoders and content users. You will also need to select a database in which the names and passwords of authenticated users will be stored; refer to “Creating a New Database” on page 240. In addition, these passwords are encrypted. To change them, refer to “Changing RealSystem 5.0 Authentication Passwords” on page 242.
 - **Windows NT Lan Manager**—use for encoders, RealSystem Administrator, and content users (on intranets only). You do not need to select a database—instead, RealServer will use the NT list of names. Use the additional steps shown here:
 - a. Type the appropriate provider in the **Provider** list, such as NTLM.
 - b. Type the Group name in the **Group** box.
7. Click **Apply**.

Creating a New Database

The list of databases groups database interfaces and the locations of databases. RealServer includes four database interfaces:

- **Flat file**—The text file method is enabled during installation, as it enables the greatest insight into the access permission structure, but the text file method lacks the flexibility of a full database application. It’s best to use the text file method only for simple tracking or for troubleshooting the system before linking a full-fledged database to

RealServer. For small-scale data, the text file method is also faster than a full-fledged database.

- **ODBC and MySQL**—The authentication package contains templates for common databases, including mSQL and the ODBC-compliant databases Microsoft Access and Microsoft SQL Server. Users can also work with databases for which templates do not exist, by setting up the data source with the appropriate table structure. The mSQL database is generally used on UNIX.
- **RN5 DB Wrapper**—If you used authentication features with RealServer version 5, or if you have a data store plug-in created by a third-party company, you can still use that plug-in with RealServer 8.

Use the instructions below to choose the name and type of database that will store users' names and passwords.

► To set up a database:

1. In RealSystem Administrator, click **Security**. Click **Databases**.
2. Click **Add New**.
A generic database name appears in the **Edit Database Name** box.
3. Type a description for the new database in the **Edit Database Name** box.
4. Click **Edit**.
5. From the **Database Type** list, select the data storage method you want to use.
6. Depending on the database type method you chose, additional information is required.

Flat File needs only the path to the main text file directory. For example, the `enc_r_db` directory under the main RealServer directory. See “Overview” on page 245.

mSQL has two required names, and three optional items:

- **Host Name**—IP address or DNS name of computer where database is stored.
- **Database Name**—Name of the database.
- **Table Name Prefix**—Prefix used to make field names unique, when used with an existing database.
- **User Name**—Name required by database application.

- **Password**—Password required by database application. Re-enter your password in the **Confirm Password** box to ensure you typed it correctly.

ODBC uses the same information as mSQL, but ODBC does not ask for a Host Name. (Refer to “Setting Up Other Types of Data Storage” on page 253 for further instructions.)

RN5 DB Wrapper—these items are needed:

- **Database Name**—name or location of the data storage plug-in. Consult your plug-in documentation for information about what should go here.
- **Plugin Path**—Location of the plug-in.
- **User Name**—Name required by the database application.
- **Password**—Password required by the database application. Re-enter your password in the **Confirm DB Login Password** box to ensure you typed it correctly.

7. After filling out the appropriate values, click **Apply**.

Changing RealSystem 5.0 Authentication Passwords

In RealSystem 5.0 authentication protocol, RealServer stores all passwords in an encrypted format. Passwords can be entered and changed through the RealServer Administration page.

If you want to change the passwords manually, without using RealSystem Administrator, you can use the supplied password command line utility. It is located in the RealServer Bin directory.

You can also use these instructions as a basis for writing your own CGI scripts and Web pages to accomplish the same purpose automatically.

► To use the password tool manually:

1. At a command line, in the Bin directory, type the following:

```
mkpnpass username realm
```

where:

username is the user name exactly as it is entered or will be entered in the authentication database or text file.

realm is the value of the Realm variable specified in the relevant list. For encoders, this is given by **Authentication Realm** on the

Broadcasting>Encoders page in RealSystem Administrator. (In the configuration file, it is given by the value of the Realm variable in the G2_Encoders list.)

For RealSystem Administrator users, use the value of the Realm variable in the RealAdministrator_Files list within the FSMount list in the configuration file. (You must open the configuration file itself to see this value.)

2. A password prompt appears, followed by a prompt to type the password again.

The resulting encrypted password is displayed on the screen.

RealServer encrypts passwords with the MD5 hashing algorithm. It uses the form MD5("username:realm:new_password"). On BSD systems and some other UNIX systems, you can generate these passwords with the following command:

```
echo -n "username:realm:new_password" | md5
```

3. Add the resulting encrypted password into the appropriate field of the database.
 - For text files, place it in the password field of the User directory (see “Users Directory” on page 247).
 - For databases, place it in the password field of the Users table (see “Users Table” on page 250).

Chapter 16

STORING AUTHENTICATION DATA

After a user has been granted access by the authentication feature, RealServer can check to see whether he or she has special permissions for viewing specific presentations or directories of presentations. You can use this information for applications such as pay-per-view. Working with the authentication feature, permission information is stored in a separate database. This chapter describes the data storage methods which can be used with the authentication feature.

Overview

To authenticate visitors, the RealServer stores user IDs and passwords or client IDs, and their associated access permission information. When a client tries to access a clip, the RealServer looks up this information to see whether the client or visitor is authorized to view the clip. The information can be stored in either a series of text files or in a database. Templates for common databases are installed during installation.

This section describes the methods for storing user name and password data. Templates for common databases are created during installation, that correspond to the database methods listed in “Creating a New Database” on page 240.

- **Text file storage**—this method uses a combination of directory structure and text files to achieve a sensible data storage method. It is the default method. See “Using Text Files for Authentication Data” on page 246 for details.
- **Database templates**—the supplied templates use a similar structure to the text file method, in a more familiar database format. Refer to “Using a Database for Authentication Data” on page 250 for more information.

Using Text Files for Authentication Data

The default configuration uses the text file storage method to provide storage for all three default realms.

The following directories contain the text files which store data. The center letter indicates the authentication protocol: *r* is for RN5, *b* is for Basic.

Supplied Data Storage Directories

Directory Name	Data Storage for the following type of information
enc_r_db	Encoder User Authentication
adm_b_db	RealSystem Administrator User Authentication
con_r_db	Content Authentication

The contents of the directories are given in the table below.:

Text File Storage Directory Structure

Directory	Contents	File or Directory Description
Main directory (con_r_db, enc_r_db, or _adm_b_db)	ppvbasic.txt	The text file indicates to RealServer that this is the storage area for the list of authenticated names.
users	(initially blank)	Files in this directory list the clips and permission types.
guids	(initially blank)	For player validation, files connect the clientID with a user name.
logs	reglog.txt accesslog.txt	See below for a description of these files.
redirect	(initially blank)	For player validation, files contain an URL to which to send the client if redirection is necessary.

The actual data storage text files do not exist when RealServer is first installed. They are created when authentication is in use and secure content is first requested. When RealServer creates the file structure, it creates the ppvbasic.txt file. The second and subsequent times you start the RealServer, the RealServer looks for this file. If the file does not exist, it recreates the directory structure.

Warning

Do not delete the ppvbasic.txt file! If you delete the ppvbasic.txt file, RealServer will rewrite the directories and will erase their prior content.

Users Directory

The files in this directory are named *username*, where *username* is the user name. This directory contains one file per registered user.

The first line of each file has the following format and is different than subsequent lines in the file:

```
password;uid;uid_writeable
```

where:

<i>password</i>	When user authentication is in use, this stores the password. Otherwise shows an asterisk (*). Note: Passwords are encrypted. See “Changing RealSystem 5.0 Authentication Passwords” on page 242.
<i>uid</i>	In player validation, stores playerID. In user authentication, an asterisk (*) appears in this field.
<i>uid_writeable</i>	A flag set and used by RealServer: 0 playerID is in database 1 record created, but playerID is not yet registered

The second and subsequent lines of each file have the following form (for further detail on allowable values in each field, see database structure later in this chapter):

```
url:url_type;permission_type;expires;debitted_time
```

where:

<i>url</i>	URL of secure directory or clip.
<i>url_type</i>	Whether URL is directory or clip: 0 clip 1 directory.
<i>permission_type</i>	Permission type associated with access. See “Permission Types” table on page 230 for values.
<i>expires</i>	If <i>permission_type</i> is 1, this is the expiration date/time, in format MM/DD/YYYY:HH:MM:SS. Otherwise blank.
<i>debitted_time</i>	If <i>permission_type</i> is 2, this is time remaining (in seconds). If <i>permission_type</i> is 3, this is the number of seconds of material the visitor has viewed. Otherwise blank.

This example file, *user1*, has the following content, when player validation is in use:

```

*;00001d00-0901-11d1-8b06-00a024406d59;0
Secure/clip1.rm;0;0;*;*
Secure/directory;1;0;*;*
Secure/time.rm;0;2;*;*;300;*
Secure/time.rm;0;1;05/24/1970:06:12:32;300;*

```

Note

If you manually edit the files, be sure that any blank (or unused) fields use an asterisk (*) as a placeholder. Do not use a space for a placeholder.

Guids Directory

The files in this directory are given the names of the unique client IDs from the registered clients, one per registered user. Each file contains only the name of the associated user name. For example, a file such as 00001d00-0901-11d1-8b06-00a024406d59 contains the name of the user, user1.

Logs Directory

This directory contains two files: reglog.txt and accesslog.txt.

Reglog.txt

Each line of reglog.txt represents the result of an attempt to register a visitor. This file has the following format:

```
status;userid;uuid;IP;register_time;url_redirect
```

where:

<i>status</i>	Result of user's attempt to connect: 0 Success 1 Failed (clientID not readable) 2 Failed (clientID already used) 3 Failed (RealAudio Player 3 or older) 4 No user (Must be entered previously in the database) 5 General failure
<i>userid</i>	Unique name of up to 50 characters.
<i>uuid</i>	Stores clientID.
<i>IP</i>	IP address from which user is attempting to connect.
<i>request_time</i>	Time of connection request.
<i>url_redirect</i>	If connection failed, URL to which user was redirected (see redirect.txt).

Accesslog.txt

Each line of `accesslog.txt` describes the result of an attempt to view a clip. This file is not created until authentication is enabled and the first user attempts to connect. Syntax of this file:

```
status;userid;uuid;ip:url;access_type;permission_on;start_time;end_time;total_time;
why_disconnect
```

where:

<i>status</i>	Result of user's attempt to connect: 0 access to clip granted 1 denied
<i>userid</i>	Unique name of up to 50 characters.
<i>uuid</i>	Stores playerID.
<i>ip</i>	IP address from which user is attempting to connect
<i>url</i>	Secured clip user is attempted to access.
<i>permission_type</i>	Permission type associated with access. See "Permissions Table" table for values.
<i>permission_on</i>	Permission type associated with url: 0 file (individual clip) 1 directory 2 none
<i>start_time</i>	Time/date clip started playing.
<i>end_time</i>	Time/date clip stopped playing.
<i>total_time</i>	Total time clip played.
<i>why_disconnect</i>	Reasons for disconnection: 0 client disconnected voluntarily 1 server access expired

Redirect Directory

Used only in player validation, the redirect directory contains files named after URLs that are restricted from unauthorized users. Within each file is the alternate URL to which RealServer sends the user if he or she tries to click the restricted URL. If no files are present in this directory, and the user attempts to click a URL to which he or she has not been given access, the user receives an error message.

Because certain characters that appear in URLs are illegal in file names, RealServer requires a substitution for these illegal symbols.

Substitutions

This character...	...is replaced with this sequence:
/	+2f
\	+2b
+	+5c

Thus, the URL “Secure/TopSecret.rm” would be converted to Secure+2fTopSecret.rm.

The URL within each file, however, is represented normally.

Using a Database for Authentication Data

This section describes the structure of the database templates included with RealServer.

To set up the database, see “Setting Up Other Types of Data Storage” on page 253.

The database templates include five tables:

- **Users table**—Together with the permissions table, contains the lists of who is registered and with what access.
- **Permissions table**—Linked to the users table, lists specific clips and directories and the permissions associated.
- **Register_log table**—Used if player validation is in use, it tracks the clientID.
- **Redirect table**—Used in player validation only.
- **Access_log table**—Used by the commerce feature.

Users Table

Gives the list of user names and passwords.

Users Table

Field	Description
<i>userid</i>	User name of up to 50 characters. Ties to permissions table.
<i>password</i>	In user authentication, this stores the password. Otherwise blank. Passwords are encrypted.

(Table Page 1 of 2)

Users Table (continued)

Field	Description
<i>uuid</i>	In player validation, stores clientID. In user authentication, an asterisk (*) appears in this field.
<i>uuid_writeable</i>	A flag set and used by RealServer: 0 clientID is in the database 1 the record has been created but the clientID is not yet registered with RealServer.

(Table Page 2 of 2)

Permissions Table

Linked to the users table via the *userid*, this identifies the specific clips or directories and the type of access for each.

Permissions Table

Field	Description
<i>userid</i>	User name of up to 50 characters. Ties to Users table.
<i>url</i>	URL of secure directory or clip.
<i>url_type</i>	Whether URL is directory or clip: 0 clip 1 directory.
<i>permission_type</i>	Permission type associated with access. See “Permission Types” table on page 230 for values.
<i>expires</i>	Permission expiration date and time, in format MM/DD/YYYY:HH:MM:SS. Used only if <i>permission_type</i> is 1 (dated). Otherwise blank.
<i>debitted_time</i>	If <i>permission_type</i> = 2 (countdown), this is the number of seconds remaining. If <i>permission_type</i> =3 (countup), this is the number of seconds of material the visitor has viewed. Otherwise blank.

Register_Log Table

The *register_log* table is only used if player validation is in use (indicated by *UseGUIDValidation=True*).

Register_log Table

Field	Description
<i>status</i>	Result of user's attempt to connect: 0 Success 1 Failed (clientID not readable) 2 Failed (clientID already used) 3 Failed (RealAudio Player 3 or older) 4 No user (Must be entered previously in the database) 5 General failure
<i>userid</i>	Unique name of up to 50 characters.
<i>uuid</i>	Stores clientID.
<i>ip</i>	IP address from which user is attempting to connect.
<i>request_time</i>	Time of connection request.
<i>url_redirect</i>	If connection failed, URL to which user was redirected (see Redirect Table, above).

Redirect Table

The redirect table is only used in player validation.

Redirect Table

Field	Description
<i>url</i>	URL of any secure clip or directory.
<i>url_redirect</i>	URL to which users could be redirected to if they are not allowed access to that clip. New URL must NOT be a secure URL.

Access_log Table

Used by the commerce feature to show which secure content has been accessed.

Access_log Table	
Field	Description
<i>status</i>	Result of user's attempt to connect: 0 access to clip granted 1 denied
<i>userid</i>	Unique name of up to 50 characters.
<i>uuid</i>	Stores player ID.
<i>ip</i>	IP address from which user is attempting to connect.
<i>url</i>	Secured clip user is attempted to access.
<i>permission_type</i>	Permission type associated with access. See "Permission Types" table on page 230 for values.
<i>permission_on</i>	Permission type associated with url: 0 file (individual clip) 1 directory 2 none
<i>start_time</i>	Time/date clip started playing.
<i>end_time</i>	Time/date clip stopped playing.
<i>total_time</i>	Total time clip played.
<i>why_disconnect</i>	Reason for disconnection: 0 client disconnected voluntarily 1 server access expired

Setting Up Other Types of Data Storage

Support for two types of databases is included.

- ▶ To set up your Windows computer for ODBC compliance:
 1. On the **Start** menu, point to **Settings**, and click **Control Panel**.
 2. Double-click **32bit ODBC**.
 3. On the **System DSN tab**, click **Add**.
 4. Select your ODBC driver from the list of drivers and click **Finish**.
 5. In the **ODBC SQL Server Setup** dialog box, type the data source name. Click **Select**.

6. Type or browse for the path to your database file and click **OK**.
7. Click **OK** to exit the ODBC Data Source Administrator.

You must now tell RealServer where to find your database.

► To set up the supplied database application on UNIX:

1. At a command line, start the database by typing the following:
`./mysql2d &`
2. Create the database by typing the following:
`./mysqladmin create databasename`
3. Note that whatever you type for *databasename* will need to match the database cited in the Databases list.
4. Create the tables using the database text file by typing the following:
`./mysql -h localhost databasename < ppvdemo.db`
Be sure to include the less-than sign (<).

Chapter 17

ISP HOSTING

ISP Hosting features provide a way to allot connections to users. If you are an Internet Service Provider (ISP), you can host streaming media on behalf of your customers.

Overview

RealServer works with your existing user accounts and directory structure to make users' media files available for streaming. You allocate a minimum and maximum number of connections for each account, based on the number of streams permitted by your license. Allocating on a per-connection basis, rather than by stream, ensures that all files, including SMIL files which reference multiple streams, will always be served.

User account information is stored in a text file, which lists pathing and connection information. List all user account information in a single file, or use separate files to make management easier. Within the user list file, create customized account path and connection information. Or, create a single entry that applies to all user accounts.

Take advantage of the RealServer 8 file system to store users' content in any directory, in any location.

Links to Users' Hosted Content

Links to hosted content have the following format if used in a Web page:

```
http://server.example.com/ramgen/~username/filename.rm
```

The link which RealServer uses, or which you can type directly into RealPlayer, has the following format:

```
rtp://server.example.com/~username/filename.rm
```

Account Information

When RealServer receives a request for streaming media, it looks at user account information, stored in user list files, to determine which user is hosting the requested content.

User list files can list account information separately for each user, or can give generic information that applies to all users.

Each account has three items associated with it:

- Account name
- Virtual path where the account's files will be stored
- Minimum and maximum connections available to the account

Account information is stored in text files, called user list files. You can put all information into a single file or use separate files to make organization easier.

Connections Available for Each Account

Each account has a reserved number of connections and a maximum number of connections associated with it. The user list file can contain a generic account description that applies to all users, specific instructions about certain accounts, or a combination of the two.

The maximum setting refers to the highest number of connections that will be available for a particular customer's content. Anyone who tries to watch a clip after that account's maximum number of connections are in use will receive an error message, even if connections are available to other accounts.

The number of connections reserved for ISP hosting depends on the type of user record within the user list file:

- Specific user account descriptions
- Generic user account description

If you use a combination of account descriptions, be sure to read both topics in this section.

Specific User Accounts

The reserved setting ensures that the specified number of connections will always be available to clients that attempt to view a particular user's hosted media.

All reserved connections are subtracted from the overall number of connections available to RealServer. The remaining connections are available

for non-ISP-hosted content, or for hosted content that hasn't yet been requested. For example, if your RealServer is licensed for 50 connections, and you reserve 20 connections through the ISP hosting feature, there are 30 connections available for general use. RealServer can use those remaining connections for streaming regular clips, or for users of ISP-hosted material that isn't yet reserved.

Reserved connections are only activated for accounts listed in the user files, and are activated as soon as RealServer starts.

Tip

To guarantee that connections will always be available for certain customers, list those account names in the user file, rather than using a generic scheme. Be careful to leave enough streams available for other use, however.

Users whose accounts are not specifically listed in the user list file default to the generic account description.

Generic User Account

For accounts not described in the user list file, minimum connections are not reserved until content is played from a user's account.

Other Considerations

It is possible to reserve more connections than are included in your license. In this case, connections are distributed on a first come, first served basis.

For example, if your RealServer is licensed for 50 connections, and you create a generic account that reserves a minimum of 3 connections for all 25 customers, all the connections will be reserved for ISP hosting customers. Since 75 connections are reserved, but only 50 connections are available, the first 50 customers who connect will be able to play content, but anyone connecting after that will not.

ISP Hosting Used with Other Features

Users inherit many features of your RealServer: hosting of on-demand content, and access control are available.

Authentication of users' content, and features related to live material, such as broadcasting of live content, splitting, and multicasting, are not available for hosted material.

As the administrator, you are able to view how many clients are connected to all material served by your RealServer, using Java Monitor. In order to discern which material belongs to users, you must examine the paths of the individual clips in use. You also can see which clips have been served by reading the access log file.

RealProxy is able to cache your users' content, just as it can cache any on-demand files served by this RealServer.

Tracking Account Usage

Like any content it serves, RealServer creates a record for each file it serves in the access log. The fourth field in each record of the access log, identified by the GET statement, lists the path and filename of each clip served. Compare this information to the path information you've set up to determine how many clips have been streamed from each account.

In most cases, RealServer creates one record for each clip served. However, SMIL presentations served from your clients' accounts may generate more than one record. You can see which records are part of a SMIL presentation by looking at the final number in the record (present if Logging Style is 5). These numbers will match if they are from the same SMIL presentation. See "SMIL Presentations, Ram Files, and Access Log Files" on page 286.

Account-Based ISP Hosting

The GET statement will include the ISP hosting mount point and the user's account name (beginning with the ~ character). For example, a file with the URL:

```
http://server.example.com/ramgen/~chris/file.rm
```

would appear in the access log as:

```
GET ~chris/file.rm
```

Dedicated ISP Hosting

Because dedicated ISP hosting RealServers can only stream content for users, and not stream any other type of content, the access log will only show material streamed for ISP customers. The mount point will always appear.

The GET statements will show the directory portion of the URL.

A file with the following URL:

```
http://server.example.com:8080/ramgen/r/ra/rabrams/file.rm
```

would appear in the access log as:

```
GET r/ra/rabrams/file.rm
```

Dedicating RealServer to ISP Hosting

RealServer can be dedicated to only serving hosted content. If you use this option, RealServer cannot stream media files from any other directories.

This option requires that users' directories are arranged in a hierarchy. Features available in dedicated hosting are the same as in account-based hosting.

URLs used in this type of hosting have a different format. Rather than use a tilde (~) to alert RealServer to an upcoming ISP request, this method relies on a directory structure shown in the URL.

```
http://server.example.com/ramgen/s/sa/sandy/media/filename.rm
```

or

```
rtsp://server.example.com/s/sa/sandy/media/filename.rm
```

A comparison of the two styles is shown below. Use only one style on a particular RealServer.

Comparison of Account Identification Styles

Issue	Account-Based Hosting	Dedicated Hosting
Hosted material	Can host content for ISP users; can also serve ordinary streamed content.	Can only host content from user accounts. Cannot serve other content.
User directory structure	Works with any directory structure; enables different structures or locations. Users may have their own subdirectories.	Works with a hierarchical directory structure, especially an alphabetic one. Organization of directories must be the same for all users. Users may have their own subdirectories.
Reserving connections	Can reserve number of connections available for material streamed from certain accounts.	Cannot guarantee any reserved connections.
User settings	Some users can have customized settings, while generic connection settings describe all other users.	All users have identical settings.

Compatibility with Earlier Versions of RealServer

If you used ISP Hosting in RealServer versions 3 through 5, you can still use the `UserList` from your previous configuration file. Refer to “Creating User Lists From Earlier Versions” on page 267 for instructions on how to use your existing `UserList`.

Earlier versions of RealServer listed minimum and maximum settings for the number of streams available to each account. In RealServer version 8, those settings now refer to the number of connections available to each account. This enables customers to serve SMIL files—which may reference several streams simultaneously—without running out of streams.

This manual uses new terminology for the methods of referring to account structures described in previous editions of *RealServer Administration Guide*.

- “Naming Convention One” is now described as the usual method of configuring user list files.
- “Naming Convention Two” is described here as a dedicated hosting RealServer, a special case.

Although they have different names in this manual, the user directory structures and user list structures in each method are functionally identical to the methods used in earlier versions of RealServer.

Example ISP Hosting Scenario—Northwest ISP

Throughout this chapter, we’ll use the example of an ISP who sets up RealServer to host its users’ media files. Northwest ISP hosts content for customers in a three-state area in the United States’ Pacific Northwest. Users’ directories are organized according to the state in which the users live—Washington, Oregon, and Idaho:

C:\home\washington

C:\home\oregon

C:\home\idaho

Individual accounts are located immediately below these directories:

Chris Anderson’s account: C:\home\washington\canderson

Pat Brown’s account: C:\home\washington\pbrown

Lee Adams’ account: C:\home\oregon\ladams

Sandy Chu’s account: C:\home\oregon\schu

```
Other accounts:      C:\home\idaho\alex
                    C:\home\idaho\sam
                    C:\home\idaho\tracy
```

The links to these users' files look different than other RealServer links. These all contain a tilde (~) and the user's usernames or account names:

```
http://server.example.com:8080/ramgen/~chris/file.rm
http://server.example.com:8080/ramgen/~lee/file.rm
http://server.example.com:8080/ramgen/~pat/file.rm
http://server.example.com:8080/ramgen/~sandy/file.rm
```

Users' Directory Structures

RealServer matches your existing directory locations of users' files, even if you use different structures for different users. Typically, user directories are named with the username of the account; the username is included in the URL.

Customers' media files are stored in their directories. If they place files in a subdirectory of their main directory, that subdirectory must be included in the URL.

Directory Structures in Dedicated Hosting

A RealServer used exclusively for hosting users' streamed media from accounts based on a strict directory structure uses an alternate method of identifying accounts. In the user list, you identify how far down the directory path to look for individual user accounts; this requires that the accounts must all be at the same level.

In the following example, accounts are divided into separate directories, according to an alphabetic arrangement:

```
...
/UserAccounts/r/ra/rabrams
/UserAccounts/r/ra/radams
...
/UserAccounts/s/sa/sanderson
/UserAccounts/s/sb/sbraun
/UserAccounts/s/sb/sbrown
/UserAccounts/s/sc/schu
...
```

Users may have their own subdirectories. If they place files in a subdirectory of their main directory, that subdirectory must be included in the URL.

Of course, if you use the account-based style of identifying customer directories rather than the method described in this section, you can also dedicate RealServer to only hosting streamed media for customers, but other streaming options are still available.

Setting Up ISP Hosting

There are three steps for configuring RealServer to host users' media files:

1. Create the user list file.

This file establishes account information, such as reserved connections and maximum connections.

Additional Information

See "Step 1: Creating the User List" on page 262.

2. Configure RealServer.

The configuration file indicates where to find the user lists, and completes the pathing information needed to locate the users' media.

Additional Information

See "Step 2: Configuring RealServer" beginning on page 267.

3. Creating the links to content.

Additional Information

See "Step 3: Linking to ISP Content" on page 269.

You will need to tell customers what format they should use in creating their links.

Step 1: Creating the User List

Create the user list, and store it anywhere that is accessible to RealServer.

The user list is a text file with the following format:

```
UserList [  
{account, /path/, minimum_connections, maximum_connections}  
]
```

where:

account is either a specific user name, or *~** to indicate that all accounts will use the same settings. See “Using Multiple User List Files” on page 265 for examples of how multiple accounts can be shown in a user list.

Note

Dedicated hosting RealServers use a slightly different format. Refer to “Dedicated Hosting User File Format” on page 266 for the correct format to use.

/path/ gives information about the location of users’ media files. It does not necessarily refer to an actual location or portion of a location; instead, it is a logical method of grouping the users.

minimum_connections is the minimum number of connections reserved for this user. 0 indicates that no connections are reserved. See “Connections Available for Each Account” on page 256 for more information.

maximum_connections is the maximum number of connections available to this user. 0 indicates that no connections may be used. See “Connections Available for Each Account” on page 256 for more information.

Tip

You can include comments in the file by preceding a line with a semi-colon (;).

Example—User List File

In this user list file (shown in the left column of the table), users are grouped according to their geographic location. Two users, Chris and Pat, are in the Washington (wa) group. Two other users, Lee and Sandy, are in the Oregon group.

User List File Contents	Matching Customer Name
UserList [{chris, /wa/canderson/, 2, 5}, {lee, /or/ladams/, 0, 100}, {pat, /wa/pbrown/, 2, 50}, {sandy, /or/schu/, 1, 35},]	Chris Anderson Lee Adams Pat Brown Sandy Chu

Listing Individual Accounts

If each account has different settings, create a separate record for each user, as in the example above.

Listing Generic Accounts

If you have a large number of accounts to create, and they will all use the same number of connections, create a single entry that refers to all accounts generically:

```
UserList [  
  {~*, /path/, minimum_connections, maximum_connections}  
]
```

In the following example, one connection is reserved for each person, and the maximum number of connections available for any account is 35. (There are some restrictions on whether the connections are actually reserved; see “Connections Available for Each Account” on page 256.)

```
UserList [  
  {~*, /users/, 1, 35}  
]
```

Combining Individual Account Listings with a Generic Listing

Custom account information and generic settings can be combined in a single user list. Combining them is convenient if most users have the same settings, but a few have different number of connections reserved, or use different paths:

```
UserList [  
  {username1, /path/, minimum_connections, maximum_connections}  
  {username2, /path/, minimum_connections, maximum_connections}  
  {username3, /path/, minimum_connections, maximum_connections}  
  ...  
  {~*, /path/, minimum_connections, maximum_connections}  
]
```

In the following example, customized accounts for four users have been created, and all other accounts will use the default settings shown in the last entry:

Sample User List	
User List	Customer Name
UserList [
{chris, /wa/canderson/, 2, 5},	Chris Anderson
{lee, /or/ladams/, 0, 100},	Lee Adams
{pat, /wa/pbrown/, 1, 35},	Pat Brown
{sandy, /or/schu/, 1, 5},	Sandy Chu
{~*, /id/, 1, 35}	All others not specified above
]	

Using Multiple User List Files

You can create as many user lists as you want; using multiple files can make administration easier. For example, an ISP provider might include commercial accounts in one user list file and personal accounts in another file.

RealServer loads the user lists in the order they appear in the configuration file, and any settings in subsequent files override settings in previously-loaded files. If the same user name appears in more than one list, RealServer uses the settings in the last user list.

Because of this behavior, bear in mind the following considerations when using multiple user list files:

- An account name must not appear in more than one file.
- The generic account information (an entry beginning with ~*) must be used carefully. Include it only in the first-loaded user list file. If you include it in the last file, RealServer will ignore all the other user lists.

Re-Reading an Updated User List File

Once you have created the user list file and the ISP hosting feature is in use, you must instruct RealServer to re-read the user list.

- On Windows-based platforms, after you edit a user list file, you must restart RealServer for the changes to take effect.

- On UNIX-based platforms, you can use the SIGHUP command to instruct RealServer to re-read the user list files. See “SIGHUP” on page 116.

Dedicated Hosting User File Format

The format of the user list file in dedicated hosting is nearly the same as the account-based method, with these exceptions:

- Create only one user file. You cannot use more than one with the dedicated hosting server.
- Create only one entry in the user file. This entry applies to all user accounts.
- Instead of giving an account name, you indicate how far to traverse the user directory structure in order to find the unique user directories.

Use the following format:

```
UserList [
{*n, /path/, minimum_connections, maximum_connections}
]
```

where *n* is a number that represents the level of directory at which individual user directories appear.

Example

In the following example, all user accounts are located under a subdirectory of the UserAccounts directory. The unique directories are located at the fourth directory level (rabrams, radams, sanderson, and so on).

```
...
/UserAccounts/r/ra/rabrams
/UserAccounts/r/ra/radams
...
/UserAccounts/s/sa/sanderson
/UserAccounts/s/sb/sbraun
/UserAccounts/s/sb/sbrown
/UserAccounts/s/sc/schu
...
```

The user list for this example uses 4 for the value of *n*:

```
UserList [
{*4, /UserAccounts/, 1, 15}
]
```

URLS created with this method have the following format:

```
rtsp://server.example.com:554/directory1/directory2/directory3/filename
```

For example,

```
rtsp://server.example.com:554/UserAccounts/r/ra/rabrums/band.rm
```

Creating User Lists From Earlier Versions

Recycle your `UserList` entry from the configuration file of earlier versions. If your `UserList` is long, you may want to create more than one file.

After you create the new user list file, follow the instructions in “Step 2: Configuring RealServer”.

- ▶ To create a user list from existing settings:
 1. Open your old configuration file in a text editor.
 2. Locate the `UserList` entry.
 3. Copy and paste the existing `UserList` setting into a new text file.
 4. Save the file. You can store it in any directory that is available to RealServer.

Another item from the previous configuration file, `UserDir`, does not have an equivalent.

Step 2: Configuring RealServer

You will need to make a note of the values for `/path/` that you used in the user list file.

These instructions describe how to create a separate mount point for each customer category, which means customer files can be stored in separate base paths or drives.

- ▶ To configure RealServer for ISP Hosting:
 1. First, look in the user list files at the `/path/` settings you have used. You will need this information in Step 15.
 2. In RealSystem Administrator, click **General Setup**. Click **Mount Points**. You will add a mount point for each path in the User List file, give a description, and indicate a base path.
 3. Click **Add New**.
A generic mount point name appears in the Edit Mount Point box.
 4. In the **Edit Mount Point** box, type a name for the new mount point.

In our example, type `/wa_isp/`.

5. Click **Edit**.
6. In the **Description** box, type a description for this mount point.
7. In the **Base Path** box, type the location in which these paths should be mapped.

In our example, type `C:\home\washington`.

8. Repeat Step 3 through Step 7 for each mount point and base path combination.

In our example, we used the following settings:

Example Settings

Mount Point	Description	Base Path
<code>/wa_isp/</code>	ISP Content (Washington users)	<code>C:\home\washington</code>
<code>/or_isp/</code>	ISP Content (Oregon users)	<code>C:\home\oregon</code>
<code>/id_isp/</code>	ISP Content (Idaho users)	<code>C:\home\idaho</code>

9. Click **Apply**.
10. In the left-hand pane of RealSystem Administrator, click **General Setup**. Click **ISP Hosting**.
11. In the **Translation Mounts** area, click **Add New**.
A generic translation mount appears in the Edit Translation Mount Description box.
12. In the **Edit Translation Mount Description** box, type a description for this Translation Mount.
13. Click **Edit**.
14. From the **Mount Points** list, select the mount point that you want to use for this Translation Mount. (You created these in Step 2 through Step 8.)
15. In the **User Path** box, type the value of `/path/` from the user list file.
For each `/path/` that appears in the user list file, repeat Step 11 through Step 15 to associate the `/path/` with a translation mount.

In our example, we have created a separate user path for /wa/, for /or/, and for /id/.

Example User Paths

Translation Mount Description	Mount Point	User Path
Washington users	/wa_isp/	/wa/
Oregon users	/or_isp/	/or/
Idaho users	/id_isp/	/id/

16. In the User List files section, click **Add New**.
A generic user list name appears in the Edit User List File Name box.
17. Type the correct path to the user list you created in “Step 1: Creating the User List” on page 262. Be sure to give the full path.
18. Click **Edit**.
19. To add more than one user list, repeat Step 16 through Step 18 for each list you want to add.
In dedicated hosting, reference only one user list file.
20. Click **Apply**.

Step 3: Linking to ISP Content

You’ll need to tell your customers what format to use for their links.

Links in a Web page use this format:

`http://address:HTTPPort/ramgen/~account/path/file`

RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used to initiate streaming.
<code>address</code>	Machine and domain name of RealServer. IP address may be substituted.
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via the protocol listed at the beginning of the URL. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<code>ramgen</code>	Required when you link in a Web page.
<code>account</code>	User’s account name.

(Table Page 1 of 2)

RealServer URL Components (continued)

Component	Meaning
<i>ramgen</i>	The mount point tells RealServer how the clip should be served.
<i>path</i>	Optional.
<i>file</i>	The file name itself, including the extension.

(Table Page 2 of 2)

For samples of links to use in the Web page, see “Example ISP Hosting Scenario—Northwest ISP” on page 260.

Links typed directly in RealPlayer or used in a Ram or SMIL file use the following format:

```
rtsp://address:RTSPPort/~account/path/file
```

The format is nearly the same as the link used in the Web page: the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

Dedicated Hosting Server

Links in a Web page use this format:

```
http://address:HTTPPort/ramgen/directory1/directory2/path/file
```

where:

RealServer URL Components

Component	Meaning
<i>http</i>	The protocol used to initiate streaming.
<i>address</i>	Machine and domain name of RealServer. IP address may be substituted.
<i>HTTPPort</i>	Port number where RealServer listens for requests sent via the protocol listed at the beginning of the URL. This value is usually 80 or 8080; see “Port Numbers” on page 95.
<i>ramgen</i>	The mount point tells RealServer how the clip should be served.
<i>directory1</i>	Each directory that is part of the hierarchy of directories. The number of directories you list must match the <i>n</i> number in the user list file.
<i>directory2</i>	

(Table Page 1 of 2)

RealServer URL Components (continued)

Component	Meaning
<i>path</i>	Optional. Represents any subdirectories of the user's home directory.
<i>file</i>	The file name itself, including the extension.

(Table Page 2 of 2)

Using the example in “Dedicated Hosting User File Format” on page 266, a link to file.rm in the user directory /UserAccounts/r/ra/rabrams would look like the following:

`http://server.example.com:8080/ramgen/r/ra/rabrams/file.rm`

Chapter 18

MONITORING REALSERVER ACTIVITY

To manage current activity on your RealServer, you'll want to track things such as which clips are most popular, what the stream load is, and whether viewers are being turned away. RealServer includes the following methods for monitoring real-time activity: Java Monitor and NT Performance Monitor (for Windows NT users). To generate reports of historical activity, see Chapter 19, "Reporting RealServer Activity".

Java Monitor

Included with RealSystem Administrator is a configurable graph that displays real-time information about the number of clients connected to RealServer, resources used, and which files are being streamed.

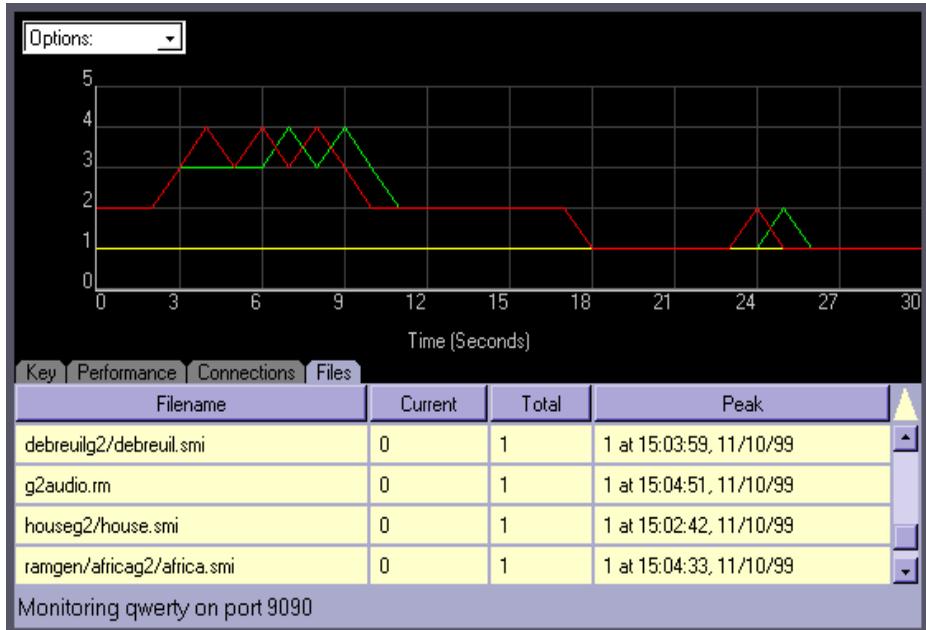
RealSystem Administrator includes a real-time Java Monitor to show activity on your RealServer, making Server management easy. It shows who is using the Server, when it is most used, and which files are the most requested, as well as other information.

Use feedback from Java Monitor to:

- Respond to customer demand
- Manage content and change the Server configuration remotely
- Make more informed business decisions

You can also create other external Java Monitors to track more than one Server, monitoring multiple RealServers side by side. A brief status message displays along the bottom of each window, telling you which Server is being monitored.

Java Monitor in RealSystem Administrator



Java Monitor Used with Other Features

Java Monitor displays all on-demand and live presentations that are currently being streamed or broadcast. It does not differentiate among the delivery methods—whether streaming, unicasting, splitting, or multicasting.

Live Archiving and Java Monitor

Java Monitor does not indicate whether live files are being archived.

G2SLTA and Java Monitor

Java Monitor does not distinguish the source of a clip; thus it never shows whether a broadcast is coming from an event in progress or **G2SLTA**.

Splitting and Java Monitor

On the transmitter, no connections are displayed in Java Monitor. On the receiver, the split connection will appear under the Connections tab as an encoder. For example, the following text appears in the Filename column:
realserver.example.com/encoder/filename.rm

Multicasting and Java Monitor

Java Monitor can show clients that are receiving back-channel multicasts, just as it shows clients receiving any other type of broadcast or stream. However, it will not show the number of clients receiving scalable multicasts.

Using Java Monitor

Start Java Monitor and you can immediately view the activity on your RealServer.

► To start Java Monitor:

In RealSystem Administrator, click **Monitor**. Java Monitor appears. You can make selections from several places in Java Monitor.

Configuring Java Monitor Settings

Java Monitor uses just two variables from RealServer: Monitor Port and Monitor Password. You don't need to change these values, unless you want to use values which are not default settings.

► To change Java Monitor Settings:

1. In RealSystem Administrator, under **Configure**, click **General Setup**. Click **Ports**.
2. In the **Monitor Port** box, type the port number for the Java Monitor to use in connecting to RealServer. The default value for MonitorPort is 9090.
3. Click **Apply**.

The password which Java Monitor uses to connect to RealServer is stored in the MonitorPassword variable. This value is set during installation, but you can change it at any time.

This value must be changed by directly modifying the configuration file. See Chapter C, "Configuration File Contents" for instructions.

Once you have changed one or both values, RealSystem Administrator will automatically use the new values when displaying the Java Monitor.

Optional Java Monitor Features

There are several ways you can control what Java Monitor displays. This section describes the commands present on the Java Monitor display area and their functions.

Options Menu

Select the drop-down **Options** menu in the upper left hand corner of Java Monitor to configure the Monitor's features or spawn an external Monitor which runs outside the browser.

Options Menu Commands

Command	Effect
New Window	Create a new, external monitor. You can then minimize the browser and resize the new monitor.
Pause	Freezes the graph. Java Monitor continues to receive data, but the graphical display of data does not change. Click Resume from Options to resume the graphing.
Reset	Clears the graph and resets all peak data.
Configure	Displays the configuration screen. Specify the update frequency in seconds, the time scale in minutes, and select which statistics to monitor.
Autofit	Rescales the graph so that it fits within the viewable area. Note: Whenever you zoom, the Autofit feature is disabled. Select AutoFit from the Options menu to re-enable AutoFit .
Zoom In	Zoom in on the graph. Use the mouse to select a range over the graph to zoom in for a closer view. Tip: Hold down the CTRL key on your keyboard, and click the mouse to Autofit the graph.
Zoom Out	Zoom out from the graph.

Tabs

The **Key**, **Performance**, **Connections**, and **Files** tabs each have a specific focus, providing you with an overall picture of server performance.

Tip

Clicking the active tab will expand or collapse the tab information and show only the tab name, leaving more room for the monitor. To show the contents of the tab again, click the tab name again.

Key Tab

The **Key** tab shows how RealServer information is graphed. By clicking different options in the Line column, you can control what colors and line

widths are used to display RealServer information (see instructions below the table).

Key Tab Columns

Column	Purpose
Line	Controls line display: width, color, and order.
Name	Type of item being monitored: Players, Monitors, Encoders, Files, and Receivers.
Current	Shows the number of the current connections.
Peak	Shows the peak numbers of files monitored, and time and date.

► **To control line width:**

In the row that contains the information whose line width you want to modify, click within the line box itself to toggle among three possible line widths.

► **To change line color:**

Click the up and down arrows within the line box, to cycle among the 16 possible colors for the line.

► **To change line display order:**

Click on the left hand arrow within the line box, to change the drawing order of the lines, which will move the line and name of item being monitored up one row.

Performance Tab

The **Performance** tab provides statistics on RealServer performance.

Performance Tab Columns

Column	Purpose
CPU Usage	Displays current central processor unit (CPU) usage (as percentage of overall CPU usage).
Memory Usage	Displays system's Memory Usage (in kilobytes).
Bandwidth	Displays the amount of data being sent (in kilobits per second).
Players Connected	Displays the number of RealPlayers connected.
File Usage	Displays the number of files being served.

Connections Tab

This tab provides background on connected clients and the files they are accessing.

Connections Tab Columns

Column	Purpose
IP Address	RealPlayer's host Internet Protocol (IP) address.
Type	Type of browser or RealPlayer.
Duration	Amount of time the client has been connected.
Filename	Name of the file being served.

Files Tab

The files tab provides statistics on all files being served.

File Tab Column

Column	Purpose
Filename	Name of the file being served.
Current	Number of current clients connected.
Total	Total number of times a file was served during this monitoring session.
Peak	Shows the peak numbers of files monitored, and time and date.

Java Monitor Modes

The Java Monitor can run as an applet or application. When you select **New Window** from the **Options** menu, the new Java Monitor runs as an applet. Another method is available for running Java Monitor as a separate application.

Review the considerations below before choosing which mode you want the Java Monitor to use.

Applet Mode Considerations

- Can be run from inside a Web browser.
- Can be run from any remote machine with a Java-enabled browser.
- Settings may not be saved when you switch among the RealSystem Administrator's Web pages.

- Developers can use a scripting language and the parameters below to customize the Java Monitor applet to their specifications.

Applet Parameters

Parameter	Possible Values	Default Value
dragZoom	enabled, disabled	enabled
viewPanel	keyPanel, resourcePanel, clientPanel, filePanel, minimized, disabled	keyPanel
StatusBar	enabled, disabled	enabled
PlayerCount	enabled, disabled	enabled
FileCount	enabled, disabled	enabled
EncoderCount	enabled, disabled	enabled
MonitorCount	enabled, disabled	enabled
SplitterCount	enabled, disabled	disabled

- To run Java Monitor in applet mode:

Applet mode is the default method for Java Monitor when you click **New Window** from the **Options** menu.

Application Mode Considerations

- No Web browser needed.
- Can switch among different servers without spawning new windows.
- Java class files, available for free download from Sun, must be installed on the local machine. They are described below.

- To run Java Monitor in application mode:

1. Download and install version 1.1 of the Java Development Kit, available as a free download from Sun's Web site: <http://java.sun.com/j2se/>. Follow the installation instructions on the Web site to install the Java Development Kit on your system.
2. Change to the directory where the newly installed Java class files are located. Change to the Bin subdirectory.
3. At a system prompt, type the following:

```
jre -cp Monitor.jar Monitor
```

The Monitor and a logon screen appear.
4. In the logon screen, type the following items:

- **RealServer name**—use the IP address or host name of the machine on which RealServer is installed.
 - **Monitor Port**—you can find this number by clicking **General Setup>Ports** in RealSystem Administrator.
 - **Monitor Password**—to find the password, look in the configuration file for the MonitorPassword variable. See “How do I look up my user name and password?” on page 338.
5. Click **OK**.
 6. Java Monitor starts.

Windows NT Performance Monitor

RealServer is designed to work with the Windows NT Performance Monitor to show activity on one or more RealServers. This option is available if you are running the RealServer on Windows NT and are viewing it from that same computer. A Performance Monitor file containing the RealServer statistics, `rmserver.pmc`, is supplied.

You can also configure the Performance Monitor to show RealServer status from any computer on your network. The Performance Monitor can show the following types of information:

- **Clients and protocol**—The number of active clients. Also shown are the protocols used by the clients to receive streams.
- **Connection type**—The number and type of connections, whether TCP or UDP.
- **Multicast connections**—The number of active multicast connections.
- **Total bandwidth**—The number of bits per second being consumed.
- **Percent of processor**—How much processor time RealServer is using.
- **Connections**—How many encoders, monitors, and receivers are connected.
- **Incoming bandwidth**—Bandwidth of streams arriving from encoders.
- **Files playing**—Number of files playing, including all the files in a SMIL presentation. Live files are also shown.
- **Files archiving**—Number of live files being saved.

Using the NT Performance Monitor, any combination of this information can be displayed in any of the following formats:

- A chart that graphs activity over time
- Alerts that notify the administrator via e-mail or run programs based on criteria
- Log files that list activity on RealServer
- Reports based on activity information

For information on configuring these formats, see the online help in NT Performance Monitor.

Chapter 19

REPORTING REALSERVER ACTIVITY

RealServer can create reports of historical data that let you see trends and gather information. Track who visited your site and for how long; what clips they watched and whether they watched them all the way through to completion. This information is stored in the access log. Any error messages are recorded in the error log. Requests for streams which will be cached are stored in the cached requests log.

Access Log

The RealServer access log records the IP addresses of the clients that have connected, the clips they listened to, the times of day they connected, and much more. This information can give you an idea of who your audience is and which clips are most popular. New information is always appended to the end of the access log.

Access Log Files Used with Other Features

This section describes the ways in which other features show up in the access log file.

Number of Records Created for Each Clip

The GET statement in the access log (described on “Access Log Format” on page 289) shows the names of the on-demand and live clips served by RealServer. Most clips generate one access log record apiece.

Clips delivered via scalable multicasting generate two records for each client that connects—one for the .sdp file and one for the actual live broadcast clip. (However, if the user saves the .sdp file and connects via that file, rather than by clicking a link on a Web page, only the live broadcast clip will generate a record.)

A record is generated for a SMIL file and for every file referenced in it. You can identify which files are associated because they will all have the same identifier at the end of the access log. (This identifier will only appear when Logging Style is 5.)

On-Demand Streaming and Access Log Files

On-demand clips appear in the access log with all the expected information—clip path and name, and statistics, if specified by the Stats Mask and Logging Style options.

Live Unicasting and Access Log Files

Client data for live events is transmitted at the conclusion of the broadcast. Entries will not appear in the access log until the client stops playing the event—which could be when the live event is over or if the user clicks the Stop button.

The GET statement shows unicast events starting with the live mount point (usually /encoder/).

Statistics Type 3, which show the user's actions during play (such as fast forward and pause), are not available for live events.

G2SLTA and Access Log Files

Client data for live events is transmitted at the conclusion of the broadcast. Entries will not appear in the access log until the client stops playing the event—which could be when the live event is over or if the user clicks the Stop button.

If you set up **G2SLTA** to do an infinite loop, and the client remains connected, no record will be created until the broadcast stops or the client halts.

The GET statement shows unicast events starting with the live mount point (usually /encoder/).

Statistics Type 3, which show the user's actions during play (such as fast forward and pause), are not available for live events.

Splitting and Access Log Files

On transmitters, the access log does not show any records pertaining to the receiver connections. However, if the same event is encoded to multiple RealServers, (described in “Using Backup Transmitters” on page 172), records will be created in the transmitter's access log.

On receivers, the access log contains records for each clip delivered, and shows the splitting mount point.

Client data for live events is transmitted at the conclusion of the broadcast. Entries will not appear in the access log until the client stops playing the event—which could be when the live event is over or if the user clicks the Stop button.

If backup transmitters are in use for push splitting (the link contains an asterisk), the access log on the transmitter will show the IP address of the transmitter where the broadcast came from, rather than an asterisk.

Multicasting and Access Log Files

Back-Channel Multicasts

Clips which were broadcast using back-channel multicasts can be identified with the *protocol* statement, which will be either PNAM or RTSPM. The same clip, delivered via unicast, will show PNAT or RTSPT if the TCP transport was used, or PNA or RTSP if UDP was used.

Scalable Multicasts

Client data for live events is transmitted at the conclusion of the broadcast. Entries will not appear in the access log until the client stops playing the event—which could be when the live event is over or if the user clicks the Stop button. In scalable multicasts, which can reach tens of thousands of clients, this volume of client data can overwhelm RealServer. The optional Send Client Statistics feature instructs clients to send their data to a Web server, which may be better equipped to handle the large quantities of HTTP posts. See “Controlling Client Statistics” on page 206 for instructions on configuring this feature.

A scalable multicast broadcast will create either one record or two for each client that connects. The number of records generated depends on whether Send Client Statistics is in use.

If Send Client Statistics is set to True, two records are created for each client that connects to a scalable multicast. The first record is created when the user clicks the link to the .sdp file. The .sdp file generates a request for the actual live file, which appears in the second record. This second record is created at the end of the multicast, or when the user clicks the Stop button.

If Send Client Statistics is set to False, only one record appears in the access log. The .sdp file, which handled the initial request, will appear in the log. No record is created for the live file.

Access Control, Authentication, and Access Log Files

The access log does not show whether access control rules are in use. Only clients whose IP addresses were approved by the access control rules, and who supplied the proper name and password (if required) are allowed to receive content.

Authenticated content is identified by the /secure/ mount point in the path shown in the GET statement.

ISP Hosting and Access Log Files

You can identify which on-demand files were served by the ISP hosting feature by comparing the filename in the GET statement to the /path/ value in the user list file.

Monitoring and Access Log Files

The Java Monitor shows files that are being viewed presently; the access log provides a historical report of all the files that have been served. All the files that Java Monitor shows will appear in the access log when they finish playing.

RealSystem Administrator and Access Log Files

The access log file shows all files served by RealServer, including all RealSystem Administrator Web pages. These appear in the GET statement; you can easily identify them because they all begin with admin. For example, "GET admin/index.html HTTP 1.0" shows the opening RealSystem Administrator page. If you make changes using RealSystem Administrator, the confirmation page that appears in RealSystem Administrator is also recorded in the access log.

When Logging Style is 5, a number at the end of each record gives *presentation_id*. For RealSystem Administrator pages, this number associates the elements on a particular page. All the images that go with each page also appear in the access log. All files served that are related to a particular page are numbered sequentially.

SMIL Presentations, Ram Files, and Access Log Files

When Logging Style is 5, the *presentation_id* field assigns the same number to all files that were delivered as part of the same presentation—whether via

SMIL file or Ram file. The numbers are generated by RealServer in sequence, restarting at 0 when RealServer restarts.

SMIL Files

Each file referenced by a SMIL presentation, including the SMIL file itself, generates a record in the access log. When Logging Style is 5, all files referenced by the SMIL file, as well as the SMIL file itself, will have the same number. For example, the sample files `presentation.smi`, `presentation.rt`, `presentation.rp`, and `presentation.rm` all have the same number in the `presentation_id` field, such as 432.

Note

If the SMIL file was requested via a link that used Ramgen in the URL, an additional record is created for the Ramgen statement, and shows a different value for the `presentation_id` field.

Ram Files

All the files referenced by the Ram file will each generate a record in the access log. Because Ram files are served by a Web server, and not RealServer, there is no record created in the access log for the Ram file itself.

When Logging Style is 5, all the files referenced by a Ram file will have the same `presentation_id` number.

Reading an Access Log

To read the contents of the access log, you must first look up the values of Logging Style and Stats Mask, as these determine how much information is present in the access log. Use RealSystem Administrator to find out the values for these variables by clicking **General Setup>Logging**. At installation, Logging Style is set to 5 and Stats Mask is 3.

Logging Style provides information about RealServer clip-serving activity. Client information is provided by Stats Mask. However, clients have the ability to prevent some statistics (Stat1, Stat2, and Stat3) from appearing in the access log. If this option is selected in the client, UNKNOWN appears in place of that statistics field.

Once you know the values of these two variables, view the access log by opening `rmaccess.log` (Windows) or `rmaccess` (UNIX) file in a word processor or text editor.

Note

Information on which authenticated files have been accessed is stored in `reglog.txt` and `accesslog.txt`. See “Logs Directory” on page 248.

Access Log Format

RealServer stores information about each clip it serves in a separate record. Each record is delimited by a new line. Fields within each record are separated by spaces.

One record is created for every clip served; if the client requests a presentation that includes several clips, one record is created for each clip in the presentation.

The fields that appear within each record depend on the settings for Logging Style and Stats Mask (these are noted in the “Access Log Format” table below). The complete syntax of each record, assuming Logging Style and Stats Mask are gathering all possible information (Logging Style is 5 and Stats Mask is 7) is shown:

```
client_IP_address - - [timestamp] "GET filename protocol/version" HTTP_error_code
bytes_sent [client_info] [client_GUID] [Stat1:][Stat2:][Stat3:] file_size file_time
sent_time resends failed_resends [stream_components] start_time server_address
average_bitrate packets_sent presentation_id
```

The optional `[Stat1:]`, `[Stat2:]`, and `[Stat3:]` fields, which are the result of the `StatsMask` variable, are described in greater detail in separate tables.

Note

Although in the rest of this manual, square brackets indicate optional material, the square brackets shown in the access log actually appear within access log records.

The following table lists the format for each access log record:

Access Log Format

Access Log Field	Description
<i>client_IP_address</i>	IP address of client, such as 123.45.123.45
- -	Two hyphens for compatibility with standard Web server log formats.

(Table Page 1 of 5)

Access Log Format (continued)

Access Log Field	Description
<i>timestamp</i>	Time that client accessed the file in the format: <i>dd/Mmm/yyyy:hh:mm:ss TZ</i> where <i>TZ</i> is the time zone expressed as the number of hours relative to the Coordinated Universal Time (Greenwich, England) and is relative to the Server. For example: [31/Oct/1996:13:44:32 -0800]
"GET <i>filename</i>	File name (and path) requested by the client. Path is everything in the URL after the port number. If the client requests a file that doesn't exist, UNKNOWN appears in place of <i>filename</i> .
<i>protocol/version "</i>	Application-layer protocol used to send the clip to the client. Possible values are: RTSP PNA HTTP In addition, a letter at the end of the string indicates which transport type was used:
	(blank) UDP connection
	T TCP connection
	H HTTP connection
	M Multicast
	For example, PNAT means that the clip was sent using the PNA protocol over a TCP connection.
	The version number indicates the edition of the protocol.
<i>HTTP_status_code</i>	Return code using HTTP standard error codes. Usually returns 200.
<i>bytes_sent</i>	Number of bytes transferred to the client.

(Table Page 2 of 5)

Access Log Format (continued)

Access Log Field	Description																
[<i>client_info</i>]	<p>Describes the version and type of client being used. Client information appears in the following format, [<i>platform_version_client_type_distribution_language_CPU</i>]. For example, Win95_4.0_3.0.0.19_play32_PN01_EN_586</p> <p>If client information can't be gathered (the request came from a client that chose not to send statistics, or from a browser connecting to RealSystem Administrator pages), UNKNOWN appears within the brackets.</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>platform</i></td> <td>Operating system RealPlayer runs on—Win16, WinNT, Mac, and so on.</td> </tr> <tr> <td><i>version</i></td> <td>Operating system version number.</td> </tr> <tr> <td><i>client</i></td> <td>Version number of RealPlayer.</td> </tr> <tr> <td><i>type</i></td> <td>Type of RealPlayer.</td> </tr> <tr> <td><i>distribution</i></td> <td>Distribution code of RealPlayer.</td> </tr> <tr> <td><i>language</i></td> <td>Language setting in RealPlayer.</td> </tr> <tr> <td><i>CPU</i></td> <td>Type of processor on which the client is running. If the processor does not have a hardware Floating Point Unit, the string "no-FPU" is appended to the end of the CPU field with no delimiter.</td> </tr> </tbody> </table> <p>RealAudio Player 1 shows only two fields for [<i>client_info</i>]. They are <i>platform</i> and <i>client</i>.</p>	Field	Description	<i>platform</i>	Operating system RealPlayer runs on—Win16, WinNT, Mac, and so on.	<i>version</i>	Operating system version number.	<i>client</i>	Version number of RealPlayer.	<i>type</i>	Type of RealPlayer.	<i>distribution</i>	Distribution code of RealPlayer.	<i>language</i>	Language setting in RealPlayer.	<i>CPU</i>	Type of processor on which the client is running. If the processor does not have a hardware Floating Point Unit, the string "no-FPU" is appended to the end of the CPU field with no delimiter.
Field	Description																
<i>platform</i>	Operating system RealPlayer runs on—Win16, WinNT, Mac, and so on.																
<i>version</i>	Operating system version number.																
<i>client</i>	Version number of RealPlayer.																
<i>type</i>	Type of RealPlayer.																
<i>distribution</i>	Distribution code of RealPlayer.																
<i>language</i>	Language setting in RealPlayer.																
<i>CPU</i>	Type of processor on which the client is running. If the processor does not have a hardware Floating Point Unit, the string "no-FPU" is appended to the end of the CPU field with no delimiter.																
[<i>client_GUID</i>]	<p>Unique ID generated during RealPlayer installation that enables you to track details for individual clients.</p> <p>If client information can't be gathered (the request came from a client that chose not to send statistics, or from a browser connecting to RealSystem Administrator pages), UNKNOWN appears within the brackets.</p> <p>If the user elects to suppress this information, this field will show a series of zeroes: 00000000-0000-0000-0000-000000000000 instead of a unique identifier. Refer to "Omitting Client Identifiers" on page 299.</p> <p>Included when Logging Style is set to 2 or higher.</p>																

(Table Page 3 of 5)

Access Log Format (continued)

Access Log Field	Description
[Stat1] (see the “Statistics Type 1 Information” table)	Connection statistics sent by the client when it completes playing a clip (see the “Statistics Type 1 Information” table). When the client blocks connection statistics, the field is replaced by [UNKNOWN]. Note that there is no space between the closing square bracket of this statistics type and the opening square bracket of the next statistics type. Included when Stats Mask is 1, 3, 5, or 7.
[Stat2] (see the “Statistics Type 2 Information” table)	Extended connection statistics sent by the client when it completes playing a clip (see the “Statistics Type 2 Information” table). When the client blocks connection statistics, the field is replaced by [UNKNOWN]. Note that there is no space between the closing square bracket of this statistics type and the opening square bracket of the next statistics type. Included when Stats Mask is 2, 3, 6, or 7.
[Stat3] (see the “Statistics Type 3 Information” table)	Actions taken by the visitor while playing the clip (see the “Statistics Type 3 Information” table). When the client preferences are set to block statistics, this field is replaced by [UNKNOWN]. Note that there is no space between the closing square bracket of the previous statistics type and the opening square bracket of this statistics type. Included when Stats Mask is 4, 5, 6, or 7.
<i>file_size</i>	Total amount in bytes of media data in the media file. This number is less than the size of the media file because it does not include the file header and other non-media information stored in the file. For live broadcasts, <i>file_size</i> is always 0. Included when Logging Style is set to 1 or higher.
<i>file_time</i>	Total length, in seconds, of media stored in the media file. For live broadcasts, <i>file_time</i> is always 0. For .smi files, this is always 20. Included when Logging Style is set to 1 or higher.
<i>sent_time</i>	Total length, in seconds, of the media sent to the client. Included when Logging Style is set to 1 or higher.
<i>resends</i>	Number of packets successfully re-sent because of transmission errors. Included when Logging Style is set to 1 or higher.
<i>failed_resends</i>	Number of packets not successfully re-sent in time to correct transmission errors. Included when Logging Style is set to 1 or higher.

(Table Page 4 of 5)

Access Log Format (continued)

Access Log Field	Description
<i>[stream_components]</i>	Type of material sent, indicated in the following pattern: RealAudio RealVideo Event Image_maps 1 shows that the stream includes this type, 0 indicates that it does not. Thus, a stream that included RealVideo and RealAudio but no events or image maps would appear in the access log as 1 1 0 0 . Included when Logging Style is set to 3 or 4.
<i>start_time</i>	Timestamp of start time. Included when Logging Style is set to 3 or 4.
<i>server_address</i>	IP address of RealServer supplying the clip. Included when Logging Style is set to 3 or 4.
<i>average_bitrate</i>	Average bitrate of clip. Included when Logging Style is set to 4.
<i>packets_sent</i>	Number of packets sent. Included when Logging Style is set to 4.
<i>presentation_id</i>	Number used by all clips in the same SMIL or Ram presentation. SMIL files are also included in the log, and use the same number as the clips they reference. The number is assigned by RealServer at the time of transmission. Included when Logging Style is 5.

(Table Page 5 of 5)

LoggingStyle Results

The format of the access log under each of the different Logging Style values is shown in the table below:

Logging Style Effect on Access Log

Logging Style value	Individual record format
0	<i>client_IP_address - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [StatsMask results]</i>
1	<i>client_IP_address - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [StatsMask results] file_size file_time sent_time resends failed_resends</i>
2	<i>client_IP_address - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_GUID] [StatsMask results] file_size file_time sent_time resends failed_resends</i>

(Table Page 1 of 2)

Logging Style Effect on Access Log (continued)

Logging Style value	Individual record format
3	<i>client_IP_address</i> - - [<i>timestamp</i>] " <i>GET filename protocol/version</i> " <i>HTTP_status_code bytes_sent [client_info] [client_GUID]</i> <i>[StatsMask results] file_size file_time sent_time resends</i> <i>failed_resends [stream_components] start_time server_address</i>
4	<i>client_IP_address</i> - - [<i>timestamp</i>] " <i>GET filename protocol/version</i> " <i>HTTP_status_code bytes_sent [client_info] [client_GUID] [Stats</i> <i>Mask results] file_size file_time sent_time resends failed_resends</i> <i>[stream_components] start_time server_address average_bitrate</i> <i>packets_sent</i>
5	<i>client_IP_address</i> - - [<i>timestamp</i>] " <i>GET filename protocol/version</i> " <i>HTTP_status_code bytes_sent [client_info] [client_GUID] [Stats</i> <i>Mask results] file_size file_time sent_time resends failed_resends</i> <i>presentation_id</i>

(Table Page 2 of 2)

StatsMask Results

The information gathered by each of the three Statistics Types are listed in this section. Stat1 and Stat2 report information about the RealAudio portion of a clip. Even if a clip includes both RealAudio and RealVideo, these statistics report solely RealAudio information. Stat3 reports information about visitor and client behavior while playing all types of clips or presentations.

When Stats Mask is 0, two square brackets ([]) appear instead of the Stat1, Stat2, and Stat3 sections.

Stat1 Syntax

Statistics Type 1 gathers basic information about how successfully audio clips were received by the client. It also tells what the client used to decode the audio portion of the clip.

Syntax of this portion of the access log record:

[Stat1: *packets_received out_of_order missing early late audio_format*]

The table below gives the information collected by this statistic type:

Statistics Type 1 Information	
Field	Description
<i>packets_received</i>	Total number of packets received by the client.
<i>out_of_order</i>	Number packets received by the client out of order. These packets are reordered as they are being played by the client.
<i>missing</i>	Number of packets requested by the client, but that the client did not receive.
<i>early</i>	Number of requested packets received too early by the client.
<i>late</i>	Number of packets received too late by the client.
<i>audio_format</i>	Name of the decoder used to play the clip. Possible values are: sivr RealAudio version 5 formats dnet RealAudio version 3 formats 28.8 RealAudio version 2, 28.8 format lpcJ RealAudio version 2, 14.4 format cook RealAudio version 6 format

Stat2 Syntax

Statistics Type 2 provides details about the success of clip delivery, giving information about bandwidth requests. Re-sent packets are described in detail here. It identifies which transport type was used to make the connection and which video decoder played the clip. This set of statistics uses the following format:

[Stat2: *bandwidth available highest lowest average requested received late rebuffering transport startup format*]

The table below explains what information is collected by this statistic type:

Statistics Type 2 Information	
Field	Description
<i>bandwidth</i>	Bandwidth of the clip, in bits per second.
<i>available</i>	Average bits per second available to the user while the clip was playing.
<i>highest</i>	Highest time between the client resend packet request and the packet resend arrival, in milliseconds.
<i>lowest</i>	Lowest time between the client resend packet request and the packet resend arrival, in milliseconds.

(Table Page 1 of 2)

Statistics Type 2 Information (continued)

Field	Description
<i>average</i>	Average time between the client resend packet request and the packet resend arrival, in milliseconds.
<i>requested</i>	Number of resend packets requested by the client.
<i>received</i>	Total number of re-sent packets received by the client.
<i>late</i>	Number of re-sent packets received by the client too late.
<i>rebuffering</i>	Rebuffering percentage for the clip.
<i>transport</i>	Transport type for the connection. Values are: 0: UDP 1: TCP 2: IP Multicast 3: PNAviaHTTP
<i>startup</i>	Time when the client receives the first clip data, in milliseconds. The data may arrive before the clip starts playing.
<i>format</i>	Name of the decoder used to play the clip. Possible values are: sivr RealAudio version 5 formats dnet RealAudio version 3 formats 28.8 RealAudio version 2, 28.8 format lpcJ RealAudio version 2, 14.4 format cook RealAudio version 6 format

(Table Page 2 of 2)

Stat3 Syntax

Statistics Type 3 provides detailed information about viewer action while listening or viewing clips. It addresses advanced features of the implementation, notably ads and image maps. You can find out at what point in the clip a viewer clicked on an image map or stopped watching the clip.

If Stats Mask is configured to gather statistics type 3 (Stat3), note that the access log file size will grow rapidly. If you configure Stats Mask to collect this information, be sure to review the log file frequently. This statistics type uses the following format:

```
[Stat3:timestamp|elapsed_time|action|;]
```

Records of activity are separated by a semicolon (;) and are in the following form:

```
timestamp|elapsed_time|action|;
```

Thus, the Stat3 record of a visitor pausing, resuming play, and watching to the clip's end would look like the following:

[Stat3:4360|2107|PAUSE|;8401|2107|RESUME|;12608|6321|STOP|;]

The table below gives the format of the Stat3 records:

Statistics Type 3 Information

Field	Description	
<i>timestamp</i>	Time in milliseconds when action occurred. It is relative to the connect time of the client.	
<i>elapsed_time</i>	Elapsed time of the clip when the behavior occurred, given in milliseconds.	
<i>action</i>	CLICK	Visitor clicked on the image map. Further information includes:
	<i>x-coord</i>	Horizontal coordinate of click.
	<i>y-coord</i>	Vertical coordinate of click.
	<i>action</i>	Action that occurred. This is one of the following:
	PLAYER=" <i>url</i> "	The URL of the link the viewer clicked, as used in the client
	URL=" <i>url</i> "	The URL of the link the viewer clicked, as used in the Browser.
	SEEK=" <i>destination</i> "	The seek destination point, in milliseconds.
	PAUSE	The visitor paused the client.
RESUME	Resume play after a pause, seek or stop.	
SEEK	The seek destination point, in milliseconds.	
STOP	End of clip reached.	
RECSTART	RealPlayer Plus began recording the clip.	
RECEND	RealPlayer Plus stopped recording the clip.	

Customizing Information Reported by the Access Log

RealServer uses the following settings for the access log (you can view these in RealSystem Administrator by clicking **General Setup**>**Logging**):

- **Logging Style**—At installation, this is set to 5.
- **Stats Mask**—The default value is 3.
- **Log Rolling Frequency**—settings for creating new log files at specified intervals. See “Log File Rolling” on page 304.
- **Access Log File**—RealSystem Administrator will place files in the Logs subdirectory of the main RealServer directory. The default file name of the access log file is `raccess.log` (Windows) or `raccess` (UNIX). The directory (if any) typed here can be absolute or relative to the base path of the main mount point.

If **Access Log File** is blank, RealServer records access information in the `raccess.log` or `raccess` file located in the same directory as the RealServer executable file.

The name of the access file will be different if Log File Rolling is enabled; see “Log File Rolling” on page 304.

To customize the information gathered in the access log, you must first decide what types of information you want to gather. Then make the appropriate changes to Logging Style, which collects information about RealServer activity, and to Stats Mask, which gathers statistics about what arrived at the client and viewer behavior while playing the clips.

Changing Information Gathered with Logging Style

Logging Style has six options, styles 0 through 5. Styles 0 through 4 each includes information of the logging styles with lower numbers. Thus, Logging Style 3 collects the information that’s collected by styles 0, 1, and 2, as well as the material gathered by style 3. Logging Style 5 consists of the fields in Logging Style 2, plus the `presentation_id` field.

If you omit this variable, RealServer uses the default style of 5.

A list of information gathered by each value is given below.

Logging Styles 0, 1, and 3 contain some additional information, as described in “Access Log Format” on page 288.

Information Collected by Logging Style

To gather this information...	...set LoggingStyle to this value
Bytes sent	0 or higher
Clip name including path	0 or higher
Client IP address and platform information	0 or higher
Timestamp	0 or higher
File size (in bytes)	1 or higher
File time (total file length in seconds)	1 or higher
Packets successfully and unsuccessfully re-sent	1 or higher
Protocol (RTSP or PNA)	1 or higher
Send time (total media sent in seconds)	1 or higher
Transport method (TCP, UDP) and version	1 or higher
Client ID	2 or higher
Server IP Address	3 or 4
Stream components	3 or 4
Timestamp for start time	3 or 4
Average bitrate	4
Packets sent	4
Common presentation identifier	5

Changing Information Gathered with Stats Mask

Stats Mask supplies more detailed information to the access log. This variable is optional. For a complete description of information collected by each statistics type, and the syntax of the types as they appear in the access log, see the “Statistics Type 1 Information” table on page 294, the “Statistics Type 2 Information” table on page 294, and the “Statistics Type 3 Information” table on page 296.

If you omit a value for Stats Mask, RealServer uses the default value of 3 (gather statistics types 1 and 2).

Collecting Combinations of Stats Mask Information

To gather this information...	...set Stats Mask to this value	Statistics types included		
		1	2	3
No additional statistics	0			
Statistics type 1 only	1	X		
Statistics type 2 only	2		X	
Both statistics types 1 and 2	3	X	X	
Statistics type 3 only	4			X
Both statistics types 1 and 3	5	X		X
Both statistics types 2 and 3	6		X	X
All statistics (types 1, 2, and 3)	7	X	X	X

Tip

If Stats Mask is configured to gather statistics type 3, the access log file size will grow rapidly. If you configure Stats Mask to collect this information, be sure to review the log file frequently, or use log file rolling.

Not all versions of RealPlayer supply the information requested by Stats Mask; Statistics type 2 is supplied by RealAudio Player versions 3 and later, and Statistics type 3 is supplied by RealPlayer 5 and later.

Omitting Client Identifiers

Normally, every access log record displays a unique client identification number for each user. However, both users and administrators have the option to omit this information from access log records.

If a user elects to withhold his software's unique client number, a string of zeroes appears instead: [00000000-0000-0000-0000-000000000000].

RealServer's default behavior is to use client identifiers, when available. It will show zeroes for those users who have opted to suppress their client software identifiers.

Regardless of the user's setting, you can instruct RealServer to always show the string of zeroes instead of the actual client identifier. If you choose this option, all access log records show zeroes, rather than the actual client identifiers. (This applies only to the logging styles that collect data for the [client_GUID] field—logging styles 2 and higher.)

There is no way to override the client's setting, should the user choose to send only zeroes.

- ▶ To disable collection of client identifiers:
 1. In RealSystem Administrator, click **General Setup**. Click **Logging**.
 2. From the **Disable Client GUID** list, select No.
 3. Click **Apply**.

Using the GET Statement to Identify Delivery Method

The GET statement within each access log record shows the path and file name of each file that RealServer served, as well as the protocol and protocol version used to stream or broadcast the file. (To see the GET statement in context, refer to the "Access Log Format" table on page 288.)

The table below summarizes the format in which each type of content is shown in the access log.

For live streams that use encoders developed for use with version 6 and later software, the file name will begin with encoder. For earlier encoders, the file name begins with live.

Summary of GET Statements

Feature	Protocol	Example Statement in Access Log
On-Demand Content		
On-demand streamed content	RTSP	"GET presentation/presentation.rm RTSP/1.0"
	PNA	"GET presentation/presentation.rm PNA/10"
	HTTP	"GET presentation/presentation.rm PNH/10"
SMIL files (1 record for the SMIL file, one record for each file listed within the SMIL file)	RTSP	"GET presentation/presentation.smi" "GET presentation/presentation.rt" "GET presentation/presentation.rp" "GET presentation/presentation.rm"

(Table Page 1 of 3)

Summary of GET Statements (continued)

Feature	Protocol	Example Statement in Access Log
ISP hosting—account-based	RTSP	"GET ~schu/music.rm RTSP/1.0"
	PNA	"GET ~schu/music.rm PNA/10"
	HTTP	"GET ~schu/music.rm PNH/10"
ISP hosting—dedicated	RTSP	"GET s/sc/schu/music.rm RTSP/1.0"
	PNA	"GET s/sc/schu/music.rm PNA/10"
	HTTP	"GET s/sc/schu/music.rm PNH/10"
RealSystem Administrator activity	HTTP	"GET admin/index.html HTTP/1.0"
View source request (for on-demand and live clips)	HTTP	"GET viewsource/template.html HTTP/1.0"
Authenticated on-demand streamed content	RTSP	"GET secure/topsecret.rm RTSP/1.0"
	PNA	"GET secure/topsecret.rm PNA/10"
	HTTP	"GET secure/topsecret.rm PNA/10"
Live Content		
Live unicast content, from version 6 or later encoding source	RTSP	"GET encoder/live.rm RTSP/1.0"
	PNA	"GET encoder/live.rm PNA/10"
	HTTP	"GET encoder/live.rm PNH/10"
G2SLTA content	any	same as live unicast content
Live redundant content, from version 6 or later encoding source	RTSP	"GET redundant/live.rm RTSP/1.0"
	PNA	"GET redundant/live.rm PNA/10"
	HTTP	"GET redundant/live.rm PNH/10"
Live unicast content, from pre-G2 encoding source	RTSP	"GET live/live.rm RTSP/1.0"
	PNA	"GET live/live.rm PNA/10"
	HTTP	"GET live/live.rm PNH/10"
Authenticated live streamed content	RTSP	"GET secure/encoder/live.rm RTSP/1.0"
	PNA	"GET secure/encoder/live.rm RTSP/1.0"
	HTTP	"GET secure/encoder/live.rm RTSP/1.0"
Push splitting—receiver's access log	RTSP	No record is created.
	PNA	
Push splitting—receiver's access log	RTSP	"GET broadcast/Japan/encoder/live.rm RTSP/1.0"
	PNA	"GET broadcast/Japan/encoder/live.rm PNA/10"

(Table Page 2 of 3)

Summary of GET Statements (continued)

Feature	Protocol	Example Statement in Access Log
Pull splitting—transmitter's access log	RTSP	No record is created.
	PNA	
Pull splitting—receiver's access log	RTSP	"GET broadcast/pull/Japan:2030/encoder/live.rm RTSP/1.0"
	PNA	"GET broadcast/pull/Japan:2030/encoder/live.rm PNA/10"
Multicasting—back-channel	RTSP	"GET encoder/live.rm RTSPM/1.0"
	PNA	"GET encoder/live.rm PNAM/10"
Multicasting—scalable (two records are usually created)	HTTP	"GET concert.rm.sdp HTTP/1.0"
	and RTP	"GET concert.rm RTP/2.0"

(Table Page 3 of 3)

Error Log

The error log contains both information and error messages about Server operation. By looking for patterns of errors, you can troubleshoot and correct possible problems on your site.

View the text of the error log using a word processor or text editor.

The error log is an excellent tool for troubleshooting any problems that may arise with your RealServer. An entry is made to the error log only when an error occurs. If no errors occur, this file will not exist.

If you have an error message that refers to a fatal error, contact the RealNetworks Technical Support Department for assistance.

RealServer uses the following settings to record information in the error log (you can view them from RealSystem Administrator by clicking **General Setup>Logging**):

- **Log Rolling Frequency**—settings for creating new log files at specified intervals. See “Log File Rolling” on page 304.
- **Error Log File**—the default location is the Logs subdirectory of the main RealServer directory. The default name of the error log file is `rmerror.log`.

Error Log File Format

The error log records client connections and RealServer errors. Each time an error is generated by RealServer, a record is created in the error log. The error log path is stored in the same directory as the access log, indicated by the `LogPath` variable.

Syntax of the file is as follows:

```
***date time servername(process_ID): error_message
```

where entries are defined below:

Error Log Syntax	
Entry	Meaning
<code>***</code>	Three asterisks indicate an error. Informational messages are not preceded by asterisks.
<code>date</code>	Date on which the error occurred. Given in the form d-Mmm-YY.
<code>time</code>	Time the error occurred, according to RealServer. Given in the form HH:MM:SS.TT.hhh

(Table Page 1 of 2)

Error Log Syntax (continued)

Entry	Meaning
<i>servername(process_ID)</i>	The Server name, followed by the process ID in parentheses.
<i>error_message</i>	Text of error message

(Table Page 2 of 2)

Log File Rolling

Log files can grow indefinitely as they accumulate data. To keep log files to a manageable size, you can limit the access log to a week's worth of information or a certain file size, and RealServer will begin a new log file when the limit is reached.

► To set up log file rolling:

1. In RealSystem Administrator, click **General Setup**. Click **Logging**.
2. In the appropriate Access Log or Error Log section, limit the log files by time period or by size:
 - To limit by time period, select the number and the period from the **Log Rolling Frequency** list. You can save hourly, daily, weekly, or monthly.
 - To limit by file size, type the maximum number of megabytes for a log file in **Log Rolling Size** box.

If you supply values for all four boxes, RealServer will use the size or time period that is reached first.

3. Click **Apply**. Files will be named according to the structure shown in “Rolled Log File Format” below.

Rolled Log File Format

Rolled log files are named with the following format:

name.log.datestamp

where:

- | | |
|-------------|---|
| <i>name</i> | Name of the regular log file, as taken from the Access Log File or Error Log File box (usually rmaccess for access logs, and rmerror for error logs). |
| <i>log</i> | The log file extension. |

<i>datestamp</i>	The date stamp, in the following format: <i>YYYYMMDDHHMMSS</i> where:
<i>YYYY</i>	the four-digit year
<i>MM</i>	two digits for the month
<i>DD</i>	date, in two digits. January would be 01.
<i>HH</i>	hour
<i>MM</i>	minutes
<i>SS</i>	seconds

Disabling Log File Rolling

If you turn off log file rolling, RealServer will create a single large log file.

► To disable log file rolling:

1. In RealSystem Administrator, click **General Setup**. Click **Logging**.
2. Select 0 from the **Log Rolling Frequency** list in the Access Log section or in the Error Log section.
3. Delete any text from the **Log Rolling Size** box in the Access Log section or in the Error Log section.
4. Click **Apply**.

Cached Requests Log

Whenever RealServer sends a stream, it records that information in the access log. In addition, if RealServer sends a stream to RealProxy, it creates an entry in the cache.log file. Requests that will be stored in caches are identified by the port number to which they send the request.

RealServer uses the following settings to create cache request log files (you can view the settings from RealSystem Administrator by clicking **Cache**>**Cache**):

- **Cache Log Path**—the path and file name of the cached requests log file. The default location is the Logs directory, and the default name is cache.log.

Reading a Cached Requests Log

The entries in the cache.log file use one of two formats: a general information format, and a clips served format.

Note

As with other log files, the brackets within the cache.log file always appear and do not indicate optional material.

General Information Format

[*Day Mmm DD hours:minutes:seconds YYYY*] *message*

where:

<i>Day</i>	three-letter abbreviation for the day, such as Thu for Thursday
<i>Mmm</i>	three-letter abbreviation for the month, such as Jun for June
<i>DD</i>	one or two-digit date
<i>hours:minutes:seconds</i>	time, in twenty-four hour format: hours:minutes:seconds
<i>YYYY</i>	four-digit year

Clips Served Format

[*Day Mmm DD hours:minutes:seconds YYYY*] *IP_address path_filename*

where *IP_address* and *path_filename* refer to the stored location of the content.

Disabling Cache Request Logging

To disable the log file of cache requests, change **Cache Requests** to Disabled.

Chapter 20

STREAMING TARGETED ADS

RealServer can dynamically insert ads into streaming presentations. Offering integration with any HTML-based ad serving system, RealServer uses SMIL (Synchronized Multimedia Integration Language) to lay out ads and requested content in RealPlayer. This chapter explains how to set up RealServer's ad streaming features.

Additional Information

The ad chapter in *RealSystem Production Guide* explains how to write SMIL-based presentations that include streaming ads. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

How Ad Streaming Works

RealServer requires no special programming to integrate with popular ad serving systems. Ad servers are designed to place ad URLs in requested Web pages. To get ads from a third-party ad server, RealServer simply requests HTML containing the ad URLs from the ad server. That HTML may come directly from the ad server, or through a page hosted on a Web server.

When it receives the returned HTML, RealServer extracts the ad's file URL and hypertext link URL. It then inserts these URLs into the SMIL presentation requested by RealPlayer. Through this method, any third-party ad server designed for HTML pages can serve ads to the SMIL-based RealPlayer.

Banner Ads

RealServer can deliver single banner ads and rotating banner ads for prerecorded content or live broadcasts. With rotating banner ads, RealServer sends RealPlayer a new GIF, animated GIF, or JPEG ad at regular intervals throughout a presentation. To include ads with requested clips, content creators write a SMIL file that has one or more regions for ads. Instead of ad URLs, the SMIL file contains one or more <RealAdInsert/> tags that RealServer

expands into unique ad URLs when RealPlayer requests the file. You can also use the SMIL generation feature, described below, to avoid writing SMIL by hand.

Streaming Media Ads

Although RealServer can deliver standard banner ads, its power lies in its ability to stream ads in formats such as RealAudio, RealVideo, RealText and Flash. The delivery mechanism for streaming media ad URLs is the same as for static image ad URLs: the ad server places URLs in HTML requested by RealServer. The only difference is that these ad URLs are for RTSP-streamed clips on a RealServer host, rather than for HTTP-downloaded image files on an ad server.

SMIL Generation

RealSystem uses SMIL for ad layout. Although you have more flexibility when writing your own SMIL files, RealServer's SMIL generation feature can automatically create or modify SMIL files that include ads. You can thereby stream ads without writing or modifying SMIL files by hand. This feature is useful if you have a large collection of existing clips or SMIL presentations for which you want to include ads. SMIL generation works for banner ads as well as lead-in ads.

Quick Start for Testing Ad Banner Insertion

RealServer comes preconfigured for inserting banner ads into streaming presentations. Follow the steps below to see a sample streaming banner ad.

- To test banner ads using a RealNetworks Web server:

1. Start RealServer.
2. Start RealPlayer and choose **File>Open Location**.
3. Enter the following URL, substituting the actual name of your RealServer for `yourserver.com`, and RealServer's RTSP port for `554`:

```
rtsp://yourserver.com:554/adtag/general/smilgen/banner/g2video.rm
```

Requesting this URL verifies that ad streaming works by playing the `g2video.rm` clip included with RealServer. The `/smilgen/banner/` mount point in the URL causes RealServer to generate a SMIL file that defines a banner ad region and a video region. The `/adtag/general/` mount point is preconfigured to pull a banner ad from a RealNetworks Web server.

Testing your Own Banner Ads

You can quickly modify RealServer to pull ads from your ad serving system instead of the RealNetworks Web server. Just point RealServer to a Web page that provides the ad URLs.

► To test banner ads using your own Web server:

1. Choose the Web page where RealServer gets ads. For testing purposes, you can use any page on your Web site that includes a 468-pixel by 60-pixel top banner ad in GIF or JPEG format.
2. In RealSystem Administrator, click **Advertising**, then click **Ad Serving**. This displays a dialog for configuring RealServer to work with your ad serving system.
3. Highlight `/adtag/general/`.
4. In the **Target HTML** field, change the default value of **`http://www.real.com/ads/g2ads_def.html`** to the fully qualified URL of the Web page where RealServer gets ad URLs.
5. Click **Apply**.
6. Restart RealServer by clicking the **Restart** icon at the top of the Administrator.
7. Open the previous RTSP URL in RealPlayer. This plays the same video, but with a banner ad pulled from the **Target HTML** URL, rather than from a RealNetworks server.

General Steps for Setting Up Ad Streaming

► Follow these steps to set up ad streaming:

1. Integrate RealServer with your ad serving system.
To retrieve ad URLs, RealServer can integrate directly with many ad serving systems. It can also request an HTML page on a Web server to get ad URLs. For more on these two methods, refer to “Getting Ad URLs from an Ad Server” beginning on page 310.
2. Create ad streaming mount points.
These mount points, which determine what type of ad RealServer streams, appear in URLs for requested content. “Configuring RealServer to Stream Ads” on page 315 tells how to set up the mount points and configure ad streaming features.

3. Set up automatic SMIL generation mount points (optional).

If you have a lot of existing content for which you want to include ads, this optional feature can save you much time and effort. See “Generating SMIL Files for Ads” starting on page 326 for more information.

4. Communicate ad streaming features to content creators.

Content creators refer to the ad chapter in *RealSystem Production Guide* for instructions on creating SMIL files with `<RealAdInsert/>` tags. They rely on you for information about RealServer’s specific ad streaming features, however. You need to communicate the following to content creators:

- whether automatic SMIL generation is in use
- the ad formats available, such as GIF, animated GIF, RealVideo, or Flash
- the width and height of available ads
- the mount points to include in URLs used to request streaming presentations
- the allowable mount point override options

Additional Information

Override options are described in “Overriding Mount Point Settings through SMIL” on page 325. To view *RealSystem Production Guide*, click **Resources** under **Help** in RealSystem Administrator.

Getting Ad URLs from an Ad Server

To stream an ad, RealServer gets the URL to the ad clip (whether a static image or a streaming clip such as video) from an ad serving system. You can integrate RealServer directly with many popular ad servers. Or RealServer can get ad URLs through a Web server integrated with an ad server. Both options are described below. The location RealServer accesses to get ad URLs is its *target URL*. Each target URL returns URLs (both the file URL and the clickthrough URL) for a specific type of ad.

Understanding Ad Types

The types of ads RealServer streams relate directly to the target URLs. To stream just one type of ad, you need just one target URL where RealServer gets

all ad URLs. If you plan to stream different types of ads, such as image banner ads and streaming video clips, though, you need target URLs for each ad type. Several things determine the “ad type”:

- ad file format

A GIF or animated GIF image has a different file format than a streaming video clip. If you host both GIF and RealVideo ads, for example, you’ll need at least two target URLs, one for each file format.

- ad file size

For some presentations, you might insert full-size banner ads (468 pixels by 60 pixels). For others, you may include half-size banner ads (234 pixels by 60 pixels). You need to set up a different target URL for each ad size. Similarly, you’ll need a different target URL for each size of RealVideo ad you stream.

- ad audience

You may want to use different ads based on the subjects of your streaming presentations. Streaming sports clips may have different advertisers than streaming news stories, for example. You can use different target URLs when ads are the same size and formats, but reach different audiences.

- ad serving system

RealServer can integrate with several ad streaming systems. Each system may work differently, however. Suppose you stream just standard banner ads, but pull the ads from two different ad serving systems. You’ll need two different target URLs, one for each system, even if the ads have the same format, size, and audience.

Combinations of file format, file size, audience, and ad serving system make up the types of ads RealServer gets from target URLs. As “Understanding Ad Streaming Mount Points” on page 316 explains, the number of target URLs you use determines how many ad streaming mount points you set up.

Guidelines for Ads in Streaming Presentations

The following points and guidelines apply to ad files and ad URLs used in streaming RealServer presentations:

- A banner ad must be a GIF, animated GIF, or JPEG image. RealPlayer cannot display ads that are HTML tables, Java applets, and so on.

- Ad clickthrough URLs are thrown to the viewer's browser, unless the URL is a SMIL hyperlink URL that targets RealPlayer.
- For rotating banner ads, RealServer requests the target URL at regular intervals throughout the presentation as defined by the ad mount point. Each time it requests the target URL, the ad serving system returns a URL to a different banner ad.
- For single banner ads or streaming media ads, RealServer requests the target URL for each <RealAdInsert/> tag in the SMIL file. If the file has two <RealAdInsert/> tags, for example, RealServer requests the target URL twice and uses the ad URL returned with the first request for the first <RealAdInsert/> tag, the URL returned with the second request for the second tag.
- RealPlayer supports cookies just like a Web browser. If the Web server hosting the ad target page attempts to set a Web browser cookie through an HTTP response header, RealServer intercepts the cookie and writes it to RealPlayer. When RealPlayer requests content that includes an ad, RealServer requests from the user's Web browser any cookies for the target Web server's domain and path, forwarding these to the Web server.

Note

RealPlayer users can disable cookie support in their RealPlayer preferences.

- You can modify your ad server to produce RTSP-based URLs to streaming media ads in RealAudio, RealVideo, and Flash formats.

Additional Information

The ad chapter in *RealSystem Production Guide* explains how to use the <RealAdInsert/> tag in SMIL files. To view this guide, click **Resources** under **Help** in RealSystem Administrator.

Integrating RealServer Directly with an Ad Server

RealServer works with a number of popular ad serving systems. To integrate RealServer with the ad serving system you're using, see this document:

<http://docs.real.com/docs/adapp.pdf>

This document explains how to get ad URLs through HTML generated directly by different ad serving systems. Typically you do this through a target

URL that causes the ad server to return the ad URLs you want. You then tie this target URL to a RealServer mount point, as described in “Creating Ad Streaming Mount Points” on page 318.

Integration information is given in this separate document, rather than this manual, so that RealNetworks can provide you with the latest information. To read this document, which is in Adobe’s Portable Document Format (PDF), use the free Acrobat Reader, available here:

<http://www.adobe.com/products/acrobat/readstep.html>

If the integration document does not cover the ad server you’re using, you can set up a target HTML page on a Web server.

Setting up a Target HTML Page on a Web Server

Integrating RealServer directly with your ad serving system is the preferred method for getting ad URLs. However, RealServer can also retrieve ad URLs by requesting an HTML page integrated with an ad serving system. This target page may be an existing Web page on your site, or a page specifically set up to provide RealServer with ad URLs. By using this page as an intermediary for exchanging ad URLs, RealServer can work with virtually any ad serving system.

You set up your HTML target page as required by your ad serving system. For example, some systems require a page to have a server-side `#include` tag that expands into ad URLs when the page is served. When RealServer requests the page, the returned page should have mark-up that includes an `` tag for the ad file, as well as a clickthrough hypertext link, similar to this:

```
<a href="http://www.real.com">  
  
</a>
```

RealServer then replaces the requested SMIL file’s `<RealAdInsert/>` tag with these URLs.

Additional Information

Refer to your ad serving system’s documentation for information on how to insert ad URLs into the target HTML page each time it is requested.

Guidelines for Creating a Target HTML Page

In addition to the general points listed in “Guidelines for Ads in Streaming Presentations” on page 311, the following points apply when you get ad URLs through an HTML page hosted on a Web server:

- You can designate the ad to use by including the variable `realad="1"` in the image source tag. The `realad` value “1” is a syntax requirement that simply tells RealServer which image to use. Using the `realad` variable requires you to configure the ad server to include the variable in the returned HTML. Here is an example:

```

```

- If no `realad="1"` value is present, RealServer uses the first `` tag in the HTML file as the ad source.

Warning

If other hyperlinked images precede the desired ad image, RealServer may not be able to distinguish the correct URL to extract.

- To minimize ad streaming latency, keep the target HTML page as small as possible, serve the page from a Web server that has fast response, and ensure that the network connection between the machines is fast.

Requesting SMIL Files from an Ad Server

As described above, the target URL typically returns ad URLs that RealServer incorporates into the requested SMIL file. However, the ad server can also return a full presentation SMIL file. To set this up, you modify your ad server to generate SMIL mark-up, including a layout, ad URLs, requested clip URLs, and any other SMIL attributes. The returned SMIL file must start with `<smil>` and end with `</smil>` as shown here:

```
<smil>  
<body>  
  ...all SMIL mark-up...  
</body>  
</smil>
```

RealServer recognizes that the ad server has returned SMIL mark-up rather than simple ad URLs. Instead of streaming the SMIL file RealPlayer originally

requested, RealServer streams the returned SMIL mark-up in full. The requested SMIL file just needs to be a shell for a `<RealAdInsert/>` tag:

```
<smil>
  <RealAdInsert/>
</smil>
```

The following table illustrates the basic steps involved in generating a SMIL file by an ad server. If you integrate RealServer directly with an ad server, you don't use an HTML target file as shown in column 2. Rather, the target URL causes the ad server to return the mark-up shown in column 3. Some syntax details have been omitted for clarity.

Generating SMIL Presentations with an Ad Server

SMIL File Requested by Viewer	HTML Target File Used to Expand <code><RealAdInsert/></code>	File Returned from Ad Server	SMIL File Delivered to Viewer
<pre><smil> <RealAdInsert/> </smil></pre>	<pre><HTML> ... <!--#include....--> ... </HTML></pre>	<pre><smil> <body> ...mark-up... </body> </smil></pre>	<pre><smil> <body> ...mark-up... </body> </smil></pre>
The requested file is a shell for <code><RealAdInsert/></code> .	The request URL points RealServer directly to an ad server, or, as shown above, to an HTML page integrated with an ad server.	The ad server returns a full SMIL file that contains mark-up for a streaming presentation.	RealServer streams the SMIL file returned by the ad server to RealPlayer.

Configuring RealServer to Stream Ads

RealServer can insert a banner ad, rotating banner ad, or streaming ad clip into a requested SMIL file. Content creators lay out the presentation with SMIL, but instead of including URLs to ad clips, they add `<RealAdInsert/>` tags that cause RealServer to get ads from an ad server. The URL for the SMIL file request determines what type of ad RealServer inserts in place of a `<RealAdInsert/>` tag.

Understanding Ad Streaming Mount Points

Once you've determined how many ad types you need to stream, you can plan the mount points you'll need. ("Understanding Ad Types" on page 310 explains how various factors make up an "ad type.") Each mount point gives RealServer a different target URL where it finds the ad URLs. If one type of ad, such as a GIF banner ad, works for all content you stream, set up just one ad streaming mount point, such as `/adtag/general/`. You'll probably need to set up several mount points, however.

Once you've set up the mount points, the ad used to replace a `<RealAdInsert/>` tag depends on the URL used to request the SMIL file. Here's an example of a SMIL request URL:

```
<a href="http://realserver.example.com:8080/ramgen/adtag/general/start.smil">
```

The target URL defined for the `/adtag/general/` mount point determines what type of ad replaces the SMIL file's `<RealAdInsert/>` tag or tags. `<RealAdInsert/>` tags may have parameters that override the settings, though. Additional mount points not related to ad streaming, such as a mount point to verify a user name and password, may precede an ad streaming mount point in the SMIL file request URL:

```
<a href="http://realserver.example.com:8080/ramgen/secure/adtag/general/start.smil">
```

To stream more than one type of ad, you define additional mount points like these:

```
/adtag/sports/  
/adtag/tech/
```

These mount points appear in different request URLs that target different types of ads:

```
<a href="http://realserver.example.com:8080/ramgen/adtag/sports/start.smil">  
<a href="http://realserver.example.com:8080/ramgen/adtag/tech/start.smil">
```

Tip

Although you can create a new mount point for every ad type you stream, you do not always have to do this. In some cases, it is easier to use SMIL to override a mount point's settings, rather than create a new mount point. Before you set up mount points, read "Overriding Mount Point Settings through SMIL" on page 325.

Choosing the Ad Streaming Base Mount Point

Ad streaming mount points like `/adtag/general/` constitute “virtual paths” that invoke RealServer’s ad streaming feature. The base mount point represents the actual file system mount point RealServer uses to find the requested file. When you define an ad streaming mount point, you also indicate its base mount point. For example, this entry for a base mount point:

`/`

means RealServer uses the file system plug-in associated with the mount point `/` to locate the requested file. In RealServer Administrator, the **General Setup** section defines these file system mount points. The value `/` typically indicates RealServer’s default file system plug-in that locates unsecured files on local disks. So for this request:

```
<a href="http://realserver.example.com:8080/ramgen/adtag/general/start.smil">
```

RealServer locates the file with a UNIX path such as the following, depending on the directory path associated with the `/` mount point:

```
/RealServer/content/start.smil
```

On Windows, the path may look like this:

```
G:\RealServer\content\start.smil
```

Using Authentication with Ad Streaming

If RealServer contains secure content, authorization is verified only on the initial request URL, not when RealServer accesses the file through the base mount point. This creates a security risk if ad streaming requests are unsecured, but secure content resides below the directory defined by the base mount point. If your RealServer hosts secure content, but ad streaming requests are unsecured, make sure the ad streaming base mount point does not lead to a secured directory.

Security Risk Example

To illustrate how a security hole can occur, suppose the ad streaming mount point uses a base mount point of `/`, which is defined in the RealSystem Administrator’s **Mount Points** section as this path:

```
/RealServer/content/
```

If this path leads to a secured directory such as:

```
/RealServer/content/protected/
```

someone can access content in this directory through the ad streaming system by using a URL such as this:

```
<a href="http://realserver.example.com:8080/ramgen/adtag/general/protected/start.smil">
```

This URL uses the Ramgen (/ramgen/) and ad streaming (/adtag/general/) mount points, but no security mount point. Here, /protected/ is not a mount point, but the directory below the base mount point directory. Because the URL has no security mount point, RealServer does not validate the request before accessing the file in this path:

```
/RealServer/content/protected/start.smil
```

To prevent security problems, keep unsecured and secured content in separate paths. For example, you might use these mount points for unsecured and secure content:

```
/
/secured/
```

These mount points might lead, respectively, to these paths on UNIX:

```
/RealServer/content/
/RealServer/secure/
```

or these paths on Windows:

```
G:\Program Files\Real\RealServer\Content
G:\Program Files\Real\RealServer\Secure
```

A security risk is not present because the unsecured directory path does not lead to the secured directory path. For information on secure directories and authenticated content, refer to Chapter 15, “Authenticating RealServer Users”.

Creating Ad Streaming Mount Points

The mount point /adtag/general/ is predefined. You can modify this mount point, as well as create new mount points.

- To create a new ad streaming mount point:
 1. In RealSystem Administrator, click **Advertising**. Then click **Ad Serving**.
 2. Click **Add New**. This creates a new mount point with a predefined name.

3. In the **Edit Mount Point** box, change the new mount point name to any name you prefer. This mount point, which will appear in request URLs, should have a format similar to this:

/adtag/tech/

Tip

Ad streaming mount points can use names such as /generalads/, /sportsads/, and /techads/. But names like /adtag/general/, /adtag/sports/, and /adtag/tech/ help RealServer to run more efficiently, and make it easier to recognize ad mount points based on the consistent presence of /adtag/.

Additional Information

For the background on ad streaming mount points, see “Understanding Ad Streaming Mount Points” on page 316.

4. Click **Edit** to update the mount point name.

► To define an ad streaming mount point:

1. Select the mount point in **Ad Mount Points** window.
2. For **Description**, type any phrase that describes the ad mount point. You might use “Sports Ads” or “Tech Ads,” for example.
3. In the **Base Mount Point** field, specify the mount point where RealServer locates the requested file. For unsecured content, this is typically the mount point “/”.

Additional Information

See “Choosing the Ad Streaming Base Mount Point” on page 317 for more information.

4. For **Target HTML**, enter a fully-qualified target URL where RealServer finds the ad URL to use. Each ad mount point typically has a unique URL through which it directly integrates with an ad serving system, or requests a Web-based HTML page containing ad URLs. After you click **Apply** to update RealServer, **Test URL** links your browser to the given URL.

Additional Information

See “Getting Ad URLs from an Ad Server” on page 310.

See also “Overriding Mount Point Settings through SMIL” on page 325 for details on overriding where RealServer gets ad URLs.

5. The **Ad Server Type** pull-down menu lets you select the type of system that supplies the ad URLs. Your RealServer offers the following, and possibly more, choices:
 - Default
 - Type 1
 - Type 2
 - DoubleClick
 - NetGravity
 - Engage
 - AdForce

The **Ad Server Type** setting affects the clickthrough URL of an ad that appears in RealPlayer. If you use the DoubleClick ad serving system, for example, choose the **DoubleClick** option. When you choose one of the named ad serving systems, make sure you have integrated RealServer with that system as described in “Integrating RealServer Directly with an Ad Server” on page 312.

If you do not use one of the named ad serving systems, choose the **Default** setting and finish configuring the mount point as described in this section. Apply the changes, and use RealPlayer to request ad content through the mount point. Click the ad to verify that it takes you to the correct Web page for the ad sponsor. If the clickthrough does not work, choose **Type 1**, apply the changes, and try again. If that doesn’t work, redo the procedure using **Type 2**. Once clickthroughs work, choose the same type setting for all mount points used with that ad serving system.

Additional Information

For background on this option, see “Why are There Different Ad Server Types?” on page 321.

6. Most users should leave **Resolve Relative URLs** set to **Yes**. In this case, RealServer resolves any relative ad URLs, sending fully qualified URLs to

RealPlayer. This prevents RealPlayer from mistakenly requesting content from the wrong server. This setting does not affect fully qualified URLs returned to RealServer.

Set **Resolve Relative URLs** to **No** only if your RealServer hosts the content specified by relative ad URLs. For example, your RealServer might host the GIFs used for rotating banner ads. The ad serving system can then return relative URLs that RealServer sends to RealPlayer. In this case, RealServer doesn't need to resolve the relative URLs because it hosts the content.

7. Enter information in the **Rotating Banner Ads** section if you want to set up rotating banner ads. The section "Setting Up Rotating Banner Ads" starting on page 322 explains these fields in detail.
8. Click **Apply** to update RealServer with the new mount point information.
9. Put the changes into effect by clicking the **Restart** icon at the top of RealSystem Administrator.

Why are There Different Ad Server Types?

For the **Ad Server Type** field, RealServer presents several name-brand ad serving systems, as well as the three generic options: **Default**, **Type1**, and **Type 2**. These options are present because different ad serving systems handle clickthrough URLs differently. Some ad serving systems return the advertiser's clickthrough URL with the ad image URL. For example, a clickthrough URL **http://www.real.com** may accompany a RealNetworks ad.

Other ad servers do not return the advertiser's clickthrough URL initially, however. Instead, they record RealServer's IP address and a user agent ID when it requests an ad. Rather than pointing to the advertiser's Web site, the clickthrough URL points back to the ad server and includes RealServer's IP address and ID. This type of URL lets the ad server log the clickthrough before redirecting the request to the advertiser's Web site.

In this case, an error may occur on the ad server because RealPlayer handles clickthroughs rather than RealServer. RealPlayer's IP address will not match RealServer's IP address, so the ad server will not recognize RealPlayer as the client that received the ad. To correct this, RealPlayer routes the clickthrough request through RealServer. Because RealServer then acts as RealPlayer's proxy, the ad server recognizes the clickthrough attempt.

The settings **Default**, **Type 1**, and **Type 2** cover three possible ways that RealServer can interact with ad servers. It is easier to try out the three

possibilities until you find that option that makes clickthroughs work correctly than to research how your ad serving system handles clickthroughs.

Setting Up Rotating Banner Ads

When you set up a mount point to stream banner ads in JPEG or GIF format, you can use RealServer's rotation feature to insert fresh ads into the SMIL presentation at regular intervals. You might stream a new ad image to RealPlayer every 30 seconds, for example. The following are the options you set in the **Rotating Banner Ads** section of the ad streaming mount point dialog.

Additional Information

The SMIL generation feature or the requested SMIL file lays out the banner ads with the streaming clips. For more on SMIL generation, see "Generating SMIL Files for Ads" on page 326. The *RealSystem Production Guide* explains how to lay out banner ads manually with SMIL.

Rotate

Select **On** in this pull-down menu to turn on banner ad rotation for the mount point.

Interval

This is the frequency in seconds that new banner ads appear in RealPlayer. Make sure that the interval and the bit rate work for the ad size. With a bit rate of 1000 and an interval of 30, for example, RealServer can stream 30,000 bits (3.6 Kilobytes) during each 30-second interval. This may not be enough for some banner ads.

Bitrate

RealServer streams banner ads to RealPlayer at this rate, which is in bits per second (bps). Keep in mind that the ad bit rate is part of the presentation's overall bit rate. If you want to deliver a 20 Kbps video over 28.8 Kbps modems, for example, do not use a 4000 bps ad stream. The total presentation bit rate of 24 Kbps is too high when you take into account the overhead reserved for network congestion, packet loss, and so on.

The following table lists some common interval times and banner ad sizes, showing the *minimum* bit rate required for each combination. For instance, the

table shows that 9-Kilobyte ads rotated every 30 seconds need to stream at a minimum of 2460 bits per second.

Minimum Ad Bit Rate Needed for Specific Intervals and Ad Sizes

Interval	Bit Rate for 6-Kilobyte Ads	Bit Rate for 9-Kilobyte Ads	Bit Rate for 12-Kilobyte Ads	Bit Rate for 18-Kilobyte Ads
15	3280	4920	6560	9840
30	1640	2460	3280	4920
45	1100	1640	2200	3280
60	820	1230	1640	2460
90	550	820	1100	1640
120	410	620	820	1240

Additional Information

The bandwidth chapter in *RealSystem Production Guide* explains how to target various connection speeds. To view this guide, click **Resources** under **Help** in RealSystem Administrator.

Startup Image

In this optional field, you can enter the location of an image to display before the first ad streams to RealPlayer. The purpose is simply to fill the ad banner region until the first ad appears in RealPlayer. If you don't use a start-up image, the ad region remains blank until the first ad arrives. The start-up image streams at the same bit rate you select for the rotating ads. Do the following to make the start-up image appear as quickly as possible:

- Keep the start-up image as small as possible. The image should be a small, non-animated GIF or JPEG.
- Stream the start-up image from RealServer rather than another server. Although not required, this minimizes delays caused by connection latency. If a start-up image named `start.gif` is in RealServer's main content directory, for example, enter this for **StartUp Image**:

`/start.gif`

Changing Timeouts Values

RealServer uses two timeout values, each set by default to 5 seconds:

- **Connection Timeout**

This value determines the number of seconds that RealServer waits for a response from an ad serving system or a Web server when requesting ad URLs.

- **Server Timeout**

This value determines the number of seconds that RealServer waits for the ad server or Web server to return the ad URLs after the connection is made.

All timeouts are recorded in the RealServer error log. (The error log is described in “Error Log” beginning on page 303.) If ads consistently fail to appear in RealPlayer, the server providing the ad URLs may not be responding fast enough to avoid a timeout. You should first try to fix this latency by using a faster Web or ad server, or increasing the speed of the connection between RealServer and the other server. If ad retrieval still times out, increase the two timeout values by increments of one second until ad retrieval consistently works. Consider the following points when raising the timeout values:

- The higher the timeout value, the longer RealServer waits for a response and, correspondingly, delays the presentation. You need to balance the requirements for retrieving ads against viewers’ expectations that presentations play back with minimal delay.
- In its preferences, each RealPlayer has a connection and a server timeout period set by default to 20 and 90 seconds, respectively. These values determine how long RealPlayer waits for RealServer to respond to its requests. Your RealServer timeout values should always be below these values. Each RealPlayer user can change these values, so some users may have set the values lower.

► To change the RealServer timeouts:

1. In RealSystem Administrator, click **Advertising**. Then click **General**.
2. Change the value in the **Connection Timeout** or **Server Timeout** field, or both. Do not set a value lower than the default of 5. Do not set a value higher than necessary to ensure that ad retrieval works consistently.
3. Click **Apply** to update RealServer with the new timeout information.
4. Put the changes into effect by clicking the **Restart** icon at the top of RealSystem Administrator.

Overriding Mount Point Settings through SMIL

RealServer lets content creators override certain ad mount point settings. Through SMIL, content creators can specify banner ad rotation parameters, as well as where RealServer gets ad URLs. If your RealServer hosts clips for many content creators who use different ad types, this override feature lets you satisfy the creators' different needs without setting up separate ad streaming mount points for each ad type. For example, instead of setting up different banner ad mount points for different audiences, you can set up one mount point, then use SMIL to point RealServer to different target URLs. Each target URL then returns ad URLs for a different audience.

Note

RealSystem Production Guide does not explain the override features. As the RealServer administrator, you are responsible for informing content creators of the override features and their acceptable values.

Overriding the Target URL Location

As described in “Creating Ad Streaming Mount Points” on page 318, you specify where RealServer gets ad URLs when you define each ad streaming mount point. RealServer then requests an ad URL for each `<RealAdInsert/>` tag included in the SMIL file. Content creators can specify a different target that provides ad URLs, however, by including an `AdURL` attribute in the `<RealAdInsert/>` tag:

```
<RealAdInsert region="adbanner" AdURL="http://www.example.com/ads.html"/>
```

In this case, RealServer requests the URL specified by the `AdURL` attribute, not the target defined through **Target HTML** in the mount point dialog. Keep in mind, though, that the `AdURL` target must provide ad URLs that work with the mount point's remaining settings. For example, the `AdURL` target should not return URLs to streaming video ads if the mount point is set up for rotating banner ads. Nor should it target an ad serving system other than the one defined for the mount point.

Overriding Banner Rotation Settings

“Setting Up Rotating Banner Ads” on page 322 explains how to configure an ad streaming mount point for rotating banner ads. To include these ads in a SMIL presentation, content creators use a `<RealAdInsert/>` tag like this:

```
<RealAdInsert region="ad_banner" dur="9min"/>
```

This tag specifies the SMIL region where the ads appear and how long RealServer sends ads to RealPlayer. Normally, the tag does not specify any ad rotation parameters, which are set instead in the ad streaming mount point. As described above, however, content creators can specify where RealServer gets the banner ad URLs. As well, content creators can override any of the banner ad mount point's **Interval**, **Bitrate**, and **Startup Image** settings:

```
<RealAdInsert region="ad_banner" dur="9min" Interval="60" Bitrate="2460"
StartupImage="/start.gif" AdURL="http://www.example.com/ads.html"/>
```

This sample tag overrides the ad mount point's rotation settings with a new ad target URL, a new start-up image, a rotation interval of 60 seconds, and a bit rate of 2,460 bits per second.

Generating SMIL Files for Ads

RealServer can automatically generate a SMIL file that inserts an ad or series of ads in each streaming presentation. This works for both single clips and existing SMIL files. If your RealServer hosts a large number of RealAudio clips, for example, you can simply have RealServer generate a SMIL file that lays out ads for each clip. Content creators then do not need to write SMIL files or include `<RealAdInsert/>` tags in existing SMIL files.

Note

Automatic SMIL generation works in conjunction with ad streaming as described in "Configuring RealServer to Stream Ads" on page 315, and you must set up ad streaming mount points first.

Limitations on Automatic SMIL Generation

Although automatic SMIL generation works in a large number of ad streaming scenarios, it does not provide all the flexibility possible when content creators write SMIL files that include `<RealAdInsert/>` tags. SMIL generation has these limitation:

- Each requested clip or SMIL file can include only one ad. You can have a rotating banner ad that refreshes every 30 seconds, for example, but you cannot have two banner ads.
- Interstitial ("commercial break") ads are not supported.

- With rotating banner ads, RealServer assumes the generated SMIL file is for a live broadcast, and it disables RealPlayer's clip position slider. For more on this, see the section on rotating banner ad durations in *RealSystem Production Guide*.
- Content creators cannot change ad streaming mount point parameters as described in "Overriding Mount Point Settings through SMIL" on page 325.

Understanding SMIL Generation Mount Points

Like ad streaming, automatic SMIL generation works through mount points included in the content request URL. The SMIL generation mount points always work in tandem with ad streaming mount points, which are described in "Understanding Ad Streaming Mount Points" on page 316. For example, a Web page hyperlink to a media clip or SMIL file on RealServer may look like this:

```
<a href="http://realserver.example.com:8080/ramgen/adtag/general/  
smilgen/banner/video/video.rm">
```

Here, the ad streaming mount point, `/adtag/general/`, determines the type of ad used with `video.rm`. The SMIL generation mount point, `/smilgen/banner/`, trails the ad streaming mount point in the URL. It causes RealServer to create a SMIL file that lays out a `<RealAdInsert/>` tag along with the video clip. If a SMIL file rather than a clip were requested, RealServer would create a new SMIL file that includes the contents of the requested SMIL file along with a `<RealAdInsert/>` tag.

A mount point such as `/smilgen/banner/` might define a layout for a banner ad that is 468 pixels wide by 60 pixels high and appears above the requested content. You can define any number of SMIL generation mount points, such as `/smilgen/lead_in/` or `/smilgen/banner_below/`, for different ad layouts. This lets you support any number of ad types, whether banner ads or streaming media ads, through SMIL generation.

When you define a SMIL generation mount point, it must be relative to the base mount point of the ad streaming mount point used with it. For example, in this request URL:

```
<a href="http://realserver.example.com:8080/ramgen/adtag/general/smilgen/  
/banner/video/video.rm">
```

`/adtag/general/` is the ad streaming mount point. If this mount point uses the default file streaming mount point (“/”) as its base mount point, you simply define the SMIL generation mount point like this:

```
/smilgen/banner/
```

If the ad streaming mount point uses a base mount point such as `/local/`, however, you need to include this base mount point in the SMIL generation mount point definition:

```
/local/smilgen/banner/
```

This causes the SMIL generation feature to intercept the file access attempt caused by the ad streaming mount point. The actual file access then occurs through the SMIL generation mount point’s base mount point. Note that in the example above, `/local/` does not appear in the request URL. It appears only in the SMIL generation mount point dialog.

Additional Information

For more on the ad streaming base mount points, see “Choosing the Ad Streaming Base Mount Point” on page 317.

Creating SMIL Generation Mount Points

SMIL generation mount points for lead-in ads, banner ads, and rotating banner ads are predefined. You can modify these mount points or create new ones.

► To create a SMIL generation mount point:

1. In RealSystem Administrator, click **Advertising**. Then click **Ad SMIL Generation**.
2. Click **Add New**. This creates a new mount point with a predefined name.
3. In the **Edit Mount Point** field, change the new mount point name to any name you prefer. This mount point, which will appear in request URLs, should have a format similar to this:

```
/smilgen/banner_above/
```

Tip

SMIL generation mount points can use names such as `/smil_lead/` and `/smil_banner/`. But names like `/smilgen/lead_in/` and `/smilgen/banner_below/` help

RealServer to run more efficiently, and make it easier to recognize mount points based on the consistent presence of /smilgen/.

Note

SMIL generation mount points are relative to ad streaming base mount points. See “Understanding SMIL Generation Mount Points” on page 327.

4. Click **Edit** to update the mount point name.
- To edit a SMIL generation mount point:
1. Highlight the mount point in the **SMIL Mount Points** window.
 2. For **Description**, enter a phrase that describes the ad mount point. You might use “Bottom Banner SMIL Generation” or “Lead-in Video SMIL Generation,” for example.
 3. In the **Base Mount Point** field, enter the mount point where RealServer locates the requested file. For unsecured content, this is typically the mount point “/”.

Note

SMIL generation base mount points have the same security issues related to the base mount points for ad streaming. See “Using Authentication with Ad Streaming” on page 317 for more information.

4. Fill in the options for SMIL generation in the remainder of the dialog.

Additional Information

See “Setting SMIL Options” on page 330 for descriptions of these options.

5. Click **Apply** to update RealServer with the new mount point configuration.
6. Put the changes into effect by clicking the **Restart** icon at the top of RealSystem Administrator.

Setting SMIL Options

The following are the SMIL generation options you can set for each mount point.

Ad Type

This pull-down menu determines the type of ad used. You can set one of these values:

- Banner** Banner ad that appears alongside requested content. For **Ad Position**, choose Top, Bottom, Left, or Right.
- Leadin** Ad that appears before the requested content begins playback. This ad is usually in a format such as RealVideo or Flash. The **Ad Position** value is typically Center.
- Rotating Banner** Rotating banner ad that appears alongside requested content. The **Ad Position** value should be Top, Bottom, Left, or Right.

Ad Position

This pull-down menu determines the ad's location relative to the requested content. It can have one of the following values:

- Top** Ad appears above the requested content. The ad and content are centered horizontally within the RealPlayer window.
- Bottom** Ad appears below the requested content. The ad and content are centered horizontally within the RealPlayer window.
- Center** Ad appears centered and in front of the requested content. The ad and content are thus centered both horizontally and vertically. In this case, the **Ad Type** value should be Leadin. Otherwise the ad appears in front of the content as the content plays.
- Left** Ad appears to the left of the requested content. The ad and content are centered vertically within the RealPlayer window.
- Right** Ad appears to the right of the requested content. The ad and content are centered vertically within the RealPlayer window.

Ad Width and Ad Height

In these fields, set the pixel width and height, respectively, of the ad included with the request. RealServer uses these values to lay out the ad in the SMIL file. Make sure that the ad serving system returns URLs to ads of this size.

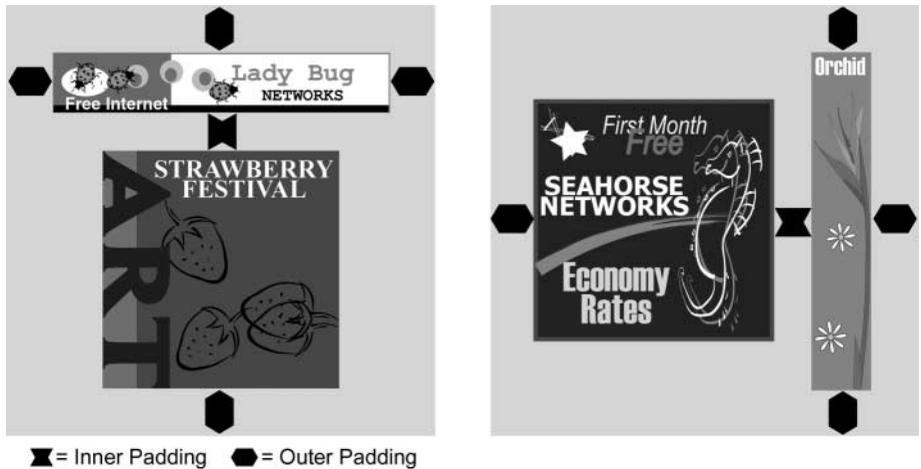
Additional Information

See "Getting Ad URLs from an Ad Server" on page 310.

Inner Padding and Outer Padding

The outer padding determines how many pixels of space RealServer adds as a border around all clips. A value of 20, for example, adds 20 pixels of outer padding. The inner padding sets the distance in pixels between the ad and the requested content. It is ignored if the ad is centered in front of the requested content.

Inner and Outer Padding Examples



Suppose a banner ad appears above the content and is wider than the content, as illustrated in the left image above. If the ad is 468 pixels wide, an outer padding value of 5 makes the RealPlayer window 478 pixels wide. The height of this window is:

- the height of the ad
- plus the height of the content (height of clip or SMIL file root-layout)
- plus the InnerPadding value
- plus 10 pixels (5 pixels of outer padding at top and bottom)

Background Color

This field sets the background color for the RealPlayer window. Empty space around the ad and content appears in this color. To specify a color value, use any RGB hexadecimal value (#RRGGBB), or one of the following predefined color names, listed here with their corresponding hexadecimal values:

white (#FFFFFF)	silver (#C0C0C0)	gray (#808080)	black (#000000)
yellow (#FFFF00)	fuchsia (#FF00FF)	red (#FF0000)	maroon (#800000)

lime (#00FF00) olive (#808000) green (#008000) purple (#800080)
aqua (#00FFFF) teal (#008080) blue (#0000FF) navy (#000080)

Enable Playlist

This field determines whether the viewer has access to the RealPlayer playlist during a lead-in ad. It does not affect banner ads. Set this field to Yes to allow the viewer to skip the lead-in ad clip.

This chapter covers general troubleshooting steps to take if something goes wrong in RealServer. Messages—both informational and error-related—are also described.

Overview

If you encounter problems when running RealServer, you can narrow down the problem with the following tasks:

- Determine scope of the problem—is the problem related to clients? Some clients or all clients? Is the problem on the RealServer side?
- Check the error logs—messages in the error log file (or files, if you’ve set up log file rolling) will direct you to the problem. For instructions on how to interpret the log file formats, see “Error Log File Format” on page 303.
- Make certain your RealServer is licensed to use the feature you have configured. Without the correct license, the correct settings won’t take effect. Read “License Files” on page 38 for a list of features that can be licensed.

General Troubleshooting Steps

These steps are good ones to check whenever you have trouble with any RealServer features.

First, isolate the problem. Is the problem on the RealServer end, or the client end? Or is there difficulty with the production tools?

Step 1: Make sure RealServer is running.

When you started RealServer, were there any error messages? If so, look up the message in the index of this document.

I can't start the Server at all.

There are several possible causes of the Server not starting:

- If you are running Windows NT, the Server is automatically installed as a service, which means that it runs automatically. If it is installed as a service, and you try to start the Server using any other method, it appears not to start. An error message may appear. To find out if it is already running, click **Start>Settings>Control Panel> Services** and look for RMServer in the list; the word “Started” in the Status field indicates that it’s running.
- If you are running UNIX, make sure you are logged on with the correct user name. The Server requires the use of port 554, and you must be logged on as root in order to access this port. The error message “Could not open port 554” appears on screen when you try to start.
- The error message “Could not open port 7070” indicates that either other software is using the port, or RealServer could not bind to the necessary address. See the next item for instructions on binding to a particular address.
- You may need to bind RealServer to a specific IP address. This is often the case when you receive the error message “Server not responding properly: Heartbeat check disabled”. (Heartbeat check is a self-monitoring feature which ensures that the RTSP port is available.) See “Binding to a Specific Address” on page 334.
- RealServer may be bound to an address that doesn’t exist. Using the information in “Binding to a Specific Address”, either delete the IPBindings section, or change it to use the single 0.0.0.0 address.

Binding to a Specific Address

Open the configuration file. If this is a new installation of RealServer, and the configuration file has not been customized, you will need to add the following text to the configuration file. The configuration file is named `rmserver.cfg`, and is located in the RealServer main directory. Add this text to the very end of the file:

```
<List Name="IPBindings">  
  <Var Address_01="0.0.0.0"/>  
</List>
```

The address 0.0.0.0 binds the Server to all IP addresses available on this machine. You can substitute the machine’s actual address, instead.

Determining the IP Address of Your Computer

Use the appropriate method for your operating system:

- **Windows NT**—Click **Start>Run>Command Prompt**. At the prompt that appears, type `ipconfig`.
- **UNIX**—Most UNIX platforms will report the IP address if you use the command `ifconfig`.

When I click the Server icon, the command window opens briefly but then disappears.

Rather than remaining visible, the window closes if the Server encounters an error. Use the following steps to find out what the error is:

1. Open a command prompt.
2. Move to the Bin directory.
3. Start the Server by typing

```
Bin\rmsserver rmsserver.cfg
```

The Server will attempt to start, and any error messages will appear on screen. The most frequent causes of this type of problem are an expired license or conflicting port use.

Also, compare your system date to the Issue and Expire date shown on the **About** page of RealSystem Administrator, and make sure your system date is accurate.

I can start the Server, but I can't connect to it.

If you know the Server is running, but you aren't able to play any content from it, the ports may be unavailable. To find out, go to a Web browser and type the following:

```
http://address:port
```

where *address* is the name or IP address of your Server, and *port* is one of the ports that the Server uses: either 554, 7070, or 8080.

A response appears in the browser, such as "File not found."

If you do not get a response, or if you get an error message, either the Server is not running, or it needs to be bound to a specific IP address. Refer to "Distributed Licensing" on page 109 for specific instructions.

If you are not sure what IP address to use for *address*, use the instructions in "Determining the IP Address of Your Computer" on page 335.

The Server is running, but many features have stopped working.

If your license files have expired, the Server runs with minimal settings. See “Minimum Licensed Features” on page 39 for a list of the features that are always available. Contact RealNetworks or your reseller to purchase an updated license.

Step 2: Try different ways of connecting.

Once the Server is running, use the instructions in this section to narrow down the source of the problem.

Try using IP addresses, rather than DNS names.

In cases where one computer must talk with another, problems may arise if the DNS name does not resolve correctly. By using IP addresses, and binding RealServer to the correct IP address, you can eliminate DNS resolutions as a potential problem.

Play the sample files.

Try playing the sample files from the same machine as RealServer. An easy way to do this is from RealSystem Administrator. Click **Samples** in the left-hand frame. Click any of the samples shown in the right-hand frame.

Next, play the sample files from a different machine, but within your network.

If you get a message that RealPlayer needs to download a new plug-in but is unable to, your firewall may not be configured correctly. See Chapter 9, “Firewalls and RealServer” for information.

Play your files.

Try to play one of your files by typing its URL in the Open Location dialog box of RealPlayer.

- Is the media clip downloading instead of playing in real time? If so, the link in the Web page is wrong. Be sure to use Ramgen in the Web page link.
- If you can play the sample files, but not your own, you may not be creating links correctly. For on-demand, or prerecorded files, use the link format in “Linking to On-Demand Clips” on page 141. For live files, use the format shown in “Creating the Link to the Live Unicast” on page 148.

- If the client is behind a firewall, its firewall may be preventing streaming media from being sent to it. See “Communicating with Clients” on page 123.
- You may have locked everyone out—or just a range of users—with an access rule. Use RealSystem Administrator to check the access control rules: in RealSystem Administrator, click **Security>Access Control**. Review “Limiting Access with the IP Address” on page 214.
- Is there unreadable text on the screen instead of media? The Web server’s MIME types need to be configured to recognize RealNetworks data types. See “MIME Types” on page 97.
- Insufficient bandwidth may be available. Look at “I get a message saying “The Server has reached capacity”” on page 350.

Step 3: Check the production tools.

If a content creator is doing a live encode, check that he or she has used the correct RealServer information. Make sure that any virtual path he or she typed is one that will be recognized by RealServer.

- If they are including any RealServer mount points in the path, be sure they are spelling it correctly.
- Be sure to use the same file name extension in the link as you typed in the encoder. RealServer will not supply a missing or incorrect extension.
- Be certain they’re using correct port numbers. For RealProducer Plus 6 and later, they need to use the port number you specified in **Broadcasting>Encoder** page of RealSystem Administrator. For earlier encoding tools, they should use the port number shown in **Broadcasting>Pre-G2 Encoder**.

Review Chapter 4, “Sources of Content”, or consult the encoding software’s documentation.

Step 4: Check the remaining areas.

Read further in this chapter for help with specific features.

- Is the RealServer host machine address correctly configured in the network routers? If the client cannot access the Server over the network, then you cannot expect media to play. Configuring IP address and routers is a complex issue. Contact a networking specialist for help.

- Is there a firewall between the client and RealServer? Firewalls must be configured to permit media to play through them. See Chapter 9, “Firewalls and RealServer”.

Step 5: Work with the system or network administrator.

Others in your organization may have information you need, such as available port numbers, or information on bandwidth restrictions.

Troubleshooting RealSystem Administrator

How do I figure out which port number to use for RealSystem Administrator?

1. Using a text editor, open the configuration file, which is named `rmserver.cfg` and is located in the main RealServer directory, and search the file for `AdminPort`.
2. You will find an entry similar to the following (your port number will be different):

```
<Var AdminPort="7845"/>
```

Make a note of the number.
3. In your Web browser, type the following, substituting your computer's IP address for *address* and the number you found in Step 2 for *AdminPort*:
`http://address:AdminPort/admin/index.html`
4. RealSystem Administrator asks you for your user name and password. Type these and click OK.
RealSystem Administrator appears.

How do I look up my user name and password?

When you install RealServer, the setup program asks you for a user name and a password. It uses these for RealSystem Administrator and for any content creators who use version 6 and later encoding software to send material to your RealServer.

If you can't remember your password, you must reinstall RealServer, or, if you have purchased an Upgrades and Support contract, contact the RealNetworks Technical Support department (see “Contacting RealNetworks Technical Support” on page 355).

I can't start RealSystem Administrator.

- Make sure RealServer is running. RealSystem Administrator cannot start if RealServer is not running.
- You may need to add an IP Bindings section. Refer to “Binding to a Specific Address” on page 334.
- Be certain you are using the name of the machine that's running RealServer in the URL. Do not use a NetBIOS name; use the DNS name or the IP address, instead.
- Use the correct browser version. RealSystem Administrator is designed to run with Netscape 4.06 or higher, and Internet Explorer 4.01 or higher.
- If it was running before, and you have recently created new access control rules, you may have locked yourself out of RealSystem Administrator. You will need to create a new rule, by editing the configuration file, that enables access to RealSystem Administrator. See “Deciding What Rules to Create” on page 216 for an explanation of the necessary rules.

I receive JavaScript errors.

JavaScript errors are usually due to an older browser version or the wrong version of RealServer for your operating system. RealSystem Administrator is designed to run with Netscape 4.06 or higher, and Internet Explorer 4.01 or higher.

Troubleshooting On-Demand Streaming

I can't stream any on-demand content.

Problems with on-demand content are usually caused by incorrect link references. Chapter 5, “Understanding Link Formats” has detailed information on the correct place to store your content, as well as how to write the URL itself.

Common problems include:

- Using spaces in file names, instead of one word.
- Not matching the capitalization of the clip name. RealServer is case-sensitive when it looks at clip names, so be sure they are identical.
- Storing content where RealServer cannot access it. Do not store media files on the Web server.

- Referencing the RealServer Content directory from a Web server.

If you receive the error “Error retrieving URL ``file name and path'` (Invalid path)” in the error log, either of the following may be true:

- A user has clicked a link that points to a file that RealServer cannot find. If the file exists, the link is probably incorrect. Make sure that the correct mount points and virtual paths (if needed) are in the link.
- The user has an old version of RealPlayer that cannot read the newer stream formats. The user will need to upgrade before he or she can play the content.

Troubleshooting Live Unicasting

Live unicasting is ready to work as soon as you install RealServer—just remember to include `/encoder/` in your links.

I can't find my live clips. Where are they?

Live content does not exist in file format. The data packets are discarded as they are broadcast.

To see which live broadcasts are arriving at your RealServer, use the Java Monitor to look at the connections to your Server. (In RealSystem Administrator, click **Monitor**. Click the **Connections** tab.)

You can archive live broadcasts as they arrive at your RealServer, and save them for later use. See “Archiving Live Broadcasts” on page 150.

Live unicasting is not working...what should I do?

Check that the broadcast is getting to the Server. Use the Java Monitor to see which encoded streams are arriving. (In RealSystem Administrator, click **Monitor**. Click the **Connections** tab.)

From the machine on which RealServer is installed, check that you can connect to the broadcast.

Make sure you are using the correct format for the link.

- Remember to use the encoder mount point if the material is coming from RealProducer Plus 6 or later. Use the live mount point if it is earlier encoding software.
- Reference the file name as it is shown in the Java Monitor.

Make sure your access control rules aren't preventing anyone from connecting.

Troubleshooting Live Archiving

Live archiving is turned off by default. Make sure you've selected **Enabled** on the RealSystem Administrator **Broadcasting > Live Archiving** page, and that you have restarted the Server if necessary.

Other possible solutions include:

- Try using a different location in **Destination Path**.
- If you have set up live archiving to only archive a few streams, be certain that the path you used on the live archiving page matches what the content creator is typing in the encoding software path.
- Check the error log for any messages related to live archiving.

Troubleshooting G2SLTA

Be sure to run the `g2slta.bat` file (Windows) or the `g2slta.sh` file (UNIX), not the `g2slta.exe` or `g2slta` file. The `.bat` and `.sh` files set the environment variables correctly.

If you get the message "Data type not supported," you're trying to use **G2SLTA** to broadcast files that are not supported. **G2SLTA** can only deliver audio and video files.

Troubleshooting Splitting

Steps involved in troubleshooting splitting fall into two general areas:

- Whether the receiver can receive from the transmitter
- Whether a client can receive a split stream from the receiver

Splitting is one of several features controlled by the license you purchased. Click **About** in the RealSystem Administrator of each RealServer to see if each is licensed for splitting; make certain both are licensed. An error message stating that your RealServer is not licensed for splitting will also appear in the error log, if your license does not permit splitting.

Transmitter-to-Receiver Connections

Before investigating any receiver-to-client issues, be sure the transmitter-to-receiver connections are working properly.

Problems with splitting may be related to:

- The transmitter is no longer broadcasting or is unable to broadcast any clip.
- The transmitter does not “know” it’s supposed to split its broadcasts to your receiver, because you have not added the receiver to the Broadcast Receivers list.
- The access control list uses restrictive rules that prohibit receivers from making resend request, and prevents pull splitting receivers from receiving any content.

On the receiver computer, use a client to connect to the transmitter to make sure the clip exists and is being broadcasted.

Be certain that you have chosen the same transport method on both the transmitter and the receiver.

Check the error log on the transmitter for messages.

Receiver-to-Client Connections

Double-check the URL you created for the split broadcast. The link formats are complex, and creating them accurately is a common problem. You may have everything configured correctly, on both the transmitter and the receiver, but if the link is not right, users will not be able to connect.

Make sure that the receiver can receive a regular unicast from the transmitter. If unicasting is not working, splitting will not work, either. On the receiver machine, make sure you can receive the stream directly from the transmitter: in the client, connect to the transmitter using the direct URL. Use the format referenced in “Unicast Live Content (from Encoders)” on page 367.

Make certain there aren’t any access control rules on the receiver that prohibit the client from receiving any broadcast or stream.

If the receiver is using multicast to distribute the split broadcast inside the network, look for multicast user list rules that insist that the client receive the broadcast in multicast mode. If the client is not configured for multicast reception, it will not be able to receive the broadcast.

Troubleshooting Multicasting

Before setting up multicasting, two conditions must exist:

- The Server must be licensed for multicasting
- The network must be set up for multicasting

If these two conditions have been met, use the following information to troubleshoot this feature.

Steps in troubleshooting multicasting fall into two areas:

- Running the multicast on the RealServer
- Connecting to the multicast from a client

Checking RealServer

The following error messages, appearing in the error log, indicate either that you have configured a back-channel multicast in RealSystem Administrator with **Delivery Only** set to Yes (`DeliveryOnly=True` in the configuration file), or that it is a scalable multicast that does not have a backup unicast configured:

- “Multicast delivery only”
- “This server is configured to support only multicast connections. Please contact the content provider for more information on listening to this broadcast.”

Clients that are not configured for multicast will show an error message if they click a link to a scalable multicast but the user has turned off multicast in the RealPlayer preference tab. The following message appears:

- “Scalable Multicast: Your player is not configured to play multicast content.”

Make certain your license permits scalable multicast. If you configure scalable multicast, yet it is not included in your license, the following error message appears in the error log:

- “Scalable Multicast: This server is NOT licensed to deliver Scalable Multicast streams.”

The message “Error in creating Back-channel multicast session. Please increase the AddressRange configuration variable.” indicates that RealServer needs more multicast addresses in order to broadcast in back-channel multicast mode. In RealSystem Administrator, use a larger range in the IP Address Range boxes.

Special Issues with the Configuration File

If you configure back-channel multicast by editing the configuration file directly, you may inadvertently omit required sections. Without a `ControlList` section, multicasting will not work. Add it, using the format shown in “Back-Channel Multicasting Configuration Elements” on page 414, or use RealSystem Administrator to set up the Client Access List. The error message that appears if this section is missing is:

- “Back-channel multicast is enabled and the control list is empty. No clients will receive multicast. Please add a control list.”

If the configuration file is in the wrong format, or contains an error, this message appears:

- “Scalable Multicast: Initialization failed.”

If the configuration file is missing the `Sources` list (it describes which paths will be multicast), the following message appears in the error log: “Scalable Multicast: Could not find Source List in the configuration file.” Add this section using RealSystem Administrator.

Similarly, the message “Scalable Multicast: Could not find MountPoint [or AddressRange or PortRange] configuration variable.” shows that the scalable multicast section of the configuration file is missing a section. The reliable way to set up this feature is with RealSystem Administrator.

Connecting with the Client

Try to play the clip from the same system on which RealServer is installed.

Problems with multicasting may be related to:

- The network or the client not being multicast-enabled.
- Access control rules prohibit client from receiving any broadcast or stream.
- Multicast user list rules insist that the client receive the broadcast in multicast mode, and the client is not configured for multicast reception.

SDP files are generated automatically and sent to the client when the user clicks the scalable multicast link. If you change any of the settings so that they are different from those initially created in the SDP file, the client will not be able to connect. Two key elements to watch out for:

- Use the same multicast address for the broadcast. This is easier to control for an internal or intra-company multicast than for an Internet-based multicast.
- Use the same encoder settings.

Troubleshooting Access Control

Make sure you have at least three rules, so that you can continue to connect to RealSystem Administrator, as described in “Deciding What Rules to Create” on page 216.

The first rule to create is always the rule that enables you to access RealSystem Administrator! If you create another rule first, and lock yourself out of RealSystem Administrator, you will need to edit the configuration file, remove the rule manually, and then restart RealServer. See “Access Control” on page 385 for a guide on what to look for in the configuration file.

If you receive the message, “Invalid player IP Address”, it is because this RealServer is configured with access rules that prevent clients from certain IP addresses from playing content. The client that tried to request content is excluded via access rules. Access rules are described in “Limiting Access with the IP Address” on page 214.

Troubleshooting Authentication

Many of these messages relate to the player validation method of authenticating users, in which the ID of the client software, rather than a user’s name, is being verified.

Additional Information

Refer to “Player Validation” on page 233.

“Your account has been locked, contact your content provider for more information.”

Another user is already connected to this RealServer with the same user name. You can configure RealServer to allow users to log in from more than one location, using the same user name: in RealSystem Administrator, click **Security>Commerce**. In the rules list, select the rule that applies to this clip or directory. In the **Allow Duplicate IDs** list, select Yes.

“Your account has expired, contact your content provider for more information.”

There is no more time left in the user’s account. The user was playing a clip when the account’s time limit was reached.

To add more time to the user’s account, or to change the type of permissions available for that clip or directory, see the instructions in the “Permission Types” table on page 230 for values.

“You must register your RealPlayer before viewing this content. Please contact your content provider for assistance.”

The client is not authorized to play the content; applies to user-authenticated content.

“This player doesn’t support user authentication”

An old client, incapable of authentication, tried to access secure content.

Additional Information

For a list of client software versions that can be authenticated, refer to “Compatible Client Versions” on page 224.

Troubleshooting Monitoring

A message in the Java Monitor indicating that the Server is unavailable can mean the Server is not running, or that an access control rule does not permit access to the Monitor Port number.

Troubleshooting Ad Streaming

If your RealServer is not licensed for ad streaming, and a client accesses your RealServer with an ad insertion URL, the following message will appear in the error log:

- “Ad Application: Ad Insertion failed! This server is NOT licensed for automatic ad insertion. Please contact RealNetworks to obtain a license for this feature.”

You can verify the features for which your RealServer is licensed by clicking **About** in RealSystem Administrator.

Incorrect <RealAdInsert/> information in a SMIL file will cause this error message to appear in the error log:

- “Ad Application: The Ad Insertion Plugin was unable to properly read the contents of a <RealAdInsert/> tag. Automatic Ad Insertion failed.”

Consult *RealSystem Production Guide* for correct syntax of this tag. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

Additional Information

To view this manual, click **Resources** under **Help** in RealSystem Administrator.

If the HTML used to convey the ad URL to RealServer does not contain an anchor tag that contains an IMG SRC entry, the following message appears:

- “Ad Application: No appropriate Ad anchor was found in the web page retrieved by the automatic Ad Insertion system. No Advertisement will be used. Please verify that the following AdURL contains an anchor tag containing an image source URL: “

The following messages refer to files and images related to RealServer’s ability to access its Target HTML URL. You can troubleshoot them by using the same computer on which RealServer is running, clicking the link, and checking whether the error message still appears. If the error does not appear, the problem is not related to RealServer configuration.

If RealServer cannot access its Target HTML URL, this message, along with the link, appears in the error log:

- “Ad Application: The AdURL specified in the Ad Insertion section of the configuration file could not be retrieved. Please verify that the following AdURL points to a valid web page: *URL is given here.*”

If RealServer is unable to make connection to the server specified for Target HTML, this message, and the link, appears in the error log:

- “Ad Application: The connection to the AdURL timed out because the web server did not respond to the initial connection. Please verify that the RealServer has a good connection to the following AdURL: *URL is given here*“

If RealServer and the server (used for the Target HTML) have a connection, but the server has not sent its data in a timely way to RealServer, this message appears in the error log:

- “Ad Application: The connection to the AdURL timed out because the web server stopped sending data for too long. Please verify that the RealServer has a good connection to the following AdURL: *URL is given here*”

If RealServer cannot retrieve an image, this message appears:

- “Ad Application: Error retrieving the following image: *URL is given here*”

Special Issues with the Configuration File

If you configure the ad streaming feature by editing the configuration file directly, you may inadvertently omit required sections. The ad streaming feature is designed to be configured and modified by RealSystem Administrator, and not by direct editing of the configuration file.

If you edit the configuration file incorrectly, you may receive the following error messages:

- “Ad Application: No HTTP file system mount point was specified in the configuration file”
- “Ad Application: No AdRetrievalMountPoint was specified in the Ad Insertion section of the configuration file. Unable to retrieve an Advertisement for automatic Ad Insertion.”
- “Ad Application: No RotationMountPoint was specified in the Ad Insertion section of the configuration file. Unable to insert a live rotating banner advertisement.”
- “Ad Application: No AdURL was specified in the Ad Insertion section of the configuration file. Unable to retrieve Advertisement for automatic Ad Insertion.”

Use RealSystem Administrator to configure the ad streaming feature.

Troubleshooting SMIL File Issues

Many of these error messages refer to the Image Source tag. These options are not SMIL parameters, but extensions to the image’s SMIL source tag. They are described in *RealSystem Production Guide*. To view this manual, click **Resources** under **Help** in RealSystem Administrator.

In general, a message that includes the phrase “URL-encoded” refers to those additional commands that can be typed as part of the image tag in a SMIL file.

“GIF [or JPEG]: Bad URL-encoded bitrate.”

You used a command to set the image bit rate, and used non-numeric text for the bit rate. The format for this command is the following:

```
image.gif?bitrate=value
```

If you use 0 for value instead of a Kbps value, one of the error message “GIF [or JPEG]: URL-encoded bitrate is zero.” appears.

“GIF [or JPEG]: Bad URL-encoded url.”

You included the URL command, but omitted a value. It’s easy to do, especially when you typed another tag immediately after it:

```
image.gif?url=&bitrate=12000
```

“GIF: Bad URL-encoded background color.”

You used the bgcolor option to override a GIF transparency color, such as:

```
image.gif?bgcolor=value
```

and used an incorrect value or format for bgcolor. The format for bgcolor must be RRGGBB, where RR, GG, and BB are hex values for red, green, and blue, respectively.

“GIF [or JPEG]: Bad URL-encoded target.” or “GIF [or JPEG]: URL-encoded target must either be _player or _browser”

You included the target command in the tag, but omitted to give a value for target. This can be overlooked when you type another command immediately after it:

```
image.gif?target=&bitrate=12000
```

The correct format is:

```
image.gif?target=value
```

where value is either _player or _browser.

“GIF [or JPEG]: Bad URL-encoded reliable flag.”

You typed a command such as:

```
image.gif?reliable=value
```

and used an incorrect value for `reliable`. The value for `reliable` must be either `true` or `false`.

“GIF: Unknown player command in URL-encoded url attribute.” or “JPEG: Unknown player command in url URL encoding.”

You typed a command such as:

```
image.gif?url=command:value
```

and used an incorrect value. The word following command must be `stop`, `seek`, and so on.)

“GIF [or JPEG]: Illegal time formatting in URL-encoded seek time.”

You used the image source tag option `command:seek(time)` and gave a value for *time* that is greater than the length of the timeline.

“GIF [or JPEG]: Cannot target browser with a player command.”

You combined the `target_browser` command and a client `seek` command, but they cannot be used together.

```
image.gif?target=_browser&url=command:seek(21)
```

Troubleshooting Other Issues

I get a message saying “The Server has reached capacity”

There are two causes for this message:

- Your RealServer has reached its connections limit. To see how many clients you are allowed to serve simultaneously, check your license. See “License Files” on page 38.
- You set Maximum Bandwidth to a large number, and that amount of bandwidth is not available on your network, though it may be available in your license.
- You left Maximum Bandwidth at the default value of 0. The number zero in this case actually means “up to the limit of the licensed number of streams”. Your network may not accommodate that many streams, even though your RealServer license allows it. You can change this value by choosing **General Setup>Connection Control** in RealSystem Administrator.

I get a message stating “License exceeded”

The number of simultaneous clients connected to this RealServer has reached its licensed limit. The number of connections to RealServer is limited by your license key file. To read this file, start RealSystem Administrator and click **About**. License information appears in a second browser window.

- To upgrade your license so that you can host more simultaneous connections, contact RealNetworks or your reseller.
- If you think the number of connections being served by RealServer is less than the number shown by your license, your license may have expired or RealServer is unable to start using the settings you’ve selected, and is starting with minimum settings.

Additional Information

A table of these minimum values is shown in the “License Files” on page 38.

- You may have reserved all the available connections through the ISP hosting feature. Reserved connections are not available for general use, even if no users are accessing content through the ISP hosting feature. See “Connections Available for Each Account” on page 256.

Troubleshooting Problems in RealPlayer

Users get the contents of Ram files, instead of launching the Ram files

Set the MIME types on your Web server to recognize Ram files. See “MIME Types” on page 97 and your Web server documentation.

Users get “File not found” error message in browser

You omitted /ramgen/ from the link in a Web page. The Web server, rather than RealServer, is looking for the file. Only RealServer knows where to find the file. Add the word /ramgen/ to the link. See Chapter A, “Summary of Link Formats” for a list of correct syntax.

Or, create a Ram file and point the link to the Ram file. You will need to store the Ram file in a location where the Web server can access it.

Check to make sure that Ramgen is still on the HTTP Delivery list: in RealSystem Administrator, click **General Setup>HTTP Delivery** and add /ramgen if it is missing.

Users get messages telling them their software is not the right version.

Messages in this section refer to the feature that lets you choose which client versions can receive your content. Use this feature if your content takes advantage of newer RealSystem features, such as SMIL.

You may have restricted RealServer to serving to only certain versions of RealPlayer. See “Limiting Access by RealPlayer Version” on page 213 and MinPlayerProtocol in “Allowance” on page 386.

If you have limited your RealServer to serving in this way, consider posting a note on your Web site so that users know to expect these messages.

The following messages can appear:

- “Invalid Player” or “Invalid version”
- “Please download a new RealPlayer from <http://www.real.com> to receive this content.”
- “You have connected to a RealMedia Server that only supports players newer than the one you are using. Please check on the Web Site you accessed this clip from for details on what players are supported. To download the latest RealPlayer, point your Web Browser to <http://www.real.com>”
- “You need to obtain a new player to play this clip. Please point your web browser to <http://www.real.com> and download the latest RealPlayer from RealNetworks. Once you have installed it you should try this clip again.”

Users get messages telling them they need RealPlayer Plus, not just RealPlayer.

In RealSystem Administrator, **RealPlayer Plus Only** is On. In the configuration file, the PlusOnly variable is set to True. If you want your content to be available only to RealPlayer Plus clients, leave the settings as is. If you want to allow any clients to be able to view your content, change the setting to Off or False.

- “Player Plus only”
- “The content you requested is available exclusively to RealPlayer Plus owners. Please point your web browser to <http://www.real.com/> for upgrade information.”

Users get messages about “insufficient bandwidth”

The client does not have enough bandwidth to satisfy any bandwidth negotiation choice.

If you have many clients who receive this message, consider asking content creators to include lower bit rates when they encode their content. If possible, you might want to use multicasts instead.

Users get messages telling them that they shouldn't use PNA or PNM.

These messages indicate that a link begins with `pnm://`, but it should begin with `rtsp://`. Since only RealAudio, RealVideo, RealEvents, and RealFlash can be streamed with the PNA protocol, any link that begins with `pnm` must point to one of those data types.

One of the following is the problem:

- A user typed a URL in the Open Location box of RealPlayer and used the wrong protocol. The user needs to type `rtsp` instead of `pna`.
- The link which the user clicked is incorrect. It begins with `pnm` instead of `rtsp`.
- A Ram file or a SMIL file contains a link that uses the wrong protocol.

In the last two items, the content creator must correct the link.

Messages appear as the following:

- “Please make sure you have downloaded the latest RealPlayer from <http://www.real.com> and try this clip again using `rtsp://` instead of `pnm://` in the URL.”
- “PNA unsupported for requested data type”
- “The file you requested cannot be streamed using the PNM protocol.”

This message appears: “You cannot receive this content. Either your network bandwidth is not fast enough to receive this data or your CPU is not powerful enough to decode it.”

There are two causes for this message:

- Your system just meets the minimum system requirements
- Client network preferences are misconfigured

For a list of the necessary system requirements for RealPlayer, consult the client documentation. Every enhancement you make to your system will improve the user experience.

Under some circumstances, the client will work even though the system requirements are not met. This can happen because each renderer plug-in uses

a different amount of system resources. It is also possible that the minimum requirements are met by a given system, yet a certain renderer will not perform well.

Variables such as the encoded bandwidth of the clip, actual connected modem speed, and the bandwidth setting in RealPlayer must match.

Common Mistakes to Avoid

RealServer has many ways to assure that users will receive the highest-quality performance possible. But even if your Server is configured correctly, it is still possible to deliver content in ways that is not the best quality your system is capable of. This section lists common mistakes administrators have made, that cause users to receive poor quality streams.

This section assumes your RealServer is set up correctly and is running well. You might want to share this information with the content creator.

Write links so that users download your clips, rather than stream them.

Users will download content if you do either of the following:

- If the link is to a single clip, refer to the clip directly and omit the ramgen mount point
- In a ram file or SMIL file, use http for the protocol for individual links, rather than rtsp

See Chapter 5, “Understanding Link Formats”.

If you are a firewall administrator, only allow HTTP traffic.

Users will still be able to view content, but it will arrive slowly and with lots of rebuffering. The best remedy is to modify the firewall to allow streaming media. Refer to Chapter 9, “Firewalls and RealServer”.

Create compelling content, and incorrectly use PNA as the protocol.

Referencing content created in RealSystem 6 with SureStream and using pnm:// for the protocol ensures that users will receive the minimum bit rate available and no SureStream switching.

Use rtsp:// for the protocol, instead.

Spell clip names correctly, but change the capitalization.

RealServer is case-sensitive. Make sure you use the same capitalization in the links as in the encoding software or the on-demand file name. You might find it easiest to always use lowercase.

Serve poorly authored content.

Your RealServer may be tuned to run perfectly, but if the clips themselves weren't created using the best production techniques, they may require too much bandwidth.

Contacting RealNetworks Technical Support

If you have followed the troubleshooting tips in this chapter and have not been able to solve the problem, check the RealNetworks Knowledge Base for help. The Knowledge Base contains solutions to many problems not covered here:

- <http://service.real.com/kb/default.htm>

For technical support with RealSystem, please fill out the form at:

- <http://service.real.com/contact/email.htm>

The information you provide in this form will help technical support personnel to give you a prompt response. For general information about RealNetworks' technical support, visit:

- <http://service.real.com/help/call.html>

In addition to asking for a detailed description of the problem you are experiencing, support technicians will want to know the information shown in the following form.

Information Needed by the RealNetworks Technical Support Department

Information About Your Server	
Exact Server version (see “Determining the Server Version” to find the version number)	8._._._._
Information About Your System	
Operating system	
Processor type and speed	
Available RAM	
Port numbers	
Type of connection to the Internet	
Is there a Web server on this system?	
Information About Other Software	
Client software version	
Encoding software version	
Information About the Problem	
Exact text of error message (if any):	
What is the bit rate of the content being streamed?	
How are you delivering content—are you streaming on-demand clips or broadcasting live clips?	
To how many clients are you streaming simultaneously?	
If the problem is with a certain feature, when was the last time it worked correctly? What has changed?	
Are there any related problems?	
What features are you using?	
What troubleshooting steps have you already tried?	

Determining the Server Version

There are two methods for finding the exact version of the server you are running.

- ▶ To determine the version of the Server (at a command prompt):

At a command prompt, navigate to the Bin directory, and type the following:

```
rmserver -v
```

The version number appears, in the form 8.x.x.xxx, where x varies according to your operating system.

- ▶ To determine the version of the Server (through RealSystem Administrator):

In RealSystem Administrator, click About.

A new browser window appears, with information about your Server.

The version number can vary according to the operating system you use. If you are contacting the RealNetworks Technical Support department for assistance, it is important that they know the exact version you have.

SUMMARY OF LINK FORMATS



The tables in this appendix summarize link formats for each type of delivery method and content type.

For more information on creating links in general, consult Chapter 5, “Understanding Link Formats”.

Link formats shown are based on default values included with RealServer.

Note

If you change or add a mount point, or change the port numbers, remember to use the new values in your links. And be sure to tell anyone else who creates links, too.

The Subject of a Link

When creating a link in a Web page, remember that you do not actually link to the clip itself; you link to a metafile that references it—either a Ram file, a SMIL file, or the automatically generated Ram file. For each type of content, two types of links are shown:

- a link in a Web page, which starts with `http://`
- a link that can be typed directly in RealPlayer, or used in a Ram or SMIL file, or created by Ramgen. This type starts with `rtsp://`

For this second type of link, the format is nearly the same as the link used in the Web page, with three exceptions: the protocol is different, the port number (if any) matches the protocol, and Ramgen is omitted.

The format you use for the link depends on two factors:

- where you will put the link (in a Web page, a Ram or SMIL file, or by typing it into RealPlayer)
- whether you are linking to clips or to metafiles

The table below summarizes the differences:

Location of and Type of Link			
Location of Link	File Type	Formats	Protocol
Web page	media clip	Ramgen mount point	http
	metafile	no special format	http
Ram or SMIL file, RealPlayer	media clip	no special format	rtsp or pna

Links to scalable multicasts are different in two ways:

- Ramgen mount point is not used
- http protocol is always used, regardless of whether the link appears in a Web page or in RealPlayer

How Authenticated Content Is Different

The tables in this appendix show how to construct a link to that type of authenticated content.

Two factors distinguish authenticated content from regular content:

- On-demand content must be stored in a different location than regular content, and it must not be a subdirectory of the main base path.
- The content creator must include `/secure/` as part of its path in the encoding software.
- The link must include the mount point `/secure/`.

Note

Merely adding the secure mount point to a link does not mean the material will be authenticated. You must set up the authentication feature, and create the correct links, for authentication to work.

Multiple Mount Points in Links

When you are combining several features at once, the following guidelines will help you to decide the order in which the relevant mount points should appear in your link:

- When ramgen is used in a link, it is the first mount point.

- When encoder is used in a link, it is always the last mount point. Everything to the right of the encoder mount point is considered path information.
- The mount point secure should appear just before the file name (for on-demand clips) or just before the encoder mount point (for live clips). If you place it ahead of the ramgen mount point, the Web browser will perform the authentication, rather than RealServer.

Port Numbers in Links

If you change the port numbers for **RTSP Port**, **PNA Port** and **HTTP Port** from their default values, you will need to tell your users so that they can include the new ports in their links. (If a link does not include a port number, RealPlayer uses default values for contacting the RealServer. But if RealServer is no longer listening on those ports, it will not receive the request.) See “Port Numbers” on page 95 for more information.

On-Demand Content

Mount points	ramgen
Link in Web page	<i>http://address:HTTPPort/ramgen/path/file</i>
Example	<i>http://realserver.example.com:8080/ramgen/presentation/presentation.rm</i>
Link within Player, Ram files, SMIL files	<i>rtsp://address:RTSPPort/path/file</i>
Example	<i>rtsp://realserver.example.com:554/presentation/presentation.rm</i>
Reference	“Linking to On-Demand Clips” on page 141.
Authenticated Content	
Mount points	ramgen, secure
Link in Web page	<i>http://address:HTTPPort/ramgen/secure/path/file</i>
Example	<i>http://realserver.example.com:8080/ramgen/secure/concerts/summer/mozart.rm</i>
Link within Player, Ram files, SMIL files	<i>rtsp://address:RTSPPort/secure/path/file</i>
Example	<i>http://realserver.example.com:8080/ramgen/secure/concerts/summer/mozart.rm</i>
Reference	“Step 4: Linking to Authenticated Content” on page 231.

Account-Based ISP Hosted Content

Mount points	ramgen
Link in Web page	<code>http://address:HTTPPort/ramgen/~account/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/~schu/music.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/~account/path/file</code>
Example	<code>rtsp://realserver.example.com:554/ramgen/~schu/music.rm</code>
Reference	“Step 3: Linking to ISP Content” on page 269.

Authenticated Content

Mount points	Authentication is not available for users' content. See “ISP Hosting Used with Other Features” on page 257.
Link in Web page	
Link within Player, Ram files, SMIL files	
Reference	

Dedicated ISP-Hosted Content

Mount points	ramgen
Link in Web page	<i>http://address:HTTPPort/ramgen/directory1/directory2/path/file</i>
Example	<i>http://realserver.example.com:8080/ramgen/s/sc/schu/music.rm</i>
Link within Player, Ram files, SMIL files	<i>rtsp://address:RTSPPort/directory1/directory2/path/file</i>
Example	<i>rtsp://realserver.example.com:554/s/sc/schu/music.rm</i>
Reference	“Dedicated Hosting Server” on page 270

Authenticated Content

Mount points	Authentication is not available for users' content. See “ISP Hosting Used with Other Features” on page 257.
Link in Web page	
Link within Player, Ram files, SMIL files	

Ad Streaming with Automatic SMIL Generation

Mount points	adtag/general, smilgen/banner, smilgen/leadin, smilgen/rbanner
Link in Web page	<code>http://address:HTTPPort:8080/ramgen/adtag/general/smilgen/banner/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/adtag/general/smilgen/banner/g2video.rm</code>
Link within Player, Ram files	<code>rtsp://address:RTSPPort/adtag/general/smilgen/banner/path/file</code>
Example	<code>rtsp://realserver.example.com:554/adtag/general/smilgen/banner/g2video.rm</code>
Reference	“Understanding SMIL Generation Mount Points” on page 327
Authenticated Content	
Mount points	ramgen, adtag/general, smilgen/banner, smilgen/leadin, smilgen/rbanner
Link in Web page	<code>http://address:HTTPPort/ramgen/adtag/general/smilgen/secure/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/adtag/general/smilgen/secure/g2video.rm</code>
Link within Player, Ram files	<code>rtsp://address:RTSPPort/adtag/general/smilgen/secure/path/file</code>
Example	<code>rtsp://realserver.example.com:554/adtag/general/smilgen/secure/path/file</code>
Reference	“Using Authentication with Ad Streaming” on page 317

Ad Streaming with SMIL Files

Mount points	adtag/general
Link in Web page	<i>http://address:HTTPPort:ramgen/ramgen/adtag/general/path/file</i>
Example	<i>http://address:HTTPPort:ramgen/ramgen/adtag/general/g2video.smi</i>
Link within Player, Ram files	<i>rtsp://address:RTSPPort/adtag/general/path/file</i>
Example	<i>rtsp://realserver.example.com:554/adtag/general/g2video.smi</i>
Reference	“Requesting SMIL Files from an Ad Server” on page 314

Unicast Live Content (from Encoders)

Mount points	ramgen, encoder
Link in Web page	<code>http://address:HTTPPort/ramgen/encoder/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/encoder/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/encoder/path/file</code>
Example	<code>rtsp://realserver.example.com:554/encoder/live.rm</code>
Reference	“Creating the Link to the Live Unicast” on page 148
Authenticated Content	
Mount points	ramgen, secure, encoder
Link in Web page	<code>http://address:HTTPPort/ramgen/encoder/secure/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/encoder/secure/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/encoder/secure/path/file</code>
Example	<code>rtsp://realserver.example.com:554/encoder/secure/live.rm</code>
Reference	“Step 4: Linking to Authenticated Content” on page 231

Unicast Live Content (from pre-G2 Encoders)

Mount points	ramgen, live
Link in Web page	<code>http://address:HTTPPort/ramgen/live/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/live/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/live/path/file</code>
Example	<code>rtsp://realserver.example.com:554/live/live.rm</code>
Reference	“Creating the Link to the Live Unicast” on page 148
Authenticated Content	
Mount points	ramgen, secure, live
Link in Web page	<code>http://address:HTTPPort/ramgen/live/secure/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/live/secure/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/live/secure/path/file</code>
Example	<code>rtsp://realserver.example.com:554/live/secure/live.rm</code>
Reference	“Step 4: Linking to Authenticated Content” on page 231

Live content from encoders designed before RealSystem 6 (such as RealEncoder and RealProducer 5) uses the `/live/` mount point.

Archived Live Content

Mount points	ramgen
Link in Web page	<code>http://address:HTTPPort/ramgen/Archive/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/Archive/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/Archive/file</code>
Example	<code>rtsp://realserver.example.com:554/Archive/live.rm</code>
Reference	“Linking to Archived Files” on page 153
Authenticated Content	
Mount points	ramgen, secure
Link in Web page	<code>http://address:HTTPPort/ramgen/secure/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/secure/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/secure/path/file</code>
Example	<code>rtsp://realserver.example.com:554/secure/live.rm</code>
Reference	“Step 4: Linking to Authenticated Content” on page 231.

If the live content is being created by a pre-G2 encoder, substitute `/live/` for `/encoder/`.

Note

Secure archived content must be stored in a different location than archived material available to everyone.

Broadcasts that Use G2SLTA as the Source

Mount points	ramgen, encoder
Link in Web page	<code>http://address:HTTPPort/ramgen/encoder/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/encoder/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/encoder/path/file</code>
Example	<code>rtsp://realserver.example.com:554/encoder/live.rm</code>
Reference	“Step 4: Linking to the Simulated Live Broadcast” on page 57
Authenticated Content	
Mount points	ramgen, encoder, secure
Link in Web page	<code>http://address:HTTPPort/ramgen/encoder/secure/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/encoder/secure/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/encoder/secure/path/file</code>
Example	<code>rtsp://realserver.example.com:554/encoder/secure/live.rm</code>
Reference	“Step 4: Linking to Authenticated Content” on page 231

If the live content is being created by a pre-G2 encoder, substitute `/live/` for `/encoder/`.

Redundant Live Content

Mount points	ramgen, redundant
Link in Web page	<i>http://address:HTTPPort/ramgen/redundant/path/file</i>
Example	<i>http://realserver.example.com:8080/ramgen/redundant/live.rm</i>
Link within Player, Ram files, SMIL files	<i>rtsp://address:RTSPPort/redundant/path/file</i>
Example	<i>rtsp://realserver.example.com:554/redundant/live.rm</i>
Reference	“Linking to Redundant Content” on page 65
Authenticated Content	
Mount points	ramgen, secure, redundant
Link in Web page	<i>http://address:HTTPPort/ramgen/redundant/secure/path/file</i>
Example	<i>http://realserver.example.com:8080/ramgen/redundant/secure/live.rm</i>
Link within Player, Ram files, SMIL files	<i>rtsp://address:RTSPPort/redundant/secure/path/file</i>
Example	<i>rtsp://realserver.example.com:554/redundant/secure/live.rm</i>
Reference	“Step 4: Linking to Authenticated Content” on page 231

Split Content (Push Split)

Mount points	ramgen, broadcast, encoder
Link in Web page	<code>http://Receiver:HTTPPort/ramgen/broadcast/TransmitterSourceName/path/file</code>
Example	<code>http://realserver.example.com.au:8080/ramgen/broadcast/Japan/encoder/concert.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://Receiver:RTSPPort/broadcast/TransmitterHostName/path/file</code>
Example	<code>rtsp://realserver.example.com.au:554/broadcast/Japan/encoder/concert.rm</code>
Reference	“Step 3: Linking to Split Content” on page 169.

Authenticated Content

Mount points	Split material can be authenticated, but it requires elaborate measures. See “Authentication and Splitting” on page 162.
Link in Web page	
Link within Player, Ram files, SMIL files	

Split Backup Transmitter Content (Push Split)

Mount points	ramgen, broadcast
Link in Web page	<code>http://Receiver:HTTPEndPoint/ramgen/broadcast/*/path/file</code>
Example	<code>http://realserver.example.com.au:8080/ramgen/broadcast/*/concert.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://Receiver:RTSPPEndPoint/broadcast/*/path/file</code>
Example	<code>rtsp://realserver.example.com.au:554/broadcast/*/concert.rm</code>
Reference	“Linking to Backup Transmitters” on page 173.

Pull Split Content

Mount points	ramgen, split, encoder
Link in Web page	<code>http://Receiver:HTTPPort/ramgen/broadcast/SourcePath/Transmitter:PullListenPort/encoder/path/file</code>
Example	<code>http://realserver.example.com.au:8080/ramgen/broadcast/realserver.example.com.jp:2030/encoder/concert.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://Receiver:RTSPPort/broadcast/SourcePath/Transmitter:PullListenPort/encoder/path/file</code>
Example	<code>rtsp://realserver.example.com.au:554/ramgen/broadcast/pull/realserver.example.com.jp:2030/encoder/concert.rm</code>
Reference	“Linking to Pull Split Content” on page 178

Authenticated Content

Mount points	Split material can be authenticated, but it requires elaborate measures. See “Authentication and Splitting” on page 162.
Link in Web page	
Link within Player, Ram files, SMIL files	

Back-Channel Multicast Content

Mount points	ramgen, encoder
Link in Web page	<code>http://address:HTTPPort/ramgen/encoder/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/encoder/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/encoder/path/file</code>
Example	<code>rtsp://realserver.example.com:554/encoder/live.rm</code>
Reference	“Linking to Back-Channel Multicasts” on page 198
Authenticated Content	
Mount points	ramgen, encoder, secure
Link in Web page	<code>http://address:HTTPPort/ramgen/encoder/secure/path/file</code>
Example	<code>http://realserver.example.com:8080/ramgen/encoder/secure/live.rm</code>
Link within Player, Ram files, SMIL files	<code>rtsp://address:RTSPPort/encoder/secure/path/file</code>
Example	<code>rtsp://realserver.example.com:554/encoder/secure/live.rm</code>
Reference	“Step 4: Linking to Authenticated Content” on page 231

If the live content is being created by a pre-G2 encoder, substitute `/live/` for `/encoder/`.

Scalable Multicast Content

Mount point	scalable
Link in Web page	<code>http://address:HTTPPort/scalable/path/file.rm.sdp</code>
Example	<code>http://realserver.example.com:8080/scalable/concert.rm.sdp</code>
Link within Player, Ram files, SMIL files	<code>http://address:HTTPPort/scalable/path/file.rm.sdp</code> (same as link in Web page)
Example	<code>http://realserver.example.com:8080/scalable/concert.rm.sdp</code> (same as link in Web page)
Reference	“Linking to Scalable Multicasts” on page 203
Authenticated Content	
Mount points	scalable, secure
Link in Web page	<code>http://address:HTTPPort/scalable/secure/path/file.rm.sdp</code>
Example	<code>http://realserver.example.com:8080/scalable/secure/concert.rm.sdp</code>
Link within Player, Ram files, SMIL files	<code>http://address:HTTPPort/scalable/secure/path/file.rm.sdp</code> (same as link in Web page)
Example	<code>http://realserver.example.com:8080/scalable/secure/concert.rm.sdp</code> (same as link in Web page)
Reference	“Step 4: Linking to Authenticated Content” on page 231

Scalable multicast links use a slightly different format than other links:

- Links point to a .sdp file, rather than to the actual live file. The Server automatically generates the .sdp file when a user clicks the link.
- Links do not include the encoder mount point because information about the nature of the broadcast is included in the .sdp file.
- Ramgen is not used in these links.

If the live content is being created by a pre-G2 encoder, substitute `/live/` for `/encoder/`.

Ram Files

Mount points	none, for most links
Link in Web page	<code>http://address:port/path/file.ram</code>
Example	<code>http://webserver.example.com:80/music.ram</code>
Reference	“Ram Files and Ramgen” on page 74

Authenticated Ram File

Mount points	Ram files are delivered by Web servers, and thus cannot be authenticated by RealServer. The files referenced by the Ram file, however, can be authenticated.
Link in Web page	
Example	

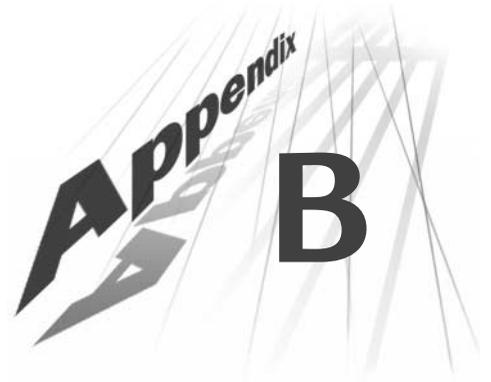
SMIL Files

Mount points	none, for most links
Link in Web page	<code>http://address:HTTPPort/ramgen/path/file.smi</code>
Example	<code>http://realserver.example.com:8080/ramgen/music.smi</code>
Reference	“SMIL Files” on page 76

Authenticated SMIL File

Mount points	There are two methods of authenticating SMIL files or the files they reference. Refer to “Working with SMIL Files” on page 238.
Link in Web page	
Example	

CONFIGURATION FILE SYNTAX



This appendix describes the structure of the configuration file.

Configuration File Components

The configuration file is constructed entirely of tags. There are four types of tags in this file: the XML declaration tag, optional comment tags, list tags, and variable tags.

Of these four types, only two make up the instructions to RealServer: lists and variables. Lists are used for instructions that have several parts, such as the MIME types or the multicast instructions. A list tag is followed by one or more list tags or variable tags.

All values for lists and variables are enclosed in double quotation marks.

XML Declaration Tag

The XML declaration tag indicates which version of XML is in use. RealServer 8 uses XML version 1.0. The declaration tag looks like this:

```
<?XML Version="1.0" ?>
```

Comment Tags

Comment tags are used in the configuration file to identify the functions of tags, but the comments aren't required. XML comment tags are just like those in HTML: they begin with `<!--` and end with `-->`. RealServer ignores these tags; they are for your benefit.

For example, this comment tag lets the administrator know that the parameters after it refer to the path settings:

```
<!-- P A T H S -->
```

Tip

To disable a feature, convert the feature's tag or tags to a comment. Rather than converting each tag to a comment, edit only the feature's first opening tag and last closing tag.

Do not nest comment tags within other comment tags.

List Tags

The list tag uses the following syntax:

```
<List Name="name">
```

```
...
```

```
</List>
```

where *name* is the list title. Using the correct capitalization for *name* is important.

Other lists or variables follow the list. The `</List>` tag signifies the end of the list. If other lists are inside the original list, they must also have closing `</List>` tags. The `MIMETypes` list is an example of a list that contains other lists.

Tip

Indenting list items is not required, but is recommended for clarity.

Variable Tags

Variable tags use the following syntax:

```
<Var name="value"/>
```

where *name* is the variable title, and *value* is a string or a number, depending on the variable. Capitalization for both *name* and *value* is important.

Unlike lists, variables do not have a closing tag; instead, a forward slash mark (/) appears before the closing angle bracket (>).

Tip

If you've restarted RealServer and it's not responding to a change you've made to a variable, make sure the variable has a closing forward slash mark, and that there is no space between them.

Variables can be independent elements (such as `LogPath`) or they may appear inside a list. When variables appear within a list, their meaning is determined by the value of the list name, although they may be apparently identical in syntax to variables that are not inside lists. If there are multiple variables within a list that do similar things, their names must be unique. For example, the `Extension` variables within each `MIMETypes` list must have different names; this is accomplished by adding a number to the end of each (`Extension_01`, `Extension_02`, and so on).

CONFIGURATION FILE CONTENTS



This appendix gives brief information about the contents of the configuration file for those administrators interested in editing it directly.

Editing the Configuration File

For those RealServer administrators who prefer to modify features by editing the configuration file directly, this appendix shows sample configuration file contents with brief descriptions. Detailed descriptions can be found in the chapters that describe each subject.

The default name of the configuration file is `rmserver.cfg`, but if you have multiple Servers you may want to rename the files so as to easily identify which server you're working with.

- ▶ To modify the configuration file with a text editor:
 1. Please read Appendix B, "Configuration File Syntax", which explains the structure of this file.
 2. Save a backup copy of the configuration file.
 3. Open the configuration file, and make any changes you like.

Be sure to use correct syntax, because RealServer looks for exact spellings and correct use of angle brackets. RealServer does not display messages related to syntax errors; instead, it will ignore those settings it does not understand. It may use minimal settings. See "Minimum Licensed Features" on page 39.
 4. Restart RealServer after changing any settings with a text editor.

RealSystem Administrator and the Configuration File

Information elsewhere in this manual on customizing RealServer features is based on the settings that appear in RealSystem Administrator. However, RealSystem Administrator mostly displays only those settings that will be changed in everyday use. Other items, such as the file system short name of the basic mount points, are not accessible through RealSystem Administrator. By viewing the configuration file and reading this section you will see the complete listing of settings for each feature.

Tip

A fast way to understand the structure of the configuration file is to first use RealSystem Administrator to make changes, and then examine the configuration file to see the effects. Noticing how lists are created and changed will be especially helpful. Note that you must exit RealSystem Administrator before opening the configuration file with a text editor or unexpected changes may result.

Some Observations About Variables

Most configuration file variable names closely match names in RealSystem Administrator. When there is a difference between the way it is configured in RealSystem Administrator and the actual variable name, the difference is noted here. RealSystem Administrator frequently adds spaces to the variable names (BasePath becomes Base Path, for instance), and those changes are not noted in this appendix.

Some variables, which are not part of lists, can appear anywhere in the configuration file, but are grouped here for clarity.

Variables that use true or false values (such as PlusOnly, the variable that determines whether RealPlayer Plus, rather than the free versions of RealPlayer, can play streams from your RealServer) may be represented in the configuration file with a 1 or the word True. In RealSystem Administrator, a choice of On, Yes, or Enabled usually corresponds to 1 or True in the configuration file, while Off, No, or Disabled usually corresponds to 0 or False. Within the configuration file, you may use either 1 or True to represent the positive condition, and 0 or False for the negative.

For example, `<Var PlusOnly="True"/>` and `<Var PlusOnly="1"/>` are equivalent statements.

RealServer Features in the Configuration File

Ad Streaming

Configuration elements of the ad streaming feature are not listed here, as the feature is designed to be configured only with RealSystem Administrator. Ad streaming elements appear within the FSMount section of the configuration file. Ad streaming is described in Chapter 20, “Streaming Targeted Ads”.

Access Control

Restricting access to RealServer content via the requesting client’s IP address is described in Chapter 14, “Limiting Access to RealServer”. For every address or address range to which you want to restrict access, create a list with a unique number. The number can be any length, but a number of more than one digit is recommended in case more lists are added later; with multiple digits, the new lists can be inserted between existing lists.

Each list is called a rule. Rules are processed in numerical order. RealServer searches the list of rules to find the first rule that matches the address. Because RealServer searches the list of rules in numerical order, make your broadest categories first.

Within each list, the following settings are used: Access, Transport, To, From, and a list named Ports.

Access Control Configuration Elements

Element	Description
<List Name="AccessControl">	
<List Name="100">	
<Var Access="Allow"/>	Whether access is allowed or denied: set to Allow or Deny.
<Var Transport="TCP"/>	Transmission method being accessed. TCP is the only option for this list.
<Var To="127.0.0.1"/>	Address of the host RealServer or network card of hosting machine. Use specific address or Any. (This is shown as Server IP Address in RealSystem Administrator.)

(Table Page 1 of 2)

Access Control Configuration Elements (continued)

Element	Description
<Var From="any"/>	Address of the client computer whose access you are limiting. Use specific address or Any. To specify a range of IP addresses, either place a colon after the IP address and give the full subnet mask, or place a slash mark after the IP address and give the number of bytes for the subnet mask. For example, the following are equivalent values to use in the From variable: 172.16.3.0:255.255.255.0 and 172.16.3.0/24. Both examples specify the range of addresses from 172.16.3.0 to 172.16.3.254. (This is shown as Client IP Address in RealSystem Administrator.)
<List Name="Ports">	List of ports to which access is restricted.
<Var Port_01="554"/>	The port number should match the port numbers which RealServer is using for other features, such as RTSPPort, HTTPPort, and the port value used by the encoder list.
<Var Port_02="4040"/>	
<Var Port_03="5050"/>	
<Var Port_04="7070"/>	
<Var Port_05="8080"/>	
<Var Port_06="9090"/>	
</List>	
</List>	
</List>	

(Table Page 2 of 2)

Allowance

Settings in this section refer to the allowance plug-in. They are described in Chapter 14, "Limiting Access to RealServer".

If you establish values for both ClientConnections and MaxBandwidth, RealServer will limit access when the lower threshold is reached.

When set to 0n, ValidPlayerOnly sends a message to any clients other than RealNetworks RealPlayer 5 or RealNetworks RealPlayer 6 directing them to upgrade to the latest version of RealPlayer. If set to 0ff, all clients can receive

all clips. In the free downloadable versions of RealServer, this is set to 0n and cannot be changed.

Allowance Configuration Elements

Element	Description
<code><Var ClientConnections="25"/></code>	Limits the number of connections that can be in use simultaneously. Must be less than or equal to the number of streams in your license. Range is 1 to 32767. If omitted or set to 0, RealServer uses the number in your license.
<code><Var MaxBandwidth="64"/></code>	Limits the amount of bandwidth in use by RealServer. The value is given in kilobits per second. The default value is 0, which means to use the maximum number of connections, as allowed by the license.
<code><Var ValidPlayersOnly="True"/></code>	Allows only RealPlayer versions 5 and 6 to access content. Any other clients attempting to view or listen to content display a message directing them to upgrade to the latest version of RealPlayer. If ValidPlayerOnly is set to Off, all clients can receive all clips. In Basic Server and Basic Server Plus, this is set to 0n and cannot be changed.
<code><Var MinPlayerProtocol="0"/></code>	Limits access by protocol number. Use one of the following values for MinPlayerProtocol: 0 All clients are permitted to connect 4 RealAudio Player 1.0 and later can connect 7 RealAudio Player 2.0 and later can connect 8 RealAudio Player 3.0 and later can connect 10 RealPlayer 4.0 and later can connect
<code><Var PlusOnly="False"/></code>	When set to True, PlusOnly enables only RealPlayer Plus to play content.

Authentication and Commerce

Authentication is described in Chapter 15, "Authenticating RealServer Users". There are four areas that apply to authentication in the configuration file, and each of these four areas is in a separate list. The four main areas refer to each other, but are kept separate for flexibility. Two separate secure paths might use the same realm (and therefore the same database) to perform the same type of

authentication for content kept in different locations. This enables different types of content to share the same list of authorized users.

Authentication Realms

A realm is a way of associating a group of users and the protocol used to verify their credentials.

Each sublist of AuthenticationRealms gives properties for a different realm. Every realm has a name (identified by the Realm variable), and a list that identifies what type of authentication is used in that realm. Depending on which authentication type you choose, different variables are required within the sublist (see the “AuthenticationRealms PluginID Settings” table). When RealServer is installed on a Windows NT system, you can take advantage of NT authentication and direct RealServer to use the list of authorized users.

Authentication Realms Configuration Elements

Element	Description
<List Name="AuthenticationRealms">	
<List Name="SecureAdmin">	A realm.
<Var Realm="AdminRealm"/>	Name of this realm. Lists in the CommerceRules and FSMount lists may refer to this.
<List Name="NTLMAuthenticator">	User-defined description of authentication to use in this realm. Use only one type of authentication per realm.
<Var PluginID="rn-auth-sspi"/>	Plug-in which performs the authentication. For a list of options, see the “AuthenticationRealms PluginID Settings” table below.
<Var Provider="NTLM"/>	
<Var Group="Administrators"/>	Name of an NT administrator-defined user group, whose members are allowed access. In this example, only members of the “Administrators” group are permitted to view content controlled by this realm.
</List>	
</List>	
<List Name="SecureEncoder">	A realm.
<Var Realm="EncoderRealm"/>	See description earlier in this section.

(Table Page 1 of 2)

Authentication Realms Configuration Elements (continued)

Element	Description
<List Name="RN5Authenticator">	User-defined description of authentication to use in this realm. Use only one type of authentication per realm.
<Var PluginID="rn-auth-rn5"/>	Plug-in which performs the authentication. For a list of options, see the "AuthenticationRealms PluginID Settings" table below.
<Var DatabaseID="Encoder_RN5"/>	Identifies which database to look in for authentication data. Refers to a list name within the Databases list.
</List>	
</List>	
<List Name="SecureContent">	A realm.
<Var Realm="ContentRealm"/>	See description earlier in this section.
<List Name="NTLMAuthenticator">	User-defined description of authentication to use in this realm. Use only one type of authentication per realm.
<Var PluginID="rn-auth-sspi"/>	Plug-in which performs the authentication. For a list of options, see the "AuthenticationRealms PluginID Settings" table below.
<Var Provider="NTLM"/>	See description earlier in this section.
</List>	
</List>	
</List>	

(Table Page 2 of 2)

AuthenticationRealms PluginID Settings

PluginID Value	Authentication Protocol	Associated Variables
rn-auth-basic	Basic	DatabaseID (required)
rn-auth-rn5	RN5	DatabaseID (required)
rn-auth-sspi	Windows NTLM Challenge/Response	Provider (required), Group (optional)

Commerce Rules List

The commerce rules list associates part of an URL with authentication. When RealServer looks through the URL to decide which plug-in should process the request, it compares each section of the URL with the ProtectedVirtualPath. Should this match, RealServer looks at the other information within the list to determine which realm protects the content, and which database lists the permissions (if any).

Each sublist within SecureContent associates the mount point with the information. The mount point for RealSystem Administrator does not need to go here.

Variables used with sublists are ProtectedVirtualPath, Realm, UseGUIDValidation, EvaluatePermissions, AllowDuplicateIDs, and DatabaseID. Use Realm or UseGUIDValidation, but not both.

Commerce Rules Configuration Elements

Element	Description
<List Name="CommerceRules">	
<List Name="SecureContent">	
<Var ProtectedVirtualPath="/secure"/>	Name of the mount path, virtual directory, or actual directory, used in URLs, that you want RealServer to authenticate.
<Var Realm="ContentRealm"/>	This points to the realm names in AuthenticationRealms list. Sets up user authentication. Don't use if UseGUIDValidation is also in use.
<Var EvaluatePermissions="True"/>	Instructs RealServer whether or not to look at the permissions list, or to just allow access to all content.
<Var DatabaseID="Content_RN5"/>	Points to the database identifiers in the Databases list.
<Var AllowDuplicateIDs="False"/>	Determines whether someone who's already logged on can successfully log on at another location. When set to False, a user gets the error message "Your account is locked" if they attempt to log on using the same account or player ID.
</List>	

(Table Page 1 of 2)

Commerce Rules Configuration Elements (continued)

Element	Description
<List Name="SecureLiveContent">	
<Var ProtectedVirtualPath="/encoder/secure"/>	Name of the mount path, virtual directory, or actual directory, used in URLs, that you want RealServer to authenticate.
<Var UseGUIDValidation="True"/>	Sets up player authentication. Gathers the client's ID, but not the user's name. Don't use if Realm is in use.
<Var EvaluatePermissions="True"/>	See description earlier in this section.
<Var DatabaseID="Content_RN5"/>	Points to the database identifiers in the Databases list.
<Var AllowDuplicateIDs="True"/>	See description earlier in this section.
</List>	
</List>	

(Table Page 2 of 2)

Player Authentication

In player authentication, the client sends a special string to RealServer indicating that the client is registering. The GUIDRegistrationPrefixes list identifies the special string (the GUIDRegistrationPrefix variable) and the database in which to store the player identification. You must embed this string in the link on the Web page.

Two variables are required for each sublist: GUIDRegistrationPrefix and DatabaseID.

Player Authentication Configuration Elements

Element	Description
<List Name="GUIDRegistrationPrefixes">	
<List Name="FirstDatabase">	
<Var GUIDRegistrationPrefix="register1"/>	String required from client in registering. Can be any single word, with any combination of letters and integers. Must be unique in the GUIDRegistrationPrefixes list.

(Table Page 1 of 2)

Player Authentication Configuration Elements (continued)

Element	Description
<Var DatabaseID="Content_RN5"/> </List>	Name of database, from Databases list, that will store this type of data.
<List Name="SecondDatabase">	
<Var GUIDRegistrationPrefix="register2"/>	String required from client in registering. Can be any single word, with any combination of letters and integers. Must be unique in the GUIDRegistrationPrefixes list.
<Var DatabaseID="Content_ODBC"/> </List>	Name of database, from Databases list, that will store this type of data.
</List>	

(Table Page 2 of 2)

Databases List

The databases list is the master list of available databases for each type of authentication. Databases store usernames and passwords of authorized users.

Within the list, sublists associate database plug-ins with location information.

In the examples shown here, PluginID is always set to rn-db-flatfile. There is only one variable associated with rn-db-flatfile, but other values for PluginID require different variables. Refer to the “Databases PluginID Settings” table on page 393.

Databases Configuration Elements

Element	Description
<List Name="Databases">	
<List Name="Admin_Basic">	Database information for RealSystem Administrator user authentication.
<Var PluginID="rn-db-flatfile"/>	Name of plug-in that will interact with the database. See “Databases PluginID Settings” table for a list of options.
<Var Path="C:\Program Files\Real\RealServer\adm_b_db"/>	Location where the database files are stored or will be stored.

(Table Page 1 of 2)

Databases Configuration Elements (continued)

Element	Description
</List>	
<List Name="Encoder_RN5">	Database information for encoder authentication.
<Var PluginID="rn-db-flatfile"/>	See description earlier in this section.
<Var Path="C:\Program Files\Real\RealServer\enc_r_db"/>	See description earlier in this section.
</List>	
<List Name="Content_RN5">	Database information for live and on-demand user authentication.
<Var PluginID="rn-db-flatfile"/>	See description earlier in this section.
<Var Path="C:\Program Files\Real\RealServer\con_r_db"/>	See description earlier in this section.
</List>	
<List Name="PlayerContent">	Database information for player authentication.
<Var PluginID="rn-db-flatfile"/>	See description earlier in this section.
<Var Path="c:\Program Files\Real\RealServer\con_p_db"/>	See description earlier in this section.
</List>	
</List>	

(Table Page 2 of 2)

The table below shows the variables required for each data storage method.

Databases PluginID Settings

Data Store Method	PluginID Value	Associated Variables
Text file.	rn-db-flatfile	Path (required)
MSQL database.	rn-db-mysql	Name (required) (called Database Name in RealSystem Administrator) Password (optional) TableNamePrefix (optional) Hostname (optional) User (optional) (called User Name in RealSystem Administrator)

(Table Page 1 of 2)

Databases PluginID Settings (continued)

Data Store Method	PluginID Value	Associated Variables
ODBC-compliant databases.	rn-db-odbc	Name (required) (called Database Name in RealSystem Administrator) Password (optional) TableNamePrefix (optional) User (optional) (called User Name in RealSystem Administrator)
Data store plug-ins created by third-party companies for use with RealServer version 5.	rn-db-wrapper	PathToDBPlugin (required) DBName (required) (Called Database Name in RealSystem Administrator) DBLoginUsername (optional) DBLoginPassword (optional)

(Table Page 2 of 2)

Each data store method requires different variables. The table below shows the significance of each variable.

Data Store Variables

Variable	Purpose
<Var DBLoginName="name"/>	Name required by database application.
<Var DBLoginPassword="password"/>	Password required by database application.
<Var DBName="path"/>	If you are using a text file storage method, <i>path</i> is the location of the main text file storage directory. If you are using any other method, <i>path</i> is the name or location of the plug-in. Consult your plug-in documentation.
<Var HostName="address"/>	IP address or DNS name of computer where database is stored.
<Var Name="name"/>	Name required by database application.
<Var Password="password"/>	Password required by database application.
<Var Path="path"/>	Path to text file storage main directory.
<Var PathToDBPlugin="path"/>	Location of plug-in
<Var TableNamePrefix="prefix"/>	Prefix used to make field names unique, when used with an existing database.
<Var User="name"/>	Name required by database application.

Secure Content

Within the FSMount list, one section refers to authentication. It uses three variables: ShortName, MountPoint, and BasePath.

Secure Configuration Elements

Element	Description
<List Name="RealSystem Secure Content">	This file system delivers secure content.
<Var ShortName="pn-local"/>	Short name of local file system plug-in. See "ShortName Variable"
<Var MountPoint="/secure/">	All authenticated content uses this mount point.
<Var BasePath="C:\Program Files\Real\RealServer\Secure"/>	Location of content to be authenticated.
</List>	

Broadcast Redundancy

This feature is described in Chapter 4, "Sources of Content". The settings shown in the table below appear in their own section of the configuration file.

Encoder Redundancy Configuration Elements

Element	Description
<List Name="RealSystem Broadcast Redundancy Plugin">	Name of this list.
<Var Enabled="1"/>	Turns on this feature (0 means turn off the feature).
<Var SeparatorCharacter="."/>	Delimiter to use in source to indicate the order that RealServer should use each source. Default is a period (.); options are options are a single quote ('), a tilde (~), or a caret (^).
<Var LiveSourceTimeout="2"/>	Number of seconds before RealServer switches to the next source if one source becomes unavailable.
<Var Reconnect="Manual"/>	Auto: RealServer will automatically switch all clients to the next available source. Manual: users see an error message and must click OK to be switched to the next source.

(Table Page 1 of 2)

Encoder Redundancy Configuration Elements (continued)

Element	Description
<code><Var ErrorMessage="Broadcast timed out; click the Play button to restart."></code>	Text of the error message that will appear if one source disappears.
<code></List></code>	

(Table Page 2 of 2)

There are also elements in the FSMount section of the configuration file related to this feature.

FSMount Encoder Redundancy Configuration Elements

Element	Description
<code><List Name="FSMount"></code>	
...	
<code><List Name="RealSystem Broadcast Redundancy"></code>	Name of this list within FSMount list.
<code><Var ShortName="pn-redundant"/></code>	Short name of this list.
<code><Var MountPoint="/redundant"/></code>	Mount point to use in links to redundant content.
<code></List></code>	
...	
<code></List></code>	

Caching

This section enables media caches to request and cache streams on behalf of clients. Caching is described in Chapter 8, "Advanced Features".

To selectively block media caches from requesting your content, add the media cache's IP address to the AccessControl list. In addition to specifying the IP address, indicate the port number to which access should be denied (usually 7802).

To block all media cache requests, set TSEnable to False.

To disable logging of cache requests, set the TSLog variable to 0.

Cache Configuration Elements

Element	Description
<Var TSEnable="True"/>	Permits media caches to request and then cache content streamed from RealServer (when set to True). (In RealSystem Administrator this is changed with Cache Requests .)
<Var TSPort="7802"/>	Port number to which media caches send their requests to RealServer. Do not change this unless you want to refuse requests from media caches. (In RealSystem Administrator this is changed with Cache Port .)
<Var TSLog="True"/>	Turns on the log of requests made by media caches. (This is not shown in RealSystem Administrator.)
<Var TSLogPath="C:\Program Files\Real\RealServer\Logs\cache.log"/>	Path and file name of cache request log. The default location is the Logs directory, and the default name is cache.log. (In RealSystem Administrator this is changed with Cache Log Path .)
<List Name="NoCacheDir">	List of directories whose content is not available to media caches. If RealServer receives a request for material included in the NoCacheDir list, it streams the file directly to the client rather than allowing it to be cached and re-transmitted. As always, RealServer records the transaction in the access log, and reports a download size of 0 bytes in the cached requests log file.
<Var Directory_01="/nocache1"/>	
<Var Directory_02="/nocache2"/>	
</List>	

Encoders

Both encoding lists appear within the FSMount section.

Encoders

Receiving streams from both RealSystem Encoders and earlier versions are explained in Chapter 11, "Unicasting Live Presentations". These variables are used in this list: ShortName, MountPoint, Port, and EncoderRealm.

Unlike other plug-ins, encoder lists cannot have multiple mount points.

RealSystem Encoders Configuration Elements

Element	Description
<List Name="RealSystem G2 Encoders">	
<Var ShortName="pn-encoder"/>	Short name of G2 live encoder plug-in.
<Var MountPoint="/encoder"/>	Portion of URL that indicates the type of request and therefore which file system will handle the request.
<Var Port="4040"/>	Port to which encoders will send their live streams. Default value is 4040.
<Var EncoderRealm="EncoderRealm"/>	List of authentication protocols and databases. See AuthenticationRealms list.
</List>	

Pre-RealSystem G2 Encoders

The list for encoders such as RealEncoder and RealPublisher versions 5 and earlier uses these variables: ShortName, MountPoint, Port, Realm, and Password.

Unlike other plugins, encoder lists cannot have multiple mount points.

Pre-RealSystem G2 Encoders Configuration Elements

Element	Description
<List Name="Pre-RealSystem G2 Encoders">	
<Var ShortName="pn-live3">	Short name of version 5 and older live encoder plug-in.
<Var MountPoint="/live"/>	Portion of URL that indicates the type of request and therefore which file system will handle the request.
<Var Port="5050"/>	Port to which older encoders will send their live streams. Default value is 5050.
<Var Password="letmein"/>	Password used by encoders to connect to RealServer.
</List>	

File Systems (FSMount)

The FSMount section gives the names of all the configurable file system plug-ins in use. The plug-ins themselves are stored in a directory indicated by the PluginDirectory variable.

All requests of the RealServer are processed by plug-ins. Plug-ins control which features are available. The modular plug-in design means that new features can be programmed and easily substituted for the existing plug-ins. New plug-ins may require different list arrangements and variables; check with the developer of the plug-in for this information.

Additional Information

RealSystem SDK Developer's Guide provides developers with the public interfaces used to extend and customize RealSystem to stream new data-types, create new clients, or to customize RealServer by building a new plug-in.

Several features are listed within the FSMount list, but they are shown in their own sections in this appendix. Those features include:

- Ad Streaming
- Encoders
- Pre-RealSystem G2 Encoders
- RealSystem Administrator
- Secure Content
- Splitting
- Ramgen
- View Source

ShortName Variable

Each list within FSMount gives a short name for the plug-in. The short name is also stored within the plug-in file itself, and RealServer uses this to identify the correct file to use. To add a plug-in to your RealServer, you must know the name to use in the FSMount section; this name is supplied by the developer of the plug-in. The short name is referenced with the ShortName variable in each file systems list.

Local File System

The local file system, which handles requests for nearly all streamed media content, is described in Chapter 3, “Overview of RealServer”. In RealSystem Administrator, this section is configured on the Mount Points page.

The local file system handles requests for static media clips. It uses the variables ShortName, MountPoint and BasePath.

If clips are stored on more than one disk drive, add multiple local file system lists, each with its own mount point. The list names need to be unique.

Local File System Configuration Elements

Element	Description
<code><List Name="RealSystem Content"></code>	Identifies this list as the main content list.
<code><Var ShortName="pn-local"/></code>	The short name indicates which file system handles requests directed to this mount point.
<code><Var MountPoint="/"/></code>	The mount point for your main content will be set to /, which means that no additional information need be specified in URLs for clips to be handled by the local file system.
<code><Var BasePath="C:\Program Files\Real \RealServer\Content"/></code>	BasePath defaults to the Content subdirectory of your RealServer directory, which refers to the Content directory created during installation. All directories that you refer to in URLs will be relative to this directory.
<code></List></code>	

HTTP Support

Two lists refer to sending and receiving information via HTTP: HTTPDeliverable and HTTPPostable.

HTTPDeliverable

This feature indicates the mount points, virtual directories, or directories whose content can be streamed via HTTP. It is explained in Chapter 14, “Limiting Access to RealServer”.

Each Path variable gives the name of a virtual directory whose content can be streamed via HTTP. Be sure that the following mount points are on this list:

- admin—refer to RealSystem Administrator, which is served via HTTP
- ramgen—clips streamed with Ramgen may be requested in HTTP format
- scalable—clients receive some data via HTTP

- viewsource—view source features use HTTP for browsing

HTTP Deliverable List Configuration Elements

Element	Description
<List Name="HTTPDeliverable">	
<Var Path_0="/admin"/>	Each Path variable gives the name of a mount point, path, or virtual path whose content can be streamed via HTTP.
<Var Path_01="/ramgen"/>	
<Var Path_02="/httpfs"/>	
<Var Path_03="/viewsource"/>	
</List>	

HTTPostable

Like the list described above, the HTTPostable list enables virtual directories to receive data from clients.

Each Path variable gives the name of a virtual directory whose content can be streamed via HTTP.

The only item on this list is scalable, and that only needs to appear if the multicast feature is set to send client statistics (SendClientStatistics="True").

There is no way of configuring this list directly in RealSystem Administrator; however, if you use RealSystem Administrator to choose **Send Client Statistics**, RealSystem Administrator will create the list automatically.

HTTP Postable Configuration Elements

Element	Description
<List Name="HTTPostable">	
<Var Path_0="/scalable"/>	Each Path variable gives the name of a mount point, directory or virtual directory to which clients can send data via HTTP.
</List>	

ISP Hosting

The ISPHosting list contains two other lists: TranslationMounts (which contains one or more lists) and UserLists. Variables are MountPoint, UserPath, and File. ISP Hosting settings are described in Chapter 17, “ISP Hosting”.

ISP Hosting scenarios frequently require special base paths, so you will need to create additional mount points in the FSMount list. Examples are shown in this section.

Four items control where RealServer looks for hosted media:

1. In the user list file, */path/* groups the users. In the configuration file, UserPath has the same value as */path/*, or with a portion of it.
2. Within the TranslationMounts list (of the ISPHosting list), UserPath is associated with the MountPoint variable.
3. The MountPoint variable of the TranslationMounts list matches a MountPoint variable in the FSMount section of the configuration file.
4. The MountPoint variable of the FSMount list is associated with a BasePath variable. The directory shown by BasePath is where user directories are located.

Through this path of associated elements, the value for */path/* in the user list file is ultimately associated with a base path.

ISP Hosting Configuration Elements

Element	Description
<code><List Name="ISPHosting"></code>	
<code><List Name="TranslationMounts"></code>	The translation mounts section.
<code><List Name="ISP Content (Washington users)"></code>	Description of this sublist.
<code><Var MountPoint="/wa_isp/"></code>	Mount point associated with UserPath.
<code><Var UserPath="/wa/"></code>	UserPath is the same value as <i>/path/</i> in the user list file.
<code></List></code>	
<code><List Name="ISP Content (Oregon users)"></code> <code><Var MountPoint="/or/"></code> <code><Var UserPath="/or/"></code> <code></List></code>	See description earlier in this section.

(Table Page 1 of 2)

ISP Hosting Configuration Elements (continued)

Element	Description
<pre> . <List Name="ISP Content (Idaho users)"> <Var MountPoint="/id_isp/" /> <Var UserPath="/id/" /> </List> </List> </pre>	See description earlier in this section.
<pre> <List Name="UserLists"> </pre>	
<pre> <Var File1="c:\accounts\commercial \local.txt" /> </pre>	Location of user list file. RealServer loads the user lists in the order they appear in the configuration file; if the same user name appears in more than one list, RealServer uses the settings in the last user list.
<pre> <Var File2="c:\accounts\commercial \remote." /> </pre>	
<pre> <Var File3="c:\accounts\personal \local.txt" /> </pre>	
<pre> </List> </pre>	
<pre> </List> </pre>	

(Table Page 2 of 2)

It is optional, but common, to create special mount points in the FSMount section.

ISP Hosting Optional FSMount Configuration Elements

Element	Description
<pre><List Name="FSMount"> ...other mount points... <List Name="ISP Mount Points--Washington"> <Var ShortName="pn-local"/> <Var MountPoint="/wa"/> <Var BasePath="C:\UserAccounts"/> </List></pre>	These sections map the /wa/, /or/, and /id/ mount point to the C:\UserAccounts directories.
<pre><List Name="ISP Mount Points--Oregon"> <Var ShortName="pn-local"/> <Var MountPoint="/or"/> <Var BasePath="C:\UserAccounts"/> </List></pre>	
<pre><List Name="ISP Mount Points--Idaho"> <Var ShortName="pn-local"/> <Var MountPoint="/id"/> <Var BasePath="C:\UserAccounts"/> </List> ...other mount points... </List></pre>	

How RealServer Looks for Users' Content (Account-Based Hosting)

RealServer uses a combination of the URL, user list file, and the configuration file to determine where to look for user files.

This section describes how RealServer processes all requests when ISP Hosting is in use:

1. When it receives a request for content, RealServer looks at the account name after the tilde (~).
2. It looks through the user list for a matching account information. If your user list contains individual account names, RealServer searches these to find a match. If it doesn't find an exact match, it uses the generic account information.
3. Once it finds a match (or the generic information), it looks at the /path/ value. The /path/ information matches the UserPath variable in the

configuration file. This is not a physical path; it is used only for the next step, and serves as a way to group user account names logically.

4. Using the `/path/` information, RealServer goes to the `ISPHosting` list of the configuration file, where it looks within the `TranslationMounts` list for a matching `UserPath`. Once it finds a match, RealServer records the `MountPoint` located within the same list. This translates the logical path to a file system.
5. Next, RealServer looks in the `FSMount` list for a file system whose mount point matches the `MountPoint` from the `ISPHosting` list. Once it finds the correct mount point, RealServer notes the associated `BasePath`.
6. The media will be in a directory relative to `BasePath`.

Typically, users' content is mapped to a special ISP hosting mount point and base path. User directories are located under the base path.

Example

In the following example, an ISP in the northwest United States has divided its users by geographical location.

Example—User List File

The user list file groups the users in the WA and OR groups, and instructs RealServer to look for all other users in the ID path.

UserList [Sample URLs:
{chris, /wa/canderson/, 0, 5},	rtsp://server.company.com/~chris/file.rm
{lee, /or/ladams/, 0, 5},	rtsp://server.company.com/~lee/file.rm
{pat, /wa/pbrown/, 0, 5},	rtsp://server.company.com/~pat/file.rm
{sandy, /or/schu/, 0, 5},	rtsp://server.company.com/~sandy/file.rm
{~*, /id/, 0, 5}	rtsp://server.company.com/~username/file.rm
]	

Example—ISPHosting Section

The `ISPHosting` section of the configuration file maps the `UserPaths` to mount points. In this example, each list within the `TranslationMounts` section maps part of each user path to its own mount point.

```
<List Name="ISPHosting">
  <List Name="TranslationMounts">
    <List Name="Washington Users">
      <Var MountPoint="/wa_isp/" />
      <Var UserPath="/wa/" />
```

```

</List>
<List Name="Oregon Users">
  <Var MountPoint="/or_isp"/>
  <Var UserPath="/or"/>
</List>
<List Name="Idaho Users">
  <Var MountPoint="/id_isp"/>
  <Var UserPath="/id"/>
</List>
</List>
<List Name="UserLists">
  <Var File="c:\users\userlist1.txt"/>
</List>
</List>

```

Example—FSMount Section

The FSMount section of this example uses the same file system, pn-local, for each group of ISP users.

```

<List Name="FSMount">
...other mount points...
  <List Name="ISP Content (Washington users)">
    <Var ShortName="pn-local"/>
    <Var MountPoint="/wa_isp"/>
    <Var BasePath="c:\home\washington"/>
  </List>
  <List Name="ISP Content (Oregon users)">
    <Var ShortName="pn-local"/>
    <Var MountPoint="/or_isp"/>
    <Var BasePath="c:\home\oregon"/>
  </List>
  <List Name="ISP Content (Idaho users)">
    <Var ShortName="pn-local"/>
    <Var MountPoint="/id_isp"/>
    <Var BasePath="c:\home\idaho"/>
  </List>
...other mount points...
</List>

```

Example—User Directories

User directories are stored under directories according to the state where the accounts are based:

C:\home\washington\canderson
C:\home\washington\pbrown
C:\home\oregon\ladams
C:\home\oregon\schu
C:\home\idaho\alex
C:\home\idaho\sam
C:\home\idaho\tracy

Note

Notice that the actual directories where the users' content is stored is different from the /path/ shown in the user list file. The /path/ information is actually a method of grouping users.

How RealServer Looks for Users' Content (Dedicated Hosting)

On RealServers dedicated to ISP hosting, the process for locating files is slightly different:

1. When it receives a request for content, RealServer looks at the directories in the URL. It uses the number of directories indicated by *number* in the user list file.
2. Using the /path/ information in the user list, RealServer goes to the ISPHosting list of the configuration file, where it looks within the TranslationMounts list for a matching UserPath. Once it finds a match (it looks for the longest possible match), RealServer records the MountPoint located within the same list. This translates the logical path to a file system.
3. Next, RealServer looks in the FSMount list for a file system whose mount point matches the MountPoint from the ISPHosting list. Once it finds the correct mount point, RealServer notes the associated BasePath.
4. The media will be in a directory relative to BasePath.

Typically, users' content is mapped to a special ISP hosting mount point and base path. User directories are located under the base path.

IP Binding

The ability to reserve specific addresses for RealServer’s use is explained in Chapter 8, “Advanced Features”. This list uses variables numbered sequentially: Address_01, Address_02, and so on. Use one for each IP address you want to set aside for RealServer. Use the RealServer’s IP address or DNS name for each variable; however, the IP address allows RealServer to be more efficient.

RealServer will bind to the specified addresses only; it will not bind to localhost.

If you don’t use any values for the variables in the IPBindings list, RealServer binds to the host IP address and localhost. It does not bind to any others.

IP Bindings List Configuration Elements

Element	Description
<code><List Name="IPBindings"></code>	
<code><Var Address_01="0.0.0.0"/></code>	Each variable gives an address to reserve for use by RealServer. To reserve all addresses, set the address variable to 0.0.0.0 and remove all other address variables from the list. Use the IP address or DNS name, but RealServer will be more efficient if you use the IP address.
<code></List></code>	

Distributed Licensing

This feature is best configured with RealSystem Administrator; consult “Distributed Licensing” on page 109.

Live Archiving

The live archive feature is described in Chapter 11, “Unicasting Live Presentations”.

For every virtual directory of live streams that you want to archive, create a list. The list must have the same name as the virtual directory. To archive all streams that arrive at the main content directory, name the list with an asterisk (*).

When live archiving is enabled, RealServer examines all arriving live streams, and compares the names of the streams with the list names in the configuration file. If it contains a list whose name matches the virtual path name of the incoming live stream, RealServer will archive the file. If no matching list name is found, RealServer does not archive the file. Files are archived in locations specified by TargetDirectory.

Each list must include either TargetDirectory (to indicate where to store the archived streams) or NoArchive (to indicate that the streams should not be archived); optional variables are BandwidthNegotiation, FileSize, and FileTime.

Live Archiving Configuration Elements

Element	Description
<List Name="LiveArchive">	
<List Name="*">	An asterisk for a list name indicates the main content directory.
<Var TargetDirectory="/Archive"/>	The path where RealServer will create the archive files. The default is the Archive subdirectory of the Content directory. (This is called "Destination Path" in RealSystem Administrator.)
<Var FileSize="4"/>	Creates archive files of live streams by their size. Given in megabytes. If you give values to both FileTime and FileSize, RealServer will use the first, or lower, limit reached. To save entire broadcasts without limiting the file size, omit both FileTime and FileSize.
<Var BandwidthNegotiation="True"/>	Indicates that RealSystem 5-style bandwidth negotiation is in use.
</List>	
<List Name="concerts">	
<Var TargetDirectory="/Archive"/>	See description earlier in this section.

(Table Page 1 of 2)

Live Archiving Configuration Elements (continued)

Element	Description
<code><Var FileTime=1h"/></code>	Creates archive files of live broadcasts in segments of this length. Format is XdYhZm where X is the number of days, Y is the number of hours, and Z is the number of minutes. You must enter them in dhm order. See also FileSize. RealServer requires that the units be in the dhm order, so if you specify a subset, be sure to use the correct order. See "Example FileTime Values" table below.
<code></List></code>	
<code><List Name="secure"></code>	
<code><Var NoArchive="True"/></code>	When set to True, disables archiving of live files for the given directory.
<code></List></code>	
<code></List></code>	

(Table Page 2 of 2)

The table below shows sample values for FileTime.

Example FileTime Values

FileTime Value	Resulting File Contents
30m	Thirty-minutes
1h	One-hour
1h30m	One-and-a-half hours
1d1m	24 hours and one minute
1d1h	25 hours (one day plus one hour)
23h59m	23 hours and 59 minutes
1d1h1m	25 hours and one minute

Logging

Logging and reporting features are described in Chapter 19, “Reporting RealServer Activity”. Variables which control the locations of the access and error log files are described in “Paths” on page 418 of this chapter .

Logging Configuration Elements

Element	Description
Access Log Variables	
<code><Var LoggingStyle="3"/></code>	Determines how much data about clips served is gathered in the access log.
<code><Var StatsMask="0"/></code>	Determines how much data about clients is gathered in the access log.
<code><Var DisableClientGUID="0"/></code>	Collects unique client identifiers (“GUIDs”). When set to 1, ignores all client GUIDs and uses 00000000-0000-0000-0000-000000000000 instead. Refer to “Omitting Client Identifiers” on page 299.
<code><Var LogRollFrequency="4W"/></code>	Creates a new access log for each period specified. The period is indicated in the format <code>xD</code> , <code>xW</code> , or <code>xM</code> , where <code>x</code> is a number. See also <code>LogRollSize</code> . For example, <code>4D</code> will keep 4 days of information in the log file. Name of the rolled access log is based on the filename given by <code>LogPath</code> . For an explanation of the naming convention for rolled log files, see “Rolled Log File Format” on page 304.
<code><Var LogRollSize="5"/></code>	Creates a new access log when the indicated file size is reached, given in megabytes. See also <code>LogRollFrequency</code> . If you include both <code>LogRollFrequency</code> and <code>LogRollSize</code> , RealServer uses the variable with the limit reached first.
Error Log Variables	

(Table Page 1 of 2)

Logging Configuration Elements (continued)

Element	Description
<code><Var ErrorLogRollFrequency="1W"/></code>	Creates a new error log for each period specified. The period is indicated in the format xD, xW, or xM, where x is a number. See also LogRollSize. For example, 4D will keep 4 days of information in the log file. Name of the rolled access log is based on the filename given by ErrorLogPath. For an explanation of the naming convention for rolled log files, see “Rolled Log File Format” on page 304.
<code><Var ErrorLogRollSize="3"/></code>	Creates a new error log when the indicated file size is reached, given in megabytes. See also ErrorLogRollFrequency. If you include both ErrorLogRollFrequency and ErrorLogRollSize, RealServer uses the variable with the limit reached first.

(Table Page 2 of 2)

Disable access log file rolling by changing the LogRollFrequency and LogRollSize variables to 0. Disable error log file rolling by changing the ErrorLogRollFrequency and ErrorLogRollSize variables to 0.

MIME Types

Setting up RealServer to send correct MIME type information with clips is described in Chapter 6, “Starting and Stopping RealServer”.

MIME Types Configuration Elements

Element

```
<List Name="MimeTypes">
  <List Name="audio/x-pn-realaudio">
    <Var Extension_02="ram"/>
  </List>
  <List Name="image/gif">
    <Var Extension_01="gif"/>
  </List>
  <List Name="image/jpg">
    <Var Extension_01="jpg"/>
    <Var Extension_02="jpeg"/>
  </List>
  <List Name="text/html">
    <Var Extension_01="html"/>
    <Var Extension_02="htm"/>
  </List>
</List>
```

Multicasting

Two methods of multicasting are available: back-channel and scalable. Multicasting methods are described in Chapter 13, “Multicasting Live Presentations”. Both methods can send SAP information, described in the next section.

SAP Information

The SAP list gives information about the Session Announcement Protocol files that can be sent to programs configured to read them. See “Publicizing Your Multicasts” on page 195 for information.

In addition to the information on this list, you will also indicate whether to send SAP files in each scalable multicast list, and in the back-channel

multicast list. Three variables appear in the SAP list: ListenAnnouncement, SendAnnouncementEnabled, and HostAddress.

SAP Configuration Elements

Element	Description
<List Name="SAP">	
<Var ListenAnnouncement="True"/>	Indicates whether RealServer should listen for other SAP files.
<Var SendAnnouncementEnabled="True"/>	Indicates whether to send the SAP file with multicasts.
<Var HostAddress="address"/>	Address of the host RealServer, where the multicasts originate.
</List>	

Back-Channel Multicasting

Back-channel multicasting is described in “Back-Channel Multicasting” beginning on page 182.

Settings used with this list are Enabled, AnnounceSAP, AddressRange, DeliveryOnly, PNAPOrt, RTSPPort, Resend, and TTL.

Back-Channel Multicasting Configuration Elements

Element	Description
<List Name="Multicast">	Back-channel multicast section.
<Var Enabled="True"/>	True indicates that back-channel multicast is available.
<Var AnnounceSAP="True"/>	Indicates whether to send SAP files.
<Var PNAPOrt="7070"/>	Client port number to which RealServer will direct its streams. Default value is 7070.
<Var RTSPPort="554"/>	Client port number to which RealServer will direct its streams. Default value is 554.
<Var TTL="16"/>	Time To Live for multicast packets travelling over the network.
<Var Resend="True"/>	Allows or denies requests from clients for resends of missing UDP packets.

(Table Page 1 of 2)

Back-Channel Multicasting Configuration Elements (continued)

Element	Description
<code><Var AddressRange="address-address"/></code>	Range of addresses to which you want to send streams, in the form of <i>address-address</i> . RealServer uses the first available address in this range. If you are using other types of multicast, be sure that the address ranges are different and do not overlap. If your multicast streams are referenced in SMIL files, you will need one address for each stream.
<code><Var DeliveryOnly="False"/></code>	Requires clients listed in <code>Controllist</code> to receive only multicast transmissions from RealServer. When <code>DeliveryOnly</code> is <code>False</code> , clients on <code>Controllist</code> can receive both multicasts and unicasts. Default value is <code>False</code> . (In RealSystem Administrator this is called "Multicast Delivery Only")
<code><List Name="Controllist"></code>	The <code>Controllist</code> list gives the addresses of clients required to receive multicast transmissions. (Called "Client Access List" in RealSystem Administrator.)
<code><List Name="100"></code>	Rule number for this list. For information on choosing rule numbers, refer to Access Control documentation.
<code><Var Allow="172.16.2.24:255.0.0.0"/></code>	Address and netmask, separated by a colon, of clients allowed to receive multicast transmissions. Uses same format as <code>From</code> variable in <code>AccessControl</code> list. There must always be at least one entry in <code>Controllist</code> . The default value for <code>Allow</code> is <code>Any</code> , which allows all clients to receive multicast.
<code></List></code>	
<code><List Name="200"></code>	
<code><Var Allow="201.34.23.0:255.255.255.254"/></code>	
<code></List></code>	
<code></List></code>	
<code></List></code>	

(Table Page 2 of 2)

Scalable Multicasting

Unlike back-channel multicasting settings, scalable multicasting settings are located within the `FSMount` list. Scalable multicasting is described in "Scalable Multicasting" beginning on page 184.

Located within the FSMount list, scalable multicasting uses the following variables:

- AddressRange
- AnnounceSAP
- Enabled
- MountPoint
- Timeout
- PortRange
- ShortName
- TTL
- VirtualPath

Optional variables include ReuseAddress, AlternateURL, ShiftToUnicast, SendClientStatistics, WebServerAddress, WebServerPort, and WebServerCGIPath.

Create one list within the Sources list for every virtual path you want to make available for scalable multicasting.

Be sure to add the mount point to the HTTPDeliverable list.

Scalable Multicasting Configuration Elements

Element	Description
<code><List Name="Scalable Multicast"></code>	
<code><Var ShortName="pn-scalable"/></code>	Gives the short name of the plug-in file.
<code><Var MountPoint="/scalable"/></code>	Mount point used in all URLs for scalable multicasts.
<code><List Name="Sources"/></code>	Each list within this list represents a virtual directory that is to be streamed via scalable multicast.
<code><List Name="Concerts"></code>	Name of this list.
<code><Var VirtualPath="French"/></code>	Live streams encoded to the French virtual directory will be available via scalable multicast. To indicate that all live sources should be available for scalable multicast, use an asterisk (*) for the virtual path name.
<code><Var Enabled="True"/></code>	Enables scalable multicasting for this virtual directory.
<code><Var ReuseAddress="True"/></code>	Instructs RealServer to send both the audio and video streams for a given bit rate over the same address.
<code><Var AddressRange="231.1.1.1-231.1.1.10"/></code>	Range of addresses to which you want to send streams. RealServer uses the first available address in this range.

(Table Page 1 of 3)

Scalable Multicasting Configuration Elements (continued)

Element	Description
<Var PortRange="7300-7321"/>	Range of ports to which RealServer can send a multicast stream.
<Var AnnounceSAP="True"/>	Indicates whether to create and send an SAP file for this scalable multicast.
<Var TTL="16"/>	Time To Live for multicast packets travelling over the network.
<Var Timeout="60">	Number of seconds a client will wait for multicast packets before it stops or uses the AlternateURL address.
</List>	
<List Name="Live Concerts">	Name of this list.
<Var VirtualPath="Liveconcerts"/>	See description earlier in this section.
<Var Enabled="True"/>	See description earlier in this section.
<Var AddressRange="231.1.1.1-231.1.1.10"/>	See description earlier in this section.
<Var PortRange="7300-7320"/>	See description earlier in this section.
<Var TTL="16"/>	See description earlier in this section.
<Var Timeout="60">	See description earlier in this section.
<Var ShiftToUnicast="True"/>	Allows clients that cannot receive multicast to receive the presentation via unicast. The URL can refer to a unicast version of the stream, or to a Web page containing information about the broadcast.
<Var AlternateURL="rtsp://myserver.com:554/encoder/live.rm"/>	Alternate URL to which client will switch if multicast data is not received.
<Var SendClientStatistics="True"/>	Clients are instructed to send their connection statistics at the end of a scalable multicast or when the user stops the presentation.
<Var WebServerAddress=192.12.12.1"/>	Address of Web server or RealServer which will receive client statistics. Required when SendClientStatistics is set to True, even if you indicate that statistics should be sent to the RealServer machine.

(Table Page 2 of 3)

Scalable Multicasting Configuration Elements (continued)

Element	Description
<code><Var WebServerPort="9090"/></code>	Port on Web server or HTTPPort on RealServer which will receive client statistics. Required when SendClientStatistics is set to True, even if you indicate that statistics should be sent to the RealServer machine.
<code><Var WebServerCGIPath="cgi-bin/stats"/></code>	Location of CGI script on Web server which will collect client statistics. Optional.
<code></List></code>	
<code></List></code>	
<code></List></code>	

(Table Page 3 of 3)

Passwords

MonitorPassword is described in Chapter 18, “Monitoring RealServer Activity”.

Passwords Configuration Element

Element	Description
<code><Var MonitorPassword="letmein"/></code>	Password used by Java Monitor in connecting to RealServer.

Paths

LogPath and ErrorLogPath are described in Chapter 19, “Reporting RealServer Activity”. PluginDirectory is described on Chapter 7, “Customizing RealServer Features”. LicenseDirectory is given on Chapter 6, “Starting and Stopping RealServer”.

Windows Variables

Path variables, along with typical paths used in Windows and Windows NT, are shown here.

Paths Configuration Elements in Windows and Windows NT

Element	Description
<code><Var LogPath="C:\Program Files\Real\RealServer\Logs\rmaccess.log"/></code>	LogPath indicates where and with what name the access log file will be stored. If omitted, RealServer places rmaccess.log in the Logs directory.
<code><Var ErrorLogPath="C:\Program Files\Real\RealServer\Logs\rmerror.log"/></code>	ErrorLogPath gives the path and name of the error log file. If this setting is omitted, RealServer places rmerror.log in the Logs directory.
<code><Var PluginDirectory="C:\Program Files\Real\RealServer\Plugins"/></code>	Shows where the plug-in files are stored.
<code><Var SupportPluginDirectory="C:\Program Files\Real\RealServer\Lib"/></code>	Shows location of the Lib directory, where files used by G2SLTA, as well as encnet.dll (Windows) and encnet.so.6.0 (UNIX) are stored.
<code><Var LicenseDirectory="C:\Program File\Real\RealServer\License"/></code>	Gives the location of the license files.

UNIX Variables

One additional setting is found on RealServer running on a UNIX system: PidPath. See “Process ID (PID)” on page 116.

Paths Configuration Elements in UNIX

Element	Description
<code><Var LogPath="/usr/bin/RealServer/Logs/rmaccess.log"/></code>	LogPath indicates where and with what name the access log file will be stored. If omitted, RealServer places rmaccess.log in the Logs directory.
<code><Var ErrorLogPath="/usr/bin/RealServer/Logs/rmerror.log"/></code>	ErrorLogPath gives the path and name of the error log file. If this setting is omitted, RealServer places rmerror.log in the Logs directory.
<code><Var PluginDirectory="/usr/bin/RealServer/Plugins"/></code>	Shows where the plug-in files are stored.
<code><Var SupportPluginDirectory="/usr/bin/RealServer/Lib"/></code>	Shows location of the Lib directory, where files used by G2SLTA are stored.

(Table Page 1 of 2)

Paths Configuration Elements in UNIX (continued)

Element	Description
<code><Var LicenseDirectory="/usr/bin/RealServer/License"/></code>	Gives the location of the license files.
<code><Var PidPath="/usr/bin/RealServer/Logs/rmserver.pid"/></code>	In UNIX systems, the location of the process id file.

(Table Page 2 of 2)

Ports

Port settings for RTSPPort, PNAPort, and HTTPPort are described in Chapter 7, “Customizing RealServer Features”. MonitorPort is described in Chapter 18, “Monitoring RealServer Activity”.

Ports Configuration Elements

Element	Description
<code><Var RTSPPort="554"/></code>	Where RealServer listens for RTSP requests. Default value is 554.
<code><Var PNAPort="7070"/></code>	Where RealServer listens for PNA requests. Default value is 7070.
<code><Var HTTPPort="8080"/></code>	Where RealServer listens for HTTP requests. Default value is 80 or 8080 if port 80 was unavailable during installation.
<code><Var MonitorPort="9090"/></code>	The port which monitors (such as Java Monitor) connect to RealServer. Default value is 9090.
<code><Var AdminPort="7845"/></code>	Port number for RealSystem Administrator connection. No default value; the port number is given a random number during Setup.

Ramgen

Ramgen is described in “Ram Files and Ramgen” on page 74 and in *RealSystem Production Guide*. There are only two variables associated with Ramgen: ShortName and MountPoint.

This list is located within the FSMount section.

Ramgen Configuration Elements

Element	Description
<List Name="RAM File Generator">	
<Var ShortName="pn-ramgen"/>	The short name of the ram file generator is pn-ramgen.
<Var MountPoint="/ramgen/">	The default mount point is /ramgen/.
</List>	

RealSystem Administrator

Two file systems work together to operate RealSystem Administrator: the local file system and the administration file system.

The administration file system accepts the initial URL for RealSystem Administrator. It requests the HTML files from the local file system. Once the local file system delivers the HTML files, the administration file system looks up your RealServer's values and displays them at the appropriate points in RealSystem Administrator.

Three variables are used for the RealAdministrator list: ShortName, MountPoint, and BasePath.

Five variables are use in the RealAdministrator_Files list: ShortName, MountPoint, Authorized_User_Group, Authentication, and Realm.

This tool is described in Chapter 7, "Customizing RealServer Features".

RealSystem Administrator Configuration Elements

Element	Description
<!-- Local File System; HTML -->	Location and files used by RealSystem Administrator.
<List Name="RealSystem Administrator HTML">	
<Var ShortName="pn-local"/>	RealSystem Administrator uses the local file system.

(Table Page 1 of 4)

RealSystem Administrator Configuration Elements (continued)

Element	Description
<Var MountPoint="/admin/html/" />	Mount point, used when RealAdministrator_Files list requests files from this plug-in. The default value is /admin/html/. If you change this, be sure to change the RealAdministrator_Files list's BaseMountPoint to match.
<Var BasePath="C:\Program Files\Real\RealServer\RealAdministrator" />	Location of the RealSystem Administrator files.
</List>	
<!-- Local File System; DOCS-->	The HTML version of this guide is served with this information.
<List Name="RealSystem Administrator DOCS">	
<Var ShortName="pn-local" />	RealSystem Administrator uses the local file system.
<Var MountPoint="/admin/Docs/" />	Mount point used for files in the guide.
<Var BasePath="C:\Program Files\Real\RealServer\RealAdministrator\Docs" />	Main location of the HTML guide files.
</List>	
<!-- Local File System; JAVAMONITOR -->	Information and file locations for Java Monitor files.
<List Name="RealSystem Administrator JAVAMONITOR">	
<Var ShortName="pn-local" />	Java Monitor uses the local file system.
<Var MountPoint="/admin/JavaMonitor/" />	Mount point used for referencing the monitor.
<Var BasePath="C:\Program Files\Real\RealServer\RealAdministrator\JavaMonitor" />	Main location of the Java Monitor files.
</List>	
<!-- Local File System; IMAGES -->	Information and file locations for graphics used by RealSystem Administrator.
<List Name="RealSystem Administrator IMAGES">	

(Table Page 2 of 4)

RealSystem Administrator Configuration Elements (continued)

Element	Description
<code><Var ShortName="pn-local"/></code>	RealSystem Administrator uses the local file system.
<code><Var MountPoint="/admin/images/"></code>	Mount point used by RealSystem Administrator in linking to graphics.
<code><Var BasePath="C:\Program Files\Real \RealServer\RealAdministrator\images"/></code>	Main location of graphics used in RealSystem Administrator.
<code></List></code>	
<code><!-- XML Tag Handler File System --></code>	
<code><List Name="RealSystem Administrator SSI"></code>	Server-side include handler; creates HTML pages in RealSystem Administrator.
<code><Var ShortName="pn-xmltag"/></code>	
<code><Var MountPoint="/admin/includes/"></code>	Mount point used in requests
<code><Var BaseMountPoint="/admin/html/"></code>	Mount point of RealSystem Administrator.
<code><List Name="TagHandlers"></code>	List of plugins used for interpreting XML tags.
<code><Var h1="pn-includer"/></code>	
<code><Var h2="pn-vsrectaghdr"/></code>	
<code></List></code>	
<code></List></code>	
<code><!-- Admin File System --></code>	
<code><List Name="RealAdministrator_Files"></code>	
<code><Var ShortName="pn-admin"></code>	RealSystem Administrator uses the pn-admin plug-in.
<code><Var MountPoint="/admin/"></code>	The default value for MountPoint is /admin/. If you change this, you will need to type a new URL to connect to RealSystem Administrator.
<code><Var BaseMountPoint="/admin/includes/"></code>	This special form of mount point reflects the mount point of the RealAdministrator list.

(Table Page 3 of 4)

RealSystem Administrator Configuration Elements (continued)

Element	Description
<Var Realm="company.AdminRealm"/>	The Realm variable identifies which AuthenticationRealm settings will be used with requests sent to the RealSystem Administrator mount point.
</List>	

(Table Page 4 of 4)

Splitting

Chapter 12, “Splitting Live Presentations” describes this feature.

In addition to the special settings required on the transmitter and on the receiver, splitting information is located in the FSMount list.

All the settings shown are required; only MountPoint can be changed by the user.

FSMount Splitting Configuration Elements

Element	Description
<List Name="FSMount">	
...	
<List Name="RealSystem Broadcast Distribution">	FSMount entry for splitting.
<Var ShortName="pn-broadcast-receiver-fs"/>	Short name of this plug-in.
<Var MountPoint="/broadcast/">	Mount point to use in broadcasts.
</List>	
...	
</List>	

Transmitter Settings

Settings necessary for the transmitter to send its live streams to receivers are shown below. The following variables are optional on the transmitter:

- AcquisitionDataInterval
- RedundancySendInterval

- FECLevel
- Protocol
- ResendSupported
- TTL

Splitting Configuration Elements—Transmitter Settings

Element	Description
<code><List Name="BroadcastDistribution"></code>	
<code><Var SourceName="example"/></code>	Name of this transmitter. Set to the host name during Setup.
<code><List Name="Destinations"></code>	List of receivers.
<code><List Name="Destinations_0"></code>	First rule for a broadcast stream.
<code><Var AcquisitionDataInterval="30"/></code>	Frequency, in seconds, that the transmitter sends metadata information. The range is 0 to 60.
<code><Var PathPrefix="/encoder"/></code>	Virtual path name of this broadcast stream. The default value is <code>/encoder/</code> .
<code><Var Address="australia.example.com"/></code>	Name of receiver for this broadcast stream.
<code><Var Protocol="udp/unicast"/></code>	Protocol to use in transmitting streams. Refer to RealSystem Administrator for a list of options. Both the transmitter and the receiver must use the same values.
<code><Var FECLevel="10"/></code>	Forward Error Correction rate—the percent of corrective packets sent. A higher number results in better quality, but consumes more network bandwidth. The default value for use over the Internet is 20. If you're using splitting within a network, you can set this value to 0, since you are unlikely to need error correction. Range of values is 0 to 100.
<code><Var RedundancySendInterval="1000"/></code>	Interval in milliseconds that elapses before repeat packets are sent by the transmitter to the receiver. In use only when FECLevel is set to 100. This variable is not added by RealSystem Administrator; it must be added by hand.

(Table Page 1 of 3)

Splitting Configuration Elements—Transmitter Settings (continued)

Element	Description
<Var ResendSupported="0"/>	Whether receivers' resend requests are honored. 0 means they are ignored; 1 indicates the transmitter will resend packets. (In RealSystem Administrator, this is shown as Honor Resend Requests .) Default is 0.
<Var SessionTimeout="30000"/>	Time (in milliseconds) that the receiver waits before cancelling a connection with the transmitter in which no data is received. Default is 30000. This variable is not added by RealSystem Administrator; it must be added by hand.
<Var RelayMode="0"/>	Permits transmitters and receivers to be configured in a chain. Default value is 0, or False; change to 1 to enable chaining. Refer to "Chaining Receivers" on page 173.
<List Name="Security">	Type of security in use. If security is in use here, the receiver must use matching settings.
<Var Type="basic"/>	Options are None and Basic. Choosing Basic means you elect to encrypt the transmission between transmitter and receiver.
<Var Password="letmein"/>	User-defined password to use when the security Type is set to Basic.
</List>	
</List>	
<List Name="Pull Settings">	Pull splitting feature.
<List Name="PullSource1">	Name for this rule.
<List Name="Security">	Security settings for pull splitting; these may be different than those used in regular splitting (shown above).
<Var Type="basic"/>	See description earlier in this section.
<Var Password="letmein"/>	See description earlier in this section.
</List>	

(Table Page 2 of 3)

Splitting Configuration Elements—Transmitter Settings (continued)

Element	Description
<Var ListenPort="2030"/>	Port number on which this transmitter should listen for resend requests. If you leave this field blank, RealServer randomly sets the value. Type here to override RealServer's setting.
<Var PathPrefix="/encoder"/>	
<Var AcquisitionDataInterval="30"/>	Frequency, in seconds, that the transmitter sends metadata information. The range is 0 to 60.
</List>	
</List>	
</List>	
</List>	

(Table Page 3 of 3)

Receiver Settings

Settings necessary for the receiver are shown below. Optional settings are:

- MulticastAddress
- ResendSupported
- Protocol
- SessionTimeout

If pull splitting is in use, these settings are optional:

- AcquisitionDataInterval
- UseTCPForPullBackchannel
- FECLevel

Splitting Configuration Elements—Receiver Settings

Element	Description
<List Name="BroadcastReceiver">	
<List Name="Receivers">	
<List Name="Transmitter1">	
<Var OriginSpec="japan.example.com"/>	Address or host name of transmitter.
<Var PortRange="30000-30050"/>	Port range on receiver where RealServer listens for the broadcast. Must match port range used on transmitter.

(Table Page 1 of 3)

Splitting Configuration Elements—Receiver Settings (continued)

Element	Description
<Var MulticastAddress="address"/>	Class D address for receiver to join the broadcast. Used only if Protocol is set to udp/multicast.
<Var SAPTransmissionEnabled="1"/>	Indicates that SAP files may be used for multicast streams. Used only if Protocol is set to udp/multicast. Refer to “Sending SAP Information with Your Multicasts” on page 204. This variable is not added by RealSystem Administrator; it must be added by hand.
<Var TTL="1"/>	Time-to-live for multicast packets. Used only if Protocol is set to udp/multicast. For a list of values, refer to the “Time to Live (TTL) Values” table on page 197.
<Var Protocol="udp/unicast"/>	Protocol; must match the setting on the transmitter. If omitted, RealServer uses udp/unicast. Refer to RealSystem Administrator for a list of options.
<Var ResendSupported="0"/>	Whether to make resend requests to the receiver. 1 means to make requests. Produces higher quality but requires more overhead.
<Var SessionTimeout="30000"/>	Duration of time (in milliseconds) the receiver waits to stop the session with the transmitter after no packets are received.
<Var PushBufferingTime="6"/>	Interval (in seconds) that the receiver maintains a packet buffer before serving it to clients. This variable is not added by RealSystem Administrator; it must be added by hand.
<List Name="Security">	These settings must match those used on transmitter.
<Var Type="basic"/>	
<Var Password="letmein"/>	
</List>	
<Var PullSplitEnabled="1"/>	Pull splitting is in use for this rule. Default value is 0 (off).

(Table Page 2 of 3)

Splitting Configuration Elements—Receiver Settings (continued)

Element	Description
<code><Var UseTCPForPullBackchannel="0"/></code>	When set to 1, creates a back channel for initiation and resend requests. Used only when Protocol is set to udp/unicast or udp/multicast. The default value is 0. This variable is not added by RealSystem Administrator; it must be added by hand.
<code><Var PathPrefix="/pull/" /></code>	Path that indicates to RealServer that this is a pull splitting request. It does not need to be unique for this broadcast.
<code><Var FECLevel="20"/></code>	Forward Error Correction rate—the percent of corrective packets sent. A higher number results in better quality, but consumes more network bandwidth. The default value for use over the Internet is 20. If you're using splitting within a network, you can set this value to 0, since you are unlikely to need error correction. Range of values is 0 to 100.
<code><Var AcquisitionDataInterval="30"/></code>	Frequency, in seconds, that the transmitter sends metadata information. The range is 0 to 60.
<code></List></code>	
<code></List></code>	

(Table Page 3 of 3)

UNIX-Only Settings

These settings are also described in “UNIX-Only Features” on page 115.

UNIX-Only Configuration Elements

Element	Description
<code><Var Group="users"/></code>	Group name under which RealServer runs. The group name must already exist on the computer on which RealServer is running; otherwise, RealServer will not start. If you do not specify a group name, this variable defaults to the group name of the user who first starts RealServer. The default value is %-1.

(Table Page 1 of 2)

UNIX-Only Configuration Elements (continued)

Element	Description
<code><Var User="canderson"/></code>	User name under which RealServer runs. The user name must exist on the computer on which RealServer is running; otherwise, RealServer will not start. If you don't specify a user name during Setup, the user name defaults to the user name of the user who first logs in and starts RealServer. The default value is %-1.

(Table Page 2 of 2)

View Source

The view source feature is described in “Displaying Source Code for SMIL Files and Media Clips” on page 99.

The following variables are in use for view source:

- ViewSourceLongName
- Path
- Mount
- AllowViewSource
- HidePaths

In addition, the view source feature adds settings to other lists within the configuration file. It adds a plug-in listing to the Real System Administrator SSI list. Also, it requires an entry in the HTTPDeliverable list.

View Source Configuration Elements

Element	Description
<code><!-- V I E W S O U R C E --></code>	
<code><List Name="ViewSourceConfiguration"></code>	
<code><Var ViewSourceLongName="View Source Tag FileSystem"/></code>	
<code><List Name="/"></code>	Mount point or virtual path to which the following settings apply. Create one such sublist for each mount point or path.
<code><Var AllowViewSource="1"/></code>	Enables the view source feature (when set to 1) or disables the feature (when set to 0).

(Table Page 1 of 2)

View Source Configuration Elements (continued)

Element	Description
<Var HidePaths="1"/>	Replaces paths with ellipses (when set to 1). Displays the entire path (when set to 0).
</List>	
</List>	

(Table Page 2 of 2)

Settings in the Content Browsing section refer to the content browsing feature of view source.

Variables are:

- Mount (Mount_1, Mount_2, and so on)
- Ext (Ext_1, Ext_2, and so on)

View Source Content Browsing Elements

<!-- C O N T E N T B R O W S I N G -->	Content browsing section of configuration file.
<List Name="ContentBrowsing">	
<List Name="BrowsableMountPoints">	List of mount points for which content browsing is enabled.
<Var Mount_1="/"/>	Create one Mount variable for each mount point that you want to be able to browse.
</List>	
<List Name="IndexExtensions">	List of file extensions which will be included in content browsing.
<Var Ext_1="*" />	An asterisk (*) refers to all extensions. Otherwise, create one Ext variable for each extension. For example, <Var Ext_2=".smi"/>.
</List>	
</List>	

The following entries appear within the FSMount list.

View Source FSMount Configuration Elements

<List Name="View Source File System">	
<Var ShortName="pn-vsrfcfsys"/>	Plug-in short name.

(Table Page 1 of 2)

View Source FSMount Configuration Elements

<code><Var MountPoint="/vsrdfs/" /></code>	Mount point used (within RealServer) for view source requests.
<code></List></code>	
<code><!-- View Source Tag File System; Source Insertion --></code>	
<code><List Name="View Source Tag FileSystem"></code>	
<code><Var ShortName="pn-xmltag" /></code>	Plug-in used by tag handler.
<code><Var MountPoint="/viewsource/" /></code>	Mount point for view source requests.
<code><Var BaseMountPoint="/vsrdfs/" /></code>	Original mount point for view source requests.
<code><List Name="TagHandlers"></code>	Plug-in used by tag handler.
<code><List Name="ViewSource Tag Handler"></code>	
<code><Var ShortName="pn-vsrtaghdr" /></code>	
<code></List></code>	
<code></List></code>	
<code></List></code>	

(Table Page 2 of 2)

Features Available Only Through Direct Editing

Some of the more specialized lists and variables are only configurable by editing the configuration file directly; they cannot be changed via RealSystem Administrator.

These elements are:

- Most settings that would affect the use of RealSystem Administrator, such as the file systems used by the various RealSystem Administrator components.
- Short names of plug-ins, which most users are unlikely to change.
- Platform-specific variables, such as those described in “UNIX-Only Settings” (Group and User) on page 429.
- PidPath variable. See “UNIX Variables” on page 419.
- HTTPPostable list, for specifying which directories can receive data. Described in “HTTP Support” on page 400.
- LicenseDirectory variable, which tells RealServer where to look for the license key file. Described in “Paths” on page 418.
- MinPlayerProtocol variable, for giving the minimum client version that can receive content. See “Allowance” on page 386.
- MonitorPassword variable, the password used by RealSystem Administrator in connecting to the Java Monitor. Described in “Passwords” on page 418.
- PluginDirectory variable; gives the location of the Plugin directory. See “Paths” on page 418.
- Some splitting variables, specifically SAPTransmissionEnabled, RedundancySendInterval, SessionTimeout, and UseTCPForPullBackchannel. See “Splitting” on page 424.
- SupportPluginDirectory variable; gives the location of the Lib directory. See “Paths” on page 418.



INDEX

Symbols * (asterisk)

- in backup splitting, 169, 173
- in content browsing, 104
- in ISP hosting, 263, 264, 265
- in live archiving, 152, 154, 155, 156
- in user list files, 266

/ (slash)

- in content browsing, 104
- in view source list, 430
- mount point, 72

~ (tilde)

- in ISP hosting, 264, 404
 - access log, 258, 301
 - links, 255, 261
 - user list format, 263, 264

A Access control

- ad streaming and, 37
- authentication and, 37, 215, 226
- described, 214
- firewalls and, 119
- G2SLTA and, 37, 51, 215
- ISP hosting and, 37, 215, 257
- Java Monitor and, 215
- list, 385, 396
- live archiving and, 37
- logs and, 37, 216, 286
- monitoring and, 37
- multicasting and, 37, 189, 215
- RealProxy and, 37, 105, 215
- setting up, 219, 220
- splitting and, 37, 162, 215
- streaming and, 37, 138, 215
- troubleshooting, 337, 341, 342, 344, 345
- unicasting and, 37, 145, 215
- versus authentication, 224
- Access log, 129, 206, 283

- cache log and, 305
- configuration file, 419
- customizing, 297
- described, 283
- firewalls and, 119, 129
- format, 288, 292
- ISP hosting and, 258, 259
- multicasting and, 190, 208
- reading, 288
- rolling, 304

Access variable, 385

access_log table, 250, 253

accesslog.txt, 246, 248, 249

Account-based hosting, 256, 259, 261

AcquisitionDataInterval variable, 424, 425, 427, 429

“Ad Application” error messages

“Ad Insertion failed!”, 346

“Error retrieving the following image”, 348

“No AdRetrievalMountPoint”, 348

“No AdURL was specified...”, 348

“No appropriate Ad anchor...”, 347

“No HTTP file system mount point...”, 348

“No RotationMountPoint...”, 348

“The Ad Insertion Plugin...”, 347

“The AdURL specified...”, 347

“The connection to the AdURL timed out...”, 347, 348

Ad streaming

access control and, 37

ad "type" explanation, 310

ad server integration

direct, 312

overriding target through SMIL, 325

through HTML page, 313

- ad server type
 - background, 321
 - setting, 320
- AdForce brand ad serving, 320
- audience targets, 311
- base mount point
 - choosing, 317
 - security risks, 317
- clickthrough URLs, 312
- content creation issues, 310
- cookies, 312
- DoubleClick brand ad serving, 320
- Engage brand ad serving, 320
- firewalls and, 37
- G2SLTA and, 37
- in license, 38
- latency reduction, 314
- mount points
 - creating, 318
 - overriding, 325
 - overview, 316
- NetGravity brand ad serving, 320
- overview, 307
- <RealAdInsert/ > tags, 315
- relative URL resolution, 320
- rotating banner ads
 - formats, 311
 - overriding settings, 325
 - rotation interval
 - overriding, 325
 - setting, 322
 - setting up, 322
 - SMIL generation, 330
 - start-up image
 - overriding, 325
 - setting, 323
 - streaming bit rate
 - overriding, 325
 - setting, 322
- SMIL generation
 - ad types, 330
 - background color, 331
 - height and width, 330
 - inner and outer padding, 331
 - layout, 330
 - limitations, 326
 - mount points
 - creating, 328
 - overview, 327
 - relative to ad mount point, 327
 - options, 330
 - overview, 326
 - playlist disabling, 332
 - rotating banner ads, 330
 - streaming media formats, 308
 - target HTML page
 - assigning mount points to, 319
 - generating through ad server, 314
 - image URLs, 320
 - overriding URL through SMIL, 325
 - realad variable, 314
 - setting up, 313
 - target URL, 310
 - timeouts
 - connection, 324
 - server, 324
 - troubleshooting, 346
- Address
 - in links, 70
 - reserving for multicasting, 191
- Address Range
 - multicasting, 197, 202
 - variable, 414, 415, 416, 417
- Address translation firewall, 131
- Address variable, 425
- Address_01 variable, 408
- Admin mount point, 212
 - in HTTP Delivery list, 400, 401
- Admin Port, 96
 - variable, 338, 420
- Admin/html mount point, 422
 - base mount point, 423
- Admin/includes mount point, 423
- Adminfs mount point, 423
- AdminPort variable, 338
- Advertising, *see* Ad streaming
- Alerts, in NT performance monitor, 281
- Allow Duplicate IDs, 345
 - variable, 390, 391
- Allow variable, 415

- Allowance plugin, 386
 - AllowDuplicateIDs variable, 390
 - AllowViewSource variable, 430
 - Alternate URL, 206
 - variable, 416, 417
 - AnnounceSAP variable, 414, 416, 417
 - Announcing multicasts, 201
 - Application-level proxy firewall, 128, 131
 - Archiving, 340
 - described, 30
 - in configuration file, 408
 - view source and, 101
 - Archiving live broadcasts, 150
 - Asterisk (*)
 - as list name, 409
 - in live archiving, 409
 - Authentication
 - access control and, 37, 211, 214, 215, 226
 - access log, 301
 - cached and, 107
 - described, 223
 - firewalls and, 37, 119, 225
 - G2SLTA and, 37, 51, 225
 - in configuration file, 387
 - in license, 38, 39
 - ISP hosting and, 226, 257
 - Java Monitor and, 226
 - link format
 - archive live content, 369
 - live content, 370
 - live archiving and, 37, 225
 - location of clips, 82
 - logs and, 37, 226, 286, 288
 - monitoring and, 37
 - mount point, 72
 - multicasting and, 37, 183, 184, 186, 189, 225
 - of encoder users, 223, 227
 - of RealSystem Administrator users, 223, 228
 - of users, 223
 - RealProxy and, 37, 106, 225
 - splitting and, 37, 162, 225
 - streaming and, 37, 138, 224
 - troubleshooting, 345, 346
 - unicasting and, 37, 45, 145, 147, 225
 - variable, 421
 - view source and, 101
 - Authentication Realm variable, 388, 390
 - Authorized_User_Group variable, 421
- B**
- “Back-channel multicast is enabled...” error message, 344
 - Back-channel multicasting
 - defined, 182
 - See also* multicasting
 - Bandwidth
 - limiting, 212
 - negotiation, 141, 142, 156, 157
 - variable, 409
 - Banner ads, *see* Ad streaming
 - Base Path, 140
 - in ISP hosting, 405
 - in local file system, 399
 - setting up, 96
 - variable, 400, 421
 - in ISP hosting, 404, 406, 407
 - in RealAdministrator list, 422
 - Base path, 72
 - described, 73
 - BaseMountPoint variable, 422, 423
 - broadcast
 - mount point, 424
 - Broadcast mount point, 72
 - BroadcastReceiver list, 427
 - BrowsableMountPoints list, 431
- C**
- Cache log, 306, 397
 - Cache Port, 107, 397
 - Cache Requests, 108, 397
 - cache.log file, 305, 397
 - Channels, 120
 - Chart *See* graph
 - CLICK, in access log, 296
 - Client Access List, 197
 - Client Connections, 212
 - variable, 386, 387
 - Client IP Address, 197, 199

- Client Netmask, 197, 200
 - CloakingHint variable, 114
 - cloakport, 114
 - Clustering *See* splitting
 - Comment tag, 379
 - CommerceRules list, 390
 - Configuration file
 - components, 379
 - editing with text editor, 94, 379, 433
 - Connections
 - in ISP hosting, 256, 263
 - license, 38
 - limiting, 212, 350
 - “The content you requested...” message, 352
 - Control channel
 - described, 120
 - Control List, 344
 - variable, 415
 - “Could not open port 7070” error message, 334
- D**
- Daisy-chaining, 173
 - Data channel
 - described, 120
 - Data types
 - license, 38
 - Database ID
 - variable, 389, 390, 391, 392
 - Databases, 241
 - DB Name variable, 394
 - DBLoginName variable, 394
 - DBLoginPassword variable, 394
 - in Database list, 394
 - DBLoginUsername variable, 394
 - DBName variable, 394
 - Dedicated hosting, 259
 - described, 259
 - directory structures, 261
 - Delivery Only, 343
 - variable, 414, 415
 - De-Militarized Zone (DMZ) *See* firewalls
 - Destination Path, 156, 341
 - Directory_01 variable, 397
 - DisableClientGUID variable, 411
 - Distributed licensing, 109
- E**
- Edit Client Access List Number, 199
 - Enabled variable, 395, 414, 416, 417
 - in Multicast list, 414
 - in Scalable Multicast list, 416
 - Encoder Authentication Realm, 228
 - Encoder mount point, 72, 340, 361
 - in configuration file
 - in scalable multicast, 417
 - inG2_Encoders list, 398
 - in links to back-channel multicasts, 375
 - in links to G2SLTA content, 370
 - in links to live content, 367
 - in links to live unicast, 148
 - in links to pull split content, 374
 - in live unicasting, 146
 - in SecureLiveContent list, 391
 - Encoder_RNS list, 393
 - EncoderRealm variable, 397, 398
 - Error log, 39, 208, 419
 - described, 283
 - format, 303
 - rolling, 304
 - See also* logs
 - use in troubleshooting, 333, 340, 342, 343
 - Error Log Path
 - variable, 419
 - Error Log Roll Frequency, 304
 - variable, 412
 - Error Log Roll Size, 304
 - variable, 412
 - Error messages *See text of message*
 - “Error retrieving URL...” error message, 354, 340
 - ErrorMessage variable, 396
 - Evaluate Permissions
 - variable, 390
 - Evaluate permissions, 237
 - variable, 390, 391
 - Ext variable, 431
 - Extensible Markup Language (XML) *See* XML

- Extension_01 variable, 381, 413
- F**
 - Features in RealServer, 6
 - FECLevel variable, 425, 427, 429
 - “File not found” error message, 113, 335, 351
 - File Size
 - variable, 409, 410
 - File systems, 399
 - File Time
 - variable, 409, 410
 - File variable, 402
 - in ISP Hosting list, 403
 - “The file you requested cannot be streamed...” error message, 353
 - Firewalls, 117, 119
 - access control and, 37, 119
 - ad streaming and, 37
 - authentication and, 37, 119, 225
 - described, 117
 - encoders and, 27
 - G2SLTA and, 37
 - ISP hosting and, 37, 120
 - multicasting and, 119, 189
 - poor service, 354
 - RealProxy and, 119
 - RealServer and, 26
 - splitting and, 119, 123, 162
 - streaming and, 37, 119, 138
 - types, 128
 - unicasting and, 37, 119, 145
 - From variable, 385, 386
 - FSMount list, 399, 416
 - Authentication Realms and, 388
 - described, 399
 - in ISP hosting, 405
 - ISP hosting and, 402, 404
 - scalable multicasting and, 415
- G**
 - G2 Java Monitor *See* Java Monitor
 - G2SLTA
 - access control and, 37, 51, 215
 - access log, 301
 - authentication and, 37, 51, 225
 - Java Monitor and, 274
 - live archiving and, 50
 - logs and, 37, 51, 284
 - monitoring and, 37, 51
 - multicasting and, 187
 - splitting and, 37, 50, 161
 - streaming and, 50, 138
 - syntax, 54
 - troubleshooting, 341
 - unicasting and, 37, 50, 145
 - view source and, 101
 - G2SLTA_PLUGIN_PATH environment variable, 54
 - G2SLTA_SUPPORT_PATH environment variable, 54
 - GET, in access log, 289
 - “GIF (or JPEG)...” error messages
 - “Bad URL-encoded reliable flag.”, 349
 - “Bad URL-encoded background color.”, 349
 - “Bad URL-encoded bitrate.”, 349
 - “Bad URL-encoded target.”, 349
 - “Bad URL-encoded url.”, 349
 - “Cannot target browser...”, 350
 - “Illegal time formatting...”, 350
 - “Unknown player command...”, 350
 - Graph of RealServer activity
 - Java Monitor, 273
 - Windows NT Performance Monitor, 281
 - Group
 - in authentication, 240
 - variable, 115, 389
 - in AuthenticationRealms list, 388
 - UNIX group name, 429
 - GUID Registration Prefix variable, 391, 392
- H**
 - HidePaths variable, 430, 431
 - Host Address variable, 414
 - Host Name
 - variable
 - in Database list, 393, 394
 - HTTP Delivery
 - described, 212
 - HTTP download, 124
 - Ramgen, 351
 - scalable multicasting, 207, 208

- HTTP Directories, 212
- HTTP Port, 95
 - described, 95
 - firewalls and, 125
 - in Access control list, 222
 - in access control list, 222, 386
 - in links, 361
 - in on-demand streaming, 140
 - links, 65, 141, 148
 - reasons for changing, 113, 125
 - variable, 386, 420
 - Web server and RealServer on same system, 113
- HTTPODeliverable
 - list, 401
 - variable, 416, 430
- Httpfs mount point
 - in HTTP Delivery list, 401
- HTTPOPostable list, 401
 - changing, 433
- I**
 - IndexExtensions list, 431
 - “Insufficient bandwidth” error message, 352
 - Invalid license file, 39
 - “Invalid player” error message, 352
 - “Invalid player IP Address” error message, 345
 - “Invalid version” error message, 352
 - IP Address Range, 197
 - IP Bindings
 - described, 113
 - list, 408
 - use in troubleshooting, 334
 - ISP hosting
 - access control and, 37, 215
 - and firewalls, 120
 - authentication and, 226
 - configuration file elements, 402
 - dedicated hosting, 261
 - described, 255
 - firewalls and, 37
 - in access log, 301
 - in license, 38
 - Java Monitor and, 258
 - list, 402
 - logs and, 37, 286
 - monitoring and, 37
 - RealProxy and, 106, 258
 - setting up, 262
 - streaming and, 37
 - unicasting and, 146
- J**
 - Java class files, 279
 - Java Monitor, 340, 420, 422
 - access control and, 37, 215
 - applet mode, 279
 - application mode, 279
 - authentication and, 37, 226
 - back-channel multicasting, 182
 - described, 273
 - file system, 422
 - G2SLTA and, 37, 51, 274
 - ISP hosting and, 258
 - live archiving and, 274
 - logs and, 286
 - Monitor Port, 420
 - MonitorPassword variable, 418
 - changing, 433
 - multicasting and, 190
 - splitting and, 274
 - streaming and, 37, 139
 - troubleshooting, 346
 - troubleshooting and, 340
 - unicasting and, 37, 146
 - Javascript
 - errors, 339
- L**
 - License Directory
 - variable, 38, 418, 419, 420
 - “License exceeded” error message, 351
 - License information, 6, 333, 351
 - LicenseDirectory variable
 - changing, 433
 - Licensing, distributed, 109
 - Limiting access
 - by bandwidth, 212
 - by IP address, 214
 - by player version, 213
 - to HTML pages, 212
 - to multicast clients, 197

- Limiting connections, 212
 - Links
 - authentication, 231
 - ad streaming, 365
 - archiving, 369
 - back-channel multicasting, 375
 - format, 360
 - G2SLTA, 370
 - scalable multicasting, 376
 - streaming, 362
 - unicasting, 367, 368, 371
 - back-channel multicasting, 198
 - dedicated ISP hosting, 259
 - G2SLTA, 57
 - ISP hosting, 255, 267
 - live archiving, 153
 - ports, 70
 - protocols, 70
 - pull splitting, 178, 179
 - push splitting, 169
 - scalable multicasting, 204
 - streaming, 141
 - unicasting, 148
 - List tag, 380
 - ListenAnnouncement variable, 414
 - ListenPort variable, 427
 - Live Archive list, 409
 - Live archiving
 - ad streaming, 37
 - authentication and, 225
 - G2SLTA and, 50
 - Java Monitor and, 274
 - links, 153
 - multicasting and, 152, 187
 - splitting and, 37, 152, 161
 - streaming and, 138, 152
 - troubleshooting, 341
 - unicasting and, 37, 144
 - Live mount point
 - (pre-G2 content), 368, 398
 - (pre-G2), 147
 - “Livefile codec” error message, 354
 - LiveSourceTimeout variable, 395
 - Local file system, 399
 - Log file, 128
 - Log Path
 - variable, 418, 419
 - location of error log, 303
 - process id, 116
 - Log Roll Frequency, 304, 305
 - variable, 411, 412
 - Log Roll Size
 - variable, 412
 - Log Roll Size variable, 411, 412
 - Logging Style
 - format, 292
 - options, 297, 298
 - scalable multicasting and, 207
 - variable, 411
 - LogRollSize variable, 411
 - Logs
 - access control and, 37, 216, 286
 - access log, 283
 - customizing, 297
 - format, 288
 - rolling, 304
 - accesslog.txt, 249
 - authentication and, 37, 226, 286
 - cache log
 - in configuration file, 396
 - cache requests log, 305
 - cached requests log, 305
 - error log, 283
 - format, 303
 - rolling, 304
 - G2SLTA and, 37, 51, 284
 - ISP hosting and, 37, 258, 286
 - Java Monitor and, 286
 - monitoring and, 37
 - multicasting and, 37, 190, 285
 - NT Performance Monitor, 281
 - reglog.txt, 248
 - reporting and, 101
 - SMIL files and, 286
 - splitting and, 37, 284
 - streaming and, 37, 139, 284
 - unicasting and, 37, 146, 284
 - Windows NT Performance Monitor, 281
- M** Master Settings, 102

MaxBandwidth variable, 386, 387
 Maximum Bandwidth, 212, 213, 350
 Maximum Client Connections, 212
 MIME types

- configuring on Web server, 97
- list, 380, 381, 413
- setting up RealServer, 97
- troubleshooting, 337, 351

 MinPlayerProtocol variable, 352, 387

- changing, 433

 Monitor Password variable, 275, 418

- changing, 433

 Monitor Port, 95

- variable, 275, 420

 Monitoring

- ISP hosting and, 37
- RealProxy and, 106
- troubleshooting, 346

 Mount point, 71

- / (forward slash), 72
- /broadcast/, 372
- /admin/, 212
- /broadcast/, 169
- in ISP Hosting list, 402, 404, 407
- in links, 68, 72
- /live/, 147
- main, 140
- multiple, 72, 360
- /ramgen/, 212
- /redundant/, 371, 396, 64
- /scalable/, 203
- variable
 - in G2_Encoders list, 397, 398
 - in ISP hosting, 406
 - in local file system, 399, 400
 - in Pre_G2_Encoders list, 398
 - in RAM_File_Generator list, 421
 - in RealAdministrator_Files list, 423
 - in RealAdminsitrator list, 422
 - in scalable multicast list, 416
- /viewsource/, 212

 Mount variable, 430, 431
 mSQL, 254
 Multicast Delivery Only, 200

- error message, 343

MulticastAddress variable, 427, 428
 Multicasting

- access control and, 37, 189, 215
- access log, 302
- announcing via SAP, 195
- authentication and, 189, 225
- compared to other delivery methods, 31
- described, 181
- firewalls and, 119, 189
- G2SLTA and, 30, 50, 187
- HTTP Postable list, 401
- in access log, 283, 289, 295
- in configuration file
 - back-channel, 414
 - scalable, 415
- ISP hosting and, 257
- Java Monitor and, 275
- license, 38
- links, 198, 204
- live archiving and, 152, 187
- logs and, 190, 285
- minimum settings, 39
- monitoring and, 190
- mount point, 72
- PNA, 183
- RealProxy and, 105, 188
- requiring use of, 213
- RTSP, 183
- SAP information, 413
- scalable, 184
- splitting and, 37, 161, 187, 342
- streaming and, 138, 186
- troubleshooting, 343
- unicasting and, 145, 187
- when to use, 144

N Name variable, 393, 394
 Naming convention one *See* account-based hosting
 Naming convention two *See* dedicated hosting
 Network address translation firewall, 128, 130
 NoArchive variable, 410
 No-Cache Paths, 107

- NoCacheDir list, 397
- NT *See* Windows NT
- NTLMAuthenticator list, 388, 389
- O**
 - ODBC compliance, 253
 - OriginSpec variable, 427
- P**
 - Packet filter firewall, 128, 129, 131
 - Password
 - for content users, 223
 - for encoder users, 33, 223
 - for encoders, 398
 - for Java Monitor, 418
 - for pre-G2 encoders, 147, 228
 - for RealSystem Administrator users, 93, 223, 228
 - in G2SLTA, 52
 - mkpnpass, 242
 - variable, 398
 - in Database list, 393, 394
 - in splitting, 426, 428
 - Password variable, 426
 - Path
 - in ISP hosting user list file, 262, 263, 266, 404, 407
 - in scalable multicasting, 202
 - variable, 393
 - in Databases list, 392, 393, 394
 - in View Source list, 430
 - Path_01 variable, 401
 - in HTTP Postable list, 401
 - in HTTPDeliverable list, 401
 - PathPrefix variable, 425, 427, 429
 - PathToDBPlugin variable, 394
 - PAUSE, in access log, 296
 - Permissions table, 251
 - Pid Path variable, 89, 419, 420
 - changing, 433
 - described, 116
 - “This player doesn't support user authentication” error message, 346
 - “Player Plus only” error message, 352
 - Player validation, 233
 - Player version, 213, 387
 - changing, 433
 - troubleshooting, 352
 - Playlist
 - changing while in use, 59
 - creating, 52
 - “Please download a new RealPlayer” error message, 352
 - “Please make sure you have downloaded the latest RealPlayer” error message, 353
 - Plugin Directory variable, 418, 419
 - changing, 433
 - described, 419
 - G2SLTA and, 61
 - Plugin ID
 - in AuthenticationRealms list, 388, 389
 - options, 389
 - in Databases list, 393
 - options, 393
 - variable, 389, 392, 393
 - PlusOnly variable, 387
 - PNA multicasting
 - defined, 183
 - PNA Port, 95, 196, 361
 - in live broadcasting, 147
 - in on-demand streaming, 140
 - variable, 414, 420
 - PNA protocol, 26, 69, 121
 - “PNA unsupported for requested data type” error message, 353
 - pn-admin, 423
 - pn-broadcast-receiver-fs, 424
 - pn-encoder, 398
 - pn-includer, 423
 - pn-live3, 398
 - pn-local, 400, 404, 421, 422
 - PNM, 70
 - pn-ramgen, 421
 - pn-redundant variable, 396
 - pn-vsrfcsys, 431
 - pn-vsrtaghd1r, 423, 432
 - pn-xmltag, 423, 432
 - Port
 - described, 70
 - for live broadcasts, 146, 147

- in links, 68, 70
 - variable
 - in G2_Encoders list, 397, 398
 - in Pre_G2_Encoders list, 398
 - Port hinting, 113
 - Port Range, 202
 - variable, 416, 417
 - Port_01 variable, 386
 - PortRange variable, 427
 - Ports
 - in multicasting, 202
 - list, 386
 - reserving for multicasting, 192
 - use by RealServer, 133
 - used through firewalls, 124
 - variable, 385
 - ppvbasic.txt
 - defined, 246
 - warning, 246
 - Pre-RealSystem G2 Encoders, 398
 - Process ID, 116
 - ProtectedVirtualPath variable, 390, 391
 - Protocol, 26, 69
 - in links, 70
 - variable, 425, 428
 - Protocol variable, 425, 427
 - Provider, 240
 - variable, 388, 389
 - Pull splitting
 - access log, 302
 - described, 176
 - links, 178, 179
 - See also* splitting
 - PullSplitEnabled variable, 428
 - Push splitting
 - See* splitting
 - PushBufferingTime variable, 428
- R**
- Ram files, 351
 - described, 75
 - troubleshooting, 351
 - Ramgen
 - back-channel multicasting, 375
 - described, 72, 76
 - G2SLTA, 370
 - in configuration file, 420
 - in links to ISP hosted content, 364
 - in links to live content, 65, 148, 153, 154, 367, 368, 371
 - in links to on-demand content, 141, 362
 - in links to pull split content, 374
 - in links to push split content, 169, 372, 373
 - links, 369
 - live archiving, 369
 - mount point, 360, 421
 - in HTTP Delivery list, 400
 - in HTTP delivery list, 212, 401
 - in links, 351
 - scalable multicasting, 360
 - Real Time Streaming Protocol *See* RTSP
 - RealAdministrator_Files list, 423
 - Realm
 - defined, 239
 - variable, 390
 - in AuthenticationRealms list, 388, 389
 - in CommerceRules list, 390
 - in Pre_G2_Encoders list, 398
 - in RealAdministrator_Files list, 421, 424
 - Realm variable, 390
 - RealPix, 69
 - RealPlayer Plus, 213
 - RealPlayer Plus Only, 213
 - RealPlayer version, 213
 - RealProducer Plus, 192, 193
 - RealProxy, 37
 - access control and, 37, 105, 215
 - administrators, 123
 - authentication and, 37, 106, 225
 - communication with RealServer, 127
 - described, 104
 - in configuration file, 396
 - ISP hosting and, 106, 258
 - logs and, 37, 106, 305
 - monitoring and, 106
 - multicasting and, 105, 188
 - restricting, 107
 - splitting and, 105, 161

- streaming and, 138
 - unicasting and, 105, 145
 - RealProxy and firewalls, 119
 - RealSystem Administrator, 92, 396
 - authenticating users, 93
 - in access log, 301
 - starting, 92
 - troubleshooting, 338, 339
 - RealSystem Administrator HTML list, 421
 - RealSystem Broadcast Distribution list, 424
 - RealSystem Broadcast Redundancy list, 396
 - RealSystem Broadcast Redundancy Plugin list, 395
 - RealSystem G2 Encoders list, 398
 - Receivers list, 427
 - RECEIVED, appearance in access log, 296
 - Reconnect variable, 395
 - Recording live broadcasts, 150
 - RECSTART, in access log, 296
 - Redirect directory, 249
 - RedundancySendInterval variable, 424, 425, 433
 - Redundant sources
 - described, 61
 - register_log table, 252
 - Registration Prefix, 233
 - reglog.txt, 246, 248
 - RelayMode variable, 426
 - Reporting, 208
 - cache log, 107
 - Reports *See* logs
 - Resend, 197
 - variable, 414
 - ResendSupported variable, 425, 426, 427, 428
 - Restricting access, 212
 - RESUME, in access log, 296
 - Reuse Address, 202
 - variable, 416
 - rm files
 - bandwidth negotiation, 142
 - rmserver.pid, 116
 - RN5Authenticator list, 389
 - rn-db-flatfile, 392, 393
 - rn-db-mysql, 393
 - rn-db-odbc, 394
 - rn-db-wrapper, 394
 - RTP, 186
 - additional reading, 190
 - ports, 194
 - RTSP, 70, 121
 - RTSP multicasting
 - defined, 183
 - RTSP Port, 95, 196
 - in Access control list, 222
 - in ISP hosting, 95, 361
 - in live broadcasting, 147
 - in on-demand streaming, 140
 - variable, 386, 414, 420
 - in Multicast list, 414
 - use in on-demand streams, 140, 147
- S**
- SAP, 190, 195, 201, 413, 414
 - SAPTransmissionEnabled variable, 428, 433
 - Saving live broadcasts, 150
 - Scalable mount point, 72, 201, 376, 416
 - in HTTP Deliverable list, 212, 401
 - in HTTP Delivery list, 400
 - in links to scalable multicasts, 376
 - Scalable multicasting
 - configuring, 200
 - defined, 182, 184
 - list, 416
 - SDP
 - access log, 302
 - additional reading, 190
 - troubleshooting, 344
 - Secure mount point, 72, 361
 - back-channel multicasting, 375
 - G2SLTA, 370
 - live archiving, 369
 - scalable multicasting, 376
 - streaming, 362
 - unicasting, 367, 368, 371
 - SecureAdmin
 - adding a user, 227
 - SecureContent

- adding a user, 227
- list, 389
- SecureContent list, 390
- SecureEncoder
 - adding a user, 227
- SecureLiveContent list, 391
- SEEKSTART, in access log, 296
- Send Client Statistics, 208
- SendAnnouncementEnabled variable, 414
- SendClientStatistics variable, 416, 417
- SeparatorCharacter variable, 395
- “The Server has reached capacity” error message, 350
- “Server not responding properly
 - Heartbeat check disabled” message, 334
- Session Announcement Protocol *See* SAP
- SessionTimeout variable, 426, 427, 428, 433
- ShiftToUnicast variable, 416, 417
- ShortName variable, 396
 - described, 399
 - in G2_Encoders list, 397, 398
 - in ISP hosting, 406
 - in Pre_G2_Encoders list, 398
 - in Ramgen list, 420
 - in RealAdministrator list, 421
 - in RealAdministrator_Files list, 421, 423
 - in RealContent list, 399
 - in Scalable Multicast list, 416
- SIGHUP command, 116
- Simulated Live Transfer Agent *See* G2SLTA
- Simultaneous content creation *See* live archiving
- SLTA *See* G2SLTA
- SMIL file
 - authentication and, 238
 - defined, 41
 - in access log, 284, 286, 291, 292, 300
 - multicasting and, 192, 415
- SMIL generation, *see* Ad streaming
- SOCKS firewall, 128, 130, 131
- Source Access, 102
- SourceName variable, 425
- Sources list, 416
- Split mount point
 - in links to pull Split content, 374
- Splitting
 - access control and, 37, 162, 215
 - and authentication, 162
 - authentication and, 37, 162, 225
 - compared to other delivery methods, 31
 - described, 159
 - firewalls and, 119, 122, 123, 162
 - G2SLTA and, 30, 50, 161
 - ISP hosting and, 257
 - Java Monitor and, 274, 277, 279, 280
 - license, 38
 - live archiving and, 152, 161
 - logs and, 37, 284, 301
 - minimum settings, 39
 - monitoring and, 37
 - mount point, 72
 - multicasting and, 37, 161, 187
 - RealProxy and, 105, 161
 - See also* pull splitting
 - streaming and, 138, 161
 - troubleshooting, 341
 - unicasting and, 145, 161
 - view source and, 101
- Starting RealServer, 85
- Stat1
 - location in access log, 288
 - syntax, 293
- Stat2
 - location in access log, 288
 - syntax, 294
- Stat3
 - location in access log, 288
 - syntax, 295
- Stateful packet filtering firewall, 128, 129, 131
- Statistics, 201
 - collecting in access log, 283
 - displaying in Java Monitor, 273
 - scalable multicasting, 206
 - See also* logs
- Statistics type 1
 - gathering with StatsMask, 299
 - syntax, 293
- Statistics type 2

- gathering with StatsMask, 299
 - syntax, 294
 - Statistics type 3
 - gathering with StatsMask, 299
 - syntax, 295
 - Stats Mask
 - default value, 287
 - options, 293, 298
 - RealPlayer and, 299
 - variable, 5, 288, 411
 - STOP, in access log, 88, 296
 - Stopping RealServer
 - UNIX, 89
 - Windows NT, 88
 - Streaming, 25
 - access control and, 37, 138, 215
 - ad streaming and, 37
 - authentication and, 37, 138, 224
 - firewalls and, 37, 119, 138
 - G2SLTA and, 50
 - in access log, 300
 - ISP hosting and, 37
 - Java Monitor and, 139
 - live archiving and, 138, 152
 - logs and, 37, 139, 284
 - monitoring and, 37
 - multicasting and, 186
 - RealProxy and, 37, 138
 - splitting and, 138, 161
 - troubleshooting, 339
 - unicasting and, 138, 144
 - Support *See* Technical Support
 - SupportPluginDirectory
 - variable, 419
 - SupportPluginDirectory variable, 419
 - changing, 433
 - G2SLTA and, 61
 - SureStream
 - in G2SLTA playlists, 53
 - multicasting, 183, 185, 186
 - RTSP, 69, 121
 - streaming, 141
- T**
- TableNamePrefix variable
 - in Database list, 393, 394
 - Tables
 - access_log, 250, 253
 - permissions, 250, 251
 - redirect, 250, 252
 - register_log, 250, 252
 - users, 250
 - TAC
 - in playlist, 58, 59
 - TagHandlers list, 423
 - Target Directory
 - variable, 409
 - Technical support, 7, 355
 - See also individual troubleshooting topics*
 - “This server is configured to support only multicast connections...” error message, 343
 - Time to Live, 197, 202
 - Timeout
 - in scalable multicasting, 202
 - variable
 - in scalable multicast, 416
 - in scalable multicast list, 417
 - Title, author, and copyright information *See* TAC
 - To variable, 385
 - TranslationMounts list, 402, 405
 - in ISP hosting, 407
 - Transparent proxy firewall, 128, 129, 131
 - Transport variable, 385
 - Troubleshooting, 333
 - access control, 345
 - ad streaming, 346
 - archiving, 341
 - authentication, 345
 - G2SLTA, 341
 - license issues, 351
 - monitoring, 346
 - multicasting, 343
 - RealSystem Administrator, 338
 - SMIL files, 348
 - splitting, 341
 - streaming, 339
 - unicasting, 340
 - ts.log file, 106
 - TSEnable variable, 396, 397

TSLog variable, 397
TSLogPath variable, 397
TSPort variable, 397
TTL variable, 197, 414, 416, 417, 425, 428
Type variable, 426, 428

U UnICASTING
 access control and, 37, 145, 215
 access log, 301
 authentication and, 37, 145, 225
 compared to other delivery methods, 31
 firewalls and, 37, 119, 145
 G2SLTA and, 37, 50, 145
 ISP hosting and, 146
 live archiving and, 37, 144
 logs and, 37, 146, 284
 monitoring and, 37
 multicasting and, 145, 187
 RealProxy and, 105, 145
 scalable multicasting and, 37
 splitting and, 145
 streaming and, 138, 144, 161
 switching from multicasts, 201
 troubleshooting, 340

UNIX
 Group variable, 115
 PID, 116
 SIGHUP, 116
 special features, 115
 starting RealServer, 88
 stopping RealServer, 89
 user name, 334

UseGUIDValidation variable, 390, 391

User authentication, 227
 administrators, 228
 content, 228
 encoders, 227

User List
 file, 262
 in dedicated hosting, 266
 using multiple files, 265
 variable
 in 5.0, 267

User list file
 example, 405

User Name, 116
User Path
 in ISPHosting list, 402
User variable, 115
 in Database list, 393, 394
 UNIX user name, 430
UserLists list, 402
UserPath variable, 402
UseTCPForPullBackchannel variable, 427, 429, 433

V ValidPlayerOnly variable, 386, 387
Variable tag, 380
Version number, 357
View source, 99
 access control and, 101
 G2SLTA and, 101
 live archiving and, 101
 logs and, 101
 mount point, 401, 432
 SMIL files, 99
 splitting and, 101
 streaming and, 101
ViewSourceLongName variable, 430
Virtual path, 46
 described, 47
 variable, 416, 417
Vsrcfsys mount point, 432

W Web server
 firewalls and, 125
 log format, 288
 MIME types on, 97
 RealServer and, 71, 95, 112

Web Server Address
 variable, 416, 417

Web Server Address or IP Address, 207, 208

Web Server CGI Path, 208
 variable, 416, 418

Web Server Port, 207, 208
 variable, 416, 418

Windows NT
 Performance Monitor, 115, 280
 running multiple RealServers, 87

- services, 85, 334
- special features, 114
- starting RealServer, 85
- stopping RealServer, 88

X XML

- configuration file, 94, 379
- declaration tag, 379
- license files, 38

Y

- “You cannot receive this content...” error message, 353
- “You have connected to a RealMedia Server” error message, 352
- “You need to obtain a new player to play this clip...” error message, 352
- “Your account has been locked...” error message, 345

