



REALSERVER ADMINISTRATION GUIDE
RealSystem G2



Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. 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.

©RealNetworks, Inc.

RealAudio, RealVideo, and RealPlayer are registered trademarks of RealNetworks, Inc.

The Real logo, RealServer, RealPlayer Plus, RealText, RealPix, RealAudio Encoder, RealVideo Encoder, RealEncoder, RealPublisher, RealProducer, RealProducer Plus, RealProducer Pro, SureStream, RealBroadcast Network, and RealSystem are trademarks of RealNetworks, Inc.

RealFlash is a trademark of Macromedia, Inc. and RealNetworks, Inc.

Macromedia is a registered trademark and Flash and Shockwave are trademarks of Macromedia, Inc.

STiNG is a trademark of Iterated Systems, Inc.

ACELP-NET codec used under license from Université de Sherbrooke. Sipro Lab Télécom, Inc. Copyright ©1994-1997. All rights reserved.

DolbyNet is a trademark of Dolby Laboratories, Inc.

Dolby Digital AC-3 audio system manufactured under license from Dolby Laboratories.

Apple, Macintosh, and Power Macintosh are registered trademarks of Apple Computer, Inc.

Microsoft, MS-DOS, Windows, and Windows NT are registered trademarks and ActiveX is a trademark of Microsoft Corporation.

Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation.

Pentium is a registered trademark and MMX and the Intel Optimizer Logo are trademarks of Intel Corporation.

Sonic Foundry and Sound Forge are registered trademarks of Sonic Foundry, Inc.

Other product and corporate names may be trademarks or registered trademarks of other companies. They are used for explanation only, with no intent to infringe.

RealNetworks, Inc.
1111 Third Avenue, Suite 2900
Seattle, WA 98101 USA

<http://www.real.com>



CONTENTS

	INTRODUCTION	1
	Overview	1
	How This Manual Is Organized	1
	Conventions in This Manual	3
	Available Features	4
	Additional RealSystem Resources	5
	Technical Support	5
1	WHAT'S NEW IN REALSERVER G2?	7
	New Features	7
	Compatibility With Previous Releases	10
2	OVERVIEW	11
	How RealServer Works	11
	Communication Between RealServer and RealPlayer	12
	SMIL Files	13
	Linking to RealSystem Content	13
	Protocols	14
	Port Settings	15
	Mount Points	16
	Ram Files and Ramgen	18
	Virtual Paths	19
	Storing Files and Presentations	19
	Linking to Files and Presentations	20
	Links from Web Pages to Streamed Media Clips	20
	Linking from SMIL or Ram Files to Media Clips	22
	Linking from RealPlayer	23
	Working with Content Creators	24
3	STARTING AND STOPPING REALSERVER	25
	Windows NT	25
	Starting RealServer Under Windows NT	25
	Stopping RealServer Under Windows NT	28
	UNIX	28
	Starting RealServer Under UNIX	29

	Stopping RealServer Under UNIX	30
	Configuring MIME Types	30
	License Information	32
4	CUSTOMIZING REALSERVER FEATURES	35
	Customizing RealServer	35
	Configuration File	37
	Editing the Configuration File with a Text Editor	37
	Common Settings	39
	Port Variables	39
	Mount Points	40
5	ADVANCED FEATURES	43
	RealServer Caching Features	43
	Making Your Content Cacheable.....	44
	Reserving IP Addresses for RealServer's Use.....	46
	Running Web Servers and RealServer on the Same System	47
	Firewalls and RealServer	47
	Communicating with Clients Behind Firewalls	49
	Locating RealServer Near the Firewall.....	49
	Features Specific to the Operating System	50
	Windows NT	50
	UNIX.....	50
6	STREAMING PRESENTATIONS ON DEMAND	53
	Overview	53
	Streaming On-Demand Clips.....	54
	Working with SureStream Files.....	55
7	BROADCASTING PRESENTATIONS	57
	Overview	57
	Live Broadcasting.....	57
	Archiving Live Files	61
	Simulating a Live Broadcast.....	64
8	SPLITTING AND MULTICASTING	67
	Overview	67
	Splitting	67
	RealServer Splitting Methods	69
	Multicasting	77
	RealServer Multicasting Methods	78
	Logging Multicasts	81
	Setting Up Multicasting.....	82

	Combining Splitting and Multicasting	88
9	LIMITING ACCESS TO REALSERVER	89
	Overview	89
	Controlling Access to HTTP Streams	90
	Limiting Access by Number of Connections or Bandwidth	90
	Limiting Access by RealPlayer Version	92
	Limiting Access to Back-Channel Multicast Reception	92
	Limiting Access Via IP Address	93
10	AUTHENTICATING REALSERVER VISITORS	95
	Overview	95
	Authentication Components	97
	Realms	97
	Databases	101
	Secure Virtual Paths	103
	Encoder User Authentication	104
	RealSystem Administrator User Authentication	104
	Content User Authentication	105
	Creating Secure Virtual Paths	108
	Allowing Users to Self-Register	109
	Linking to Authenticated Content	110
	Combining Authentication with Other RealServer Features	111
11	STORING AUTHENTICATION DATA	113
	Overview	113
	RealServer Data Storage	113
	Using Text Files	114
	Using a Database	118
	Setting Up Other Types of Data Storage	121
12	MONITORING ACTIVITY	123
	G2 Java Monitor	123
	Using G2 Java Monitor	124
	Configuring G2 Java Monitor Settings	128
	Using Windows NT Performance Monitor	128
13	REPORTING	131
	Access Log	131
	Reading an Access Log	131
	Customizing Information Reported by the Access Log	140
	Log File Rolling	143
	Disabling Log File Rolling	144

	Error Log	144
	Cached Requests Log	145
A	CONFIGURATION FILE SYNTAX	147
	Configuration File Components	147
	XML Declaration Tag	147
	Comment Tags	147
	List Tags	148
	Variable Tags	148
B	CONFIGURATION FILE CONTENTS	151
	Editing the Configuration File	151
	Elements of the Configuration File	151
	Ports	152
	Paths	152
	Passwords	153
	MIME Types	153
	Caching	154
	IP Binding	155
	Live Archiving	156
	Allowance	157
	HTTP Support	158
	Access Control	159
	File Systems	160
	Ramgen	163
	Encoders	163
	Splitting	165
	Multicasting	167
	Authentication and Commerce	170
	Logging	175
C	CONFIGURATION FILE EQUIVALENTS	177
	INDEX	181



INTRODUCTION

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

Overview

This manual is aimed at the information services administrator who will be setting up RealServer but is not necessarily creating content. Instructions for authoring media are in the separate *RealSystem™ G2 Production Guide*.

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

How This Manual Is Organized

This manual contains the following chapters:

Chapter 1: What's New in RealServer G2?

If you're familiar with previous versions of RealSystem, this chapter will give you a quick update on the many aspects of streaming that have changed in RealSystem G2.

Chapter 2: Overview

This chapter gives the “big picture” of how RealServer works with a Web server to stream media to a player.

Chapter 3: Starting and Stopping RealServer

This is a guide to starting and stopping RealServer. Depending on which platform your RealServer runs on, different automatic starting options are available. The license structure and MIME types are also discussed.

Chapter 4: 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 5: Advanced Features

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

Chapter 6: Streaming Presentations On Demand

In this chapter, instructions are given for delivering pre-recorded or prepared clips.

Chapter 7: Broadcasting Presentations

Live clips are streamed much like static clips, with a few differences. Learn how to make broadcasting work well.

Chapter 8: Splitting and Multicasting

Whether you are streaming over an intranet or the Internet, splitting and multicasting can help you make the best use of bandwidth and can provide highest-quality reception.

Chapter 9: Limiting Access to RealServer

You can limit access to RealServer by specifying restrictions such as maximum bandwidth and IP addresses.

Chapter 10: Authenticating RealServer Visitors

Control and limit who can view your content; this chapter describes the different RealServer authentication methods and the advantages of each.

Chapter 11: Storing Authentication Data

RealServer comes with some different methods for tracking authentication information. Use such data for billing or to track who's watching what.

Chapter 12: Monitoring Activity

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

Chapter 13: Reporting

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. Errors are reported in their own log, which can help you troubleshoot any problems that arise.

Appendixes**Appendix A: Configuration File Syntax**

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

Appendix B: Configuration File Contents

This is a guide to the configuration file contents, for those who prefer to edit it directly rather than using RealSystem Administrator.

Appendix C: Configuration File Equivalents

For those RealServer administrators who've worked with a previous version of RealServer, this chapter lists settings in the old configuration file along with their new XML-based equivalents.

Conventions in This Manual

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

RealSystem clients, such as RealPlayer, are referred to generically as “clients”. Where information applies specifically to the RealNetworks RealPlayer® or RealPlayer Plus™, this is spelled out. Although most clients in use are RealNetworks' own RealPlayer, RealNetworks also makes a software development kit that enables other companies to develop their own players which can also receive streamed data types.

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

“Clips,” “content,” “media files,” and “files” 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>	Syntax of configuration files, URLs, or command-line instructions are given in this typeface.
<i>value</i>	Placeholder words are given in an italic monospaced typeface. Substitute the appropriate value for your system.
...	Ellipses indicate nonessential information omitted from the example.
[]	Square brackets indicate optional material. If you choose to use the material within the brackets, do not type the brackets themselves. An exception to this is in the access log, where statistics generated by the StatsMask variable are enclosed within actual brackets.

Examples of URLs that point to the RealServer are given like this:

realserver.company.com

where:

realserver is meant to be the machine name of the computer that is running your RealServer. Substitute the name of your organization's computer where you see this text.

company.com is meant to be an example of a domain name. Substitute the domain name of your organization's machines where you see this text.

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 on your RealServer. If you would like to add to your RealServer's capabilities, contact RealNetworks or your reseller.

Additional Information

Instructions on reading license files with RealSystem Administrator are given in "License Information" on page 32.

Additional RealSystem Resources

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

- *RealSystem G2 Production Guide*

This manual explains the basics of creating streaming files with the RealSystem tools. It tells how to calculate bandwidth needs and shows how to put a multimedia presentation together.

- *Embedded RealPlayer Extended Functionality Guide*

This guide supplements *RealSystem G2 Production Guide*. It explains how to use JavaScript or VBScript to control RealPlayer functions for a presentation embedded in a Web page.

- *RealText™ Authoring Guide*

This manual explains how to create streaming text. You can use RealText, for example, to create a live stock ticker feed or provide video subtitles.

- *RealPix™ Authoring Guide*

With RealPix you can create streaming slide shows of still images. *RealPix Authoring Guide* tells you how to put a slide show together and use special effects such as fades and zooms.

- RealSystem G2 Software Development Kit (SDK)

RealNetworks has developed a Software Development Kit (SDK) that lets you integrate applications with RealSystem or create new plug-ins for RealServer or RealPlayer. Knowledge of programming is required to use the SDK. Register for and download the SDK from

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

Technical Support

For technical support with RealSystem G2, 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>

WHAT'S NEW IN REALSERVER G2?

RealServer G2 is designed on a new architecture that allows greater extensibility and interoperability with third-party solutions.

New Features

In addition to the improvements added to RealSystem G2, RealServer G2 includes many new features.

License Files

Previous versions of RealServer used an encrypted license key string, which was used during installation and placed in the configuration file, to indicate which features were available.

RealServer G2 introduces new license files that allow for greater flexibility in upgrading available features.

Additional Information

See “License Information” on page 32.

RealSystem Administrator

RealServer G2 includes RealSystem Administrator, a new HTML interface for working with nearly every aspect of Server operations. Use this tool to fine-tune RealServer features, monitor Server activity, and play sample presentations.

RealSystem Administrator can be accessed from any Web browser on the network. Security features for RealSystem Administrator are also included.

Additional Information

See “Customizing RealServer” on page 35.

New Protocols

RealServer G2 now uses RealTime Streaming Protocol (RTSP) as its control protocol and RealNetworks' proprietary RDT as its packet protocol.

New URL Format

A visible change in this version of RealServer is the change in URLs that point to RealServer presentations.

- URLs that point to G2 presentations begin with `rtsp://`.
- Included in the URLs are mount points and virtual directories. These tell RealServer which file system to use in processing the presentation request.

Additional Information

See Chapter 2: Overview.

New Configuration File Format

The configuration file, which stores all the settings used by the RealServer, is now in Extensible Markup Language (XML) format. This new format allows greater flexibility and extensibility by third parties. The file is easy to modify with the new RealSystem Administrator, or you can still use a text editor to make changes.

Additional Information

See Chapter 4: Customizing RealServer Features.

Open Architecture

Most features are handled by separate files, called plug-ins. Located in the plug-ins directory, these plug-ins are read by RealServer when it starts and they control what happens to client requests. The RealServer open architecture allows third-party companies to develop plug-ins that can be added easily to RealServer for future functionality. This open architecture lends itself to modularity and customizability.

Additional Information

See Chapter 4: Customizing RealServer Features.

New Splitting Method

In addition to the splitting method of previous versions, splitters can now rebroadcast material upon request.

Additional Information

See “Pull Splitting” on page 69.

Control Access to Ports Based on IP Address

RealServer G2 allows you to limit access to RealServer based on the IP address of the requesting client just as earlier versions did, but RealServer G2 adds the ability to restrict access to certain ports. In this way, you can better control traffic flow on your RealServer computer.

Additional Information

See “Limiting Access Via IP Address” on page 93.

Authentication

New options for verifying the identity of visitors to your RealServer presentations include Windows NT authentication.

Additional Information

See Chapter 10: Authenticating RealServer Visitors.

New Monitoring Methods

Use the constantly updating G2 Java Monitor in RealSystem Administrator. Zoom in for a closer look. Change the colors of the display.

Additional Information

See “G2 Java Monitor” on page 123.

If you have Windows NT, use the NT Performance Monitor with the RealServer `rmserver.pmc` file.

Additional Information

See “Using Windows NT Performance Monitor” on page 128.

Integration with Windows NT User Groups

RealServer works with Windows NT User Authentication to give access to users who are already in the NT user group lists.

Additional Information

See “Windows NTLM Challenge/Response” on page 99.

Compatibility With Previous Releases

RealServer G2 is fully compatible with RealServer 3.0 through 5.0:

- Presentations created with earlier versions of RealSystem tools still work seamlessly with RealServer G2.

To use new features, such as RealText, in an existing presentation, you must update the presentation by creating a SMIL file and modifying the URL that refers to the presentation.

Chapter 2

OVERVIEW

Welcome to RealServer, the streaming media solution! RealServer streams virtually any datatype, including audio, video, and images. RealServer allows you to grow as your needs and use expand. This chapter shows how you can put RealServer to work for you.

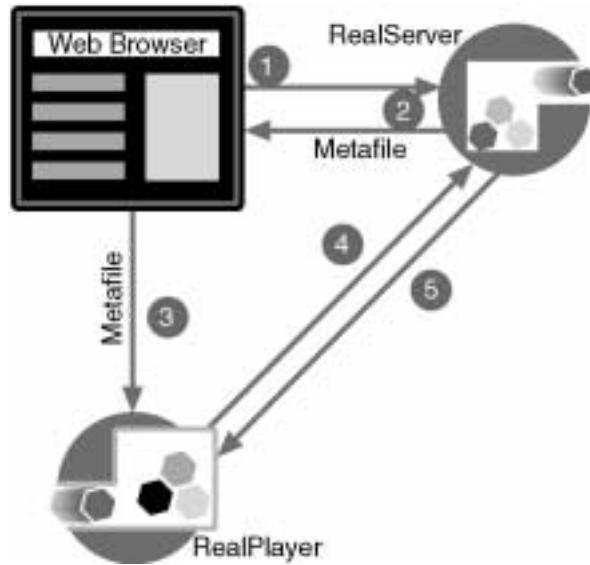
How RealServer Works

RealServer streams both live and on-demand material, through unicasting or multicasting. It works with Web servers to stream to clients over networks and the Internet. Some RealServer features can interact with third-party products to create specialized functions.

This guide is intended for the technical system administrator who will manage RealServer and its activities, but not necessarily create the material to be streamed. Information on creating content is available in a companion book, *RealSystem G2 Production Guide*.

IS professionals, server administrators, Web masters and others providing Web pages for the Internet and intranet may also find this document useful.

The following diagram shows an overview how RealSystem components work together.

Overview

1. A visitor browses a Web page and clicks a link to a streaming media presentation served by RealServer.
2. RealServer creates a small metafile and sends it to the visitor's Web browser.
3. The browser downloads the metafile and sends it to the visitor's RealPlayer. The metafile, called a Ram file, contains the address (or addresses) of the media presentation mentioned in the link.
4. RealPlayer reads the link in the metafile and requests the presentation directly from RealServer.
5. RealServer streams the files in the presentation to the RealPlayer. Finally, RealPlayer plays the presentation.

Communication Between RealServer and RealPlayer

As the user clicks a link that points to a RealServer presentation, RealPlayer opens a two-way connection with RealServer. This connection uses TCP to send information back and forth between RealPlayer and RealServer.

Initial TCP Control Connection



Once RealServer approves the request, it sends the requested clip along a one-way UDP channel.

UDP Data Connection



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

SMIL Files

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. For detailed information on creating SMIL files, see *RealSystem G2 Production Guide*.

Linking to RealSystem Content

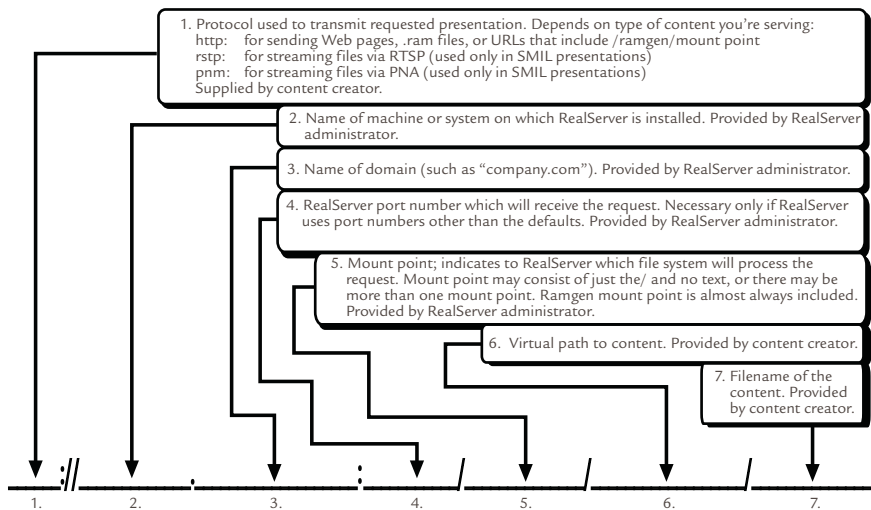
A visitor to a Web site clicks links that point to content served by RealServer. The way these links are constructed tells RealServer how to stream them. To understand how to construct a link to RealServer content, you must first learn

about key RealServer features such as ports, mount points, and base paths. You will need to give this information to the content creators who will be using your RealServer. They won't need to know the details of RealServer operation, just what information to include in their links.

Once you have read about ports, mount points, and base paths, you'll see how to create links, described in "Linking to Files and Presentations" on page 20.

This illustration shows the parts of a URL that points to material served by RealServer. More detailed explanations follow this diagram.

Parts of a Link



For example, the following link to a RealVideo file would appear in a Web page:

```
http://realserver.company.com:8080/Concerts/French/debussy.rm
```

The key to understanding where to put media files and how to reference them—and therefore how to take full advantage of RealSystem's power—are ports, mount points, file systems, base paths, and virtual paths.

Protocols

RealServer uses two main 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 RealServer sends information about

the clips' titles to the client. Authentication demands from RealServer and passwords supplied by the user are also sent along this connection.

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 with their multiple bandwidth encoding. RealPix also requires RTSP.

PNA is the proprietary client-server protocol designed and used by RealNetworks in RealSystem versions 5.0 and earlier. The ability to serve via PNA is supported in RealServer G2 for compatibility with older versions of RealPlayer.

In both RTSP and PNM, media clips are streamed over a one-way UDP channel which is separate from the TCP channel.

Port Settings

Port settings tell RealServer where it should listen for RTSP, PNA, and HTTP requests. These settings are implied in the URL that points to the content and are assumed by RealPlayer to have certain values. When RealPlayer requests an URL that begins with `rtsp://`, it sends the request to the RealServer's port 554. RealPlayer directs an URL that begins with `pnm://` to port 7070. Requests beginning with `http://` are first sent to port 80, and if no response is received, they are redirected to port 8080.

Additional Information

RTSP and PNA are described in "Protocols" on page 14.

Using Different Port Numbers

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 47 for additional information.

You can use alternate port numbers if multiple RealServers are using the same IP address, or if you want to segregate requests for different material. If you do use alternate port numbers, be sure to include the new numbers in the links.

Warning

If you change the RealServer port settings, you will have to update URLs to reflect the new, non-default values. If RealPlayer attempts to play a clip for which the port information is incorrect, it may try to request the

information via HTTP (a much less efficient delivery method.)

Mount Points

A mount point reference appears in every URL. It tells RealServer which plug-in to use when processing the URL request.

When RealServer receives a request for a clip, it examines the URL of the requested clip. RealServer looks through its configuration file for a plug-in whose mount point variable matches the first string after the domain name. It then gives the request to the plug-in named by that list. Most on-demand presentations are handled by the local file system plug-in; the G2 encoder plug-in handles live streams.

To specify a mount point that points to the RealServer main content location, use a forward slash mark (“/”).

Changing Mount Points

If you change the name of a mount point, remember to update the URLs with the new mount point name.

Multiple Mount Points in One Link

In some cases, a link will include more than one mount point. The Ramgen file system is sometimes used in addition to mount points.

Using different mount points that point to the same base path or use 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.

Base Paths

While the mount point identifies a virtual directory, the base path gives the actual directory where the clips are stored. The base path identifies the “root directory” for the URLs that point to clips; it can even point to a completely different physical drive. The base path is similar to an alias.

Consider the following directory structure:

```
(RealServer main directory)
    Content
        Speeches
        Concerts
            French
            Liveconcerts
```

If the main mount point is / and the base path is C:\Program Files\RealServer\Content (Windows) or /RealServer/Content (UNIX), a SMIL file link for a clip in the main Content directory would look like this:

```
rtsp://realserver.company.com/intro.smil
```

A file in the Speeches subdirectory of the Content directory would appear this way in the URL:

```
rtsp://realserver.company.com/Speeches/keynote.smil
```

The Content directory is not mentioned in the URL, because it is already implied in the main mount point's base path.

Using Mount Point and Base Path Together

There are two advantages to using mount point and base path:

- If you relocate content, you need change only the base path to reflect the content's new location, and the URLs can remain the same.
- If content providers' presentations are stored in obscure directories, a mount point can be a brief and sensible name for the content provider to use.

Actual Directories

A link to a file named `debussy.rm` in the French subdirectory of Concerts would look like `rtsp://realserver.company.com/Concerts/French/debussy.rm`. In this case, the portion of the URL corresponding to a mount point is merely the forward slash after the domain name; the rest matches a directory structure. The directory structure is relative to the base path of the mount point, which in this case is identical to the actual directory structure.

If a mount point named `/education/` was added, and its base path was Content (the same base path as the main mount point), a file named `lesson1.rm` would be referenced as `rtsp://realserver.company.com/education/lesson1.rm`. In this case, no "education" directory actually exists; the `lesson1.rm` file can be found in the main content directory.

Virtual Directories

There are two types of virtual directories.

A virtual directory is a combination of a mount point and the actual directories below it.

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

(RealServer main directory)

```

Content
  Speeches
  Concerts
    French
    Liveconcerts

```

In the url `rtsp://realserver.company.com/Concerts/French/debussy.rm`, we refer to `/Concerts/French` as a virtual directory. The actual `French` directory is located at the end of a long path.

In the case of live files, the virtual directory can be the location typed in the production tool. It may look like a directory, but not correspond to any actual directories. For example, if a content creator indicated that live files should be encoded to `Speeches/Famous/Lincoln.ra`, the link to that live file would look like `rtsp://realserver.company.com/Speeches/Famous/Lincoln.ra`. The `Speeches` directory is an actual directory in this case, but it has no `Famous` subdirectory. The virtual directory is `Speeches/Famous`.

Identifying Mount Points and Virtual Directories

You can't determine which parts of an URL refer to mount points and which parts refer to virtual directories just by looking at the URL; you must examine the mount points pages or the configuration file to see which parts of the URL are mount points.

Using Several Mount Points in One URL

A single link may include more than one mount point, in addition to virtual directories. The `Ramgen` mount point is frequently combined with other mount points.

Ram Files and Ramgen

A `Ram` file is a small text file, also known as a metafile, that lists presentations in sequence. `SMIL` files can do sophisticated presentations, but `Ram` files can

provide a quick way of sequencing clips. Ram files have the extension .ram or .rpm.

The ram file generator, known as Ramgen, sends a temporary file to the RealPlayer. This temporary file contains the address of the presentation given in the URL. The Ramgen component is necessary because 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.

If you are using Ram files and are storing them on RealServer (rather than on the Web server), be sure to add the virtual path to the HTTP Deliverable list. “Controlling Access to HTTP Streams” on page 90.

Additional Information

See *RealSystem G2 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 G2 Production Guide*.

Virtual Paths

This manual makes a distinction between virtual directories and virtual paths. A virtual directory is part of a physical directory; a virtual path is a combination of a mount point and a virtual directory. The notion of a virtual path is used in the authentication feature.

Storing Files and Presentations

The most straightforward location for content files is in the base path directory of the main mount point. Links to these files will be short.

If you have many files however, 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 match the lengthy paths, will shorten the links.

Linking to Files and Presentations

Links to media files streamed by RealServer can appear in four places, and use different protocols, as shown in the following table:

Links in RealSystem Files		
This location...	...links to this	Uses protocol
Web page	SMIL file or individual clip	http
Web page	Ram file	http
SMIL files	individual file or files	rtsp or pnm
Ram files	individual file or files	rtsp or pnm
The Open Location dialog box of RealPlayer	individual file	rtsp or pnm

Web pages require a slightly different link format than the other three venues. SMIL files, Ram files, and RealPlayer all use the same format. SMIL files can include additional, sophisticated instructions. Some additional options are available in Ram files. Detailed information on the additional options can be found in *RealSystem G2 Production Guide*.

For examples of the different types of links, as well as the features of SMIL files, examine the content in the Demo subdirectory of the RealServer Content directory. You can view the demonstrations by clicking **Samples** in the left-hand frame of RealSystem Administrator, and then clicking one of the SMIL demonstration links.

Links from Web Pages to Streamed Media Clips

Within a Web page you can link to an individual clip, a .ram or .rpm file, or a SMIL presentation.

Linking a Web Page to a SMIL File or Individual Clip

A link in a Web page to an individual RealServer file uses the following format:

```
http://realserver.company.com:HTTPPort//ramgen/MountPoint/virtual_directory
/filename
```

RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used for streaming. Always use <code>http</code> in Web pages.
<code>realserver.company.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 Variables” on page 39.
<code>ramgen</code>	Ram file generator. Omit this only if you are playing clips locally or if this is a live, push splitting, or multicast stream.
<code>MountPoint</code>	If the clip is in the main directory, you can omit the mount point. (In this case, a single forward slash is considered the mount point.)
<code>virtual_directory</code>	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>virtual_directory</code> .
<code>filename</code>	The file name itself, including the extension. Scalable multicasting adds <code>.sdp</code> after the extension.

Note

Push splitting and pull splitting use a different link format. See “To create the link for push splitting within a SMIL file or Ram file:” on page 74 and “To create the link for pull splitting:” on page 76.

Linking a Web Page to a Ram File

A link to a Ram or Rpm file, which you must create and then store on RealServer, has the following format:

```
http://realserver.company.com:HTTPPort/MountPoint/virtual_directory/
filename.ram
```

RealServer URL Components

Component	Meaning
http	The protocol used for streaming. Because you are adding this link to a Web page, the protocol is HTTP.
<i>realserver.company.com</i>	Machine and domain name of RealServer. IP address may be substituted.
<i>HTTPPort</i>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Variables” on page 39.
<i>MountPoint</i>	If the clip is in the main directory, you can omit the mount point. (In this case, a single forward slash is considered the mount point.)
<i>virtual_directory</i>	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 <i>virtual_directory</i> .
<i>filename.ram</i> or <i>filename.rpm</i>	The name of the .ram or .rpm file.

Unlike the link to the single file, the link to a Ram file does not include the /ramgen/ mount point.

Note

Ram and Rpm files can also be stored on the Web server; adjust the link accordingly.

Linking from SMIL or Ram Files to Media Clips

The content creator will be making SMIL files, which will contain references to streamed media files served by RealServer. Links to streamed media that appear in SMIL files have the following format:

```
rtsp://realserver.company.com:RTSPPort/MountPoint/virtual_directory/filename
```

The following table explains the components in the example above. As the RealServer administrator, you will need to supply content creators with the

RealServer address, RTSP port, and mount points. You may also need to supply information about virtual directories.

RealServer URL Components	
Component	Meaning
<i>rtsp</i>	The protocol used for streaming. In SMIL files, this is usually RTSP. Occasionally will be PNM.
<i>realserver.company.com</i>	Machine and domain name of RealServer. IP address may be substituted.
<i>RTSPPort</i>	Port number where RealServer listens for requests sent via RTSP. This value is usually 554; see “Port Variables” on page 39. If you used PNM for the protocol, substitute the value of PNAPort for RTSPPort here.
<i>MountPoint</i>	If the clip is in the main directory, you can omit the mount point. (In this case, a single forward slash is considered the mount point.)
<i>virtual_directory</i>	The virtual directory is any actual directory, as relative to the base path of the mount point. If the file is located in the base path itself, omit <i>virtual_directory</i> .
<i>filename</i>	The file name itself.

Note

Pull Splitting uses a different link format. See “To create the link for pull splitting:” on page 76.

Additional Information

For detailed information on SMIL files and their syntax, see *RealSystem G2 Production Guide*.

Remember that the Web page links to the SMIL file, and it has a slightly different format. See “Links from Web Pages to Streamed Media Clips” on page 20.

Linking from RealPlayer

The **Open Location** dialog box of RealPlayer uses the same syntax as SMIL files. Links begin with *rtsp://*. See “Linking from SMIL or Ram Files to Media Clips” on page 22.

Working with Content Creators

Unless you are creating media files and SMIL presentations as well as administering RealServer, you will need to give certain information to content creators so that they can create the proper links in their SMIL files and Web pages. If the content providers are encoding live material, they will need to know where to direct their live data.

On-Demand Content

Content creators will need the following information:

- where to place their files
- the RealServer address
- port numbers that match the protocol
- whether Ramgen is in use.

Live Broadcasts and Multicasts

In order to encode a live stream to RealServer, content creators need to know this information:

- the RealServer address
- which port number to connect to
- authentication information such as passwords (if any)
- the URL to use in Web pages that point to a live broadcast
- the URL to use in a SMIL file

STARTING AND STOPPING REALSERVER

Chapter 3

This chapter gives information on starting and stopping RealServer on both Windows NT and UNIX platforms, setting up MIME types, and explains the RealServer license method.

Windows NT

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

Starting RealServer Under Windows NT

RealServer can be started manually or as a service. You can configure each service to use different configuration files.

Whether you start RealServer manually or as a service, if you start it without including a configuration file, RealServer uses the most recently used configuration settings.

Starting RealServer Manually

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

► **To start RealServer from the Start menu:**

On the **Start** menu, click **Programs**, then click **Real**, and finally click **RealServer G2**. This starts the `rmserver.exe` program. If this is the first time you have run RealServer, it loads the default configuration file.

Additional Information

The configuration file is described in Chapter 4:
Customizing RealServer Features.

► **To start RealServer from a command line:**

Move to the RealServer Bin directory and type the following at a command line:

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

To limit the amount of memory that RealServer G2 uses, start RealServer with the `-m` parameter:

```
rmserver ..\rmserver.cfg -m 32
```

where the number after `-m` can be any amount of memory in megabytes, 32 or greater. Each megabyte of RealServer memory accomodates 3 to 4 simultaneous connected users. To allow 200 users to connect, specify 50 megabytes of memory instead of 32. (This parameter is optional.)

Setting Up RealServer as a Service

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.

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 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. 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\6.0\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, such as the `-m` switch, 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.

► 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.

Running Multiple Servers on One System

You can have configuration files with different names for different configurations of a single RealServer, or use different names for different RealServer installations.

You can load configuration files into separate registry keys. Then, run RealServer as a service, one for each configuration file you loaded.

► **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”.
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\6.0\Key.
```

In the example from Step 2 of “Setting Up RealServer as a Service”, in which the configuration settings are loaded into the “Server1” key, the full key name would be HKEY_CLASSES_ROOT\Software\RealNetworks\RealMedia Server\6.0\Server1.

Stopping RealServer Under Windows NT

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

If RealServer was started as a service, stop RealServer through the Services control panel.

UNIX

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

Starting RealServer Under UNIX

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

► **To start RealServer under UNIX:**

Run the `rmserver` program. It is located in the `bin` subdirectory of the RealServer directory, and the configuration file (`rmserver.cfg`) is located in the main RealServer directory.

Move to the `bin` directory and type the following:

```
rmserver ../rmserver.cfg
```

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

You can run RealServer in the background by typing the following from the `bin` directory:

```
rmserver ../rmserver.cfg &
```

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

To limit the amount of memory that RealServer G2 uses, start RealServer with the `-m` parameter:

```
rmserver ../rmserver.cfg -m 32
```

where the number after `-m` can be any amount of memory in megabytes, 32 or greater. Each megabyte of RealServer memory accomodates 3 to 4 simultaneous connected users. To allow 200 users to connect, specify 50 megabytes of memory instead of 32. (This parameter is optional on FreeBSD and Linux.)

Stopping RealServer Under UNIX

To stop RealServer under UNIX, first obtain the process identification number, and then issue the **kill** command with that process number. The process ID is stored in the `rmserver.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 which 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. The usual name for this file is `rmserver.pid`.

Configuring MIME Types

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

In addition, RealServer serves its own HTML pages. To this end, be sure that RealServer has the correct MIME type information.

► **To set up MIME types on the Web server:**

Refer to the instructions accompanying your Web server to define the following MIME types on your Web server:

- audio/x-pn-realaudio (files with a `.ra`, `.rm` or `.ram` file extension)
- audio/x-pn-realaudio-plugin (files with a `.rpm` file extension)
- application/smil (files with a `.smi` or `.smil` extension)
- application/sdp (files with a `.sdp` extension)
- application/x-pn-realmedia (files with `.rp`, extension)
- text/html (files with a `.html` or `.htm` extension)
- image/gif (files with a `.gif` extension)
- image/jpg (files with a `.jpg` or `.jpeg` extension)

When you install RealServer, the MIME Types section is present in the configuration file. You need only examine this list if something happened in the meantime and you think the list might be complete. You can examine the MIME types section using the following instructions.

Additional Information

See “Customizing RealServer Features” for instructions on using RealSystem Administrator.

► **To set up MIME types used by RealServer:**

1. In RealSystem Administrator, click **General Setup**. Click **MIME Types**.
2. The list should match the table below:

Names	Extensions
audio/x-pn-realaudio	.ra .ram
application/smil	.smi .smil
application/sdp	.sdp
application/x-pn-realmedia	.rm .rp .rt
text/html	.html .htm
image/gif	.gif
image/jpg	.jpg .jpeg

You should only modify the list if you will be streaming a data type via HTTP that is not on the list.

- To add another MIME type, click **Add**. Type the name and extension in the respective boxes, and click **Submit**.
- To edit an existing MIME type, select it from the **Names** list, and click **Edit**. Change the name or extension and click **Add**.
- To remove a MIME type, select it from the **Names** list, and click **Remove**. Click **OK**.

License Information

Information about the license for your RealServer, including a list of enabled features, is stored in a file in a license directory. If you purchase additional features, these will be listed in additional files stored in the same directory. The license files are written in XML format.

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

Additional Information

To learn how to modify RealServer settings, see “Customizing RealServer” on page 35.

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. However, if you have multiple files, you will need to read them individually and calculate any additive features (such as number of streams) yourself.

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

The following features are controlled by the license:

- Number of streams
- RealPlayer versions—whether the administrator can modify which versions are allowed to connect
- Splitting—whether the RealServer can act as a source or splitter
- Multicasting
- Authentication
- Data types (such as RealVideo, and WAV)

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 5.0 and RealPlayer G2 are allowed to connect.
Splitting	Receive only. Cannot act as source.
Multicasting	Disabled
Authentication	Encoder and RealSystem Administrator users can be authenticated. Links to content cannot be authenticated.
Data types RealVideo streaming	RealVideo® is enabled; all other types (RealFlash™, WAV, AVI, VIVO) are disabled

Note

Evaluation versions may have lower minimum values.

CUSTOMIZING REALSERVER FEATURES

Chapter 4

All 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.

Customizing RealServer

When the RealServer installation program completes, it asks if you want to run RealSystem Administrator. If you choose yes, RealSystem Administrator displays. To make changes to any feature, click on the appropriate category listed under **Configure**. Make the changes and click **Apply**.

Starting RealSystem Administrator

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

► **To start RealSystem Administrator:**

1. Start RealServer. See Chapter 3: Starting and Stopping RealServer.
2. In a browser, type the following address:

`http://realserver.company.com:AdminPort/admin/index.html`

where:

realserver is the name the machine on which RealServer is installed.

company.com is the name of the domain in which RealServer exists.

Or, rather than typing the name and domain of the system on which RealServer is installed, you can type the IP address.

AdminPort is the port which RealSystem Administrator uses to connect to RealServer. You are 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 10: Authenticating RealServer Visitors.) Click **OK**.

RealSystem Administrator appears.



Using RealSystem Administrator

Once you have started RealServer and then 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**.

RealSystem Administrator makes the changes to the configuration file.

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 “RealSystem Administrator User Authentication” on page 104.

Configuration File

Changes made with RealSystem Administrator are stored in the configuration file. It is a text file formatted with tags which are based on XML (Extensible Markup Language). This language introduces great flexibility to the configuration file format and allows third-parties to use this file and add to its functionality. Syntax of this file is given in Appendix B: 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

You can change the RealServer settings by opening the configuration file with any text editor. You can also add variables that aren't included in the initial file, but are listed in this manual in Appendix B: Configuration File Contents. In addition, third-party plug-ins may require their own parameters and variables. Use a text editor to add them to the configuration file.

To make changes to existing settings in this file is simple; this manual provides guidance. If, however, you plan to add new sections, you will need to understand the syntax of the entire file. The file is organized into sections. This is not strictly necessary, but helps with clarity. The structure of the

configuration file is described in detail in Appendix A: Configuration File Syntax.

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.

When you edit the configuration file manually, 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 the “Minimum Settings” table on page 33.

Note

Always restart RealServer after changing any settings in the configuration file with a text editor.

RealSystem Administrator shows the configuration file settings of the RealServer configuration file in use; use caution if you are switching between manually editing the file and using RealSystem Administrator to edit it.

Warning

Exit RealSystem Administrator before opening the configuration file with a text editor or unexpected changes may result.

Common Settings

Regardless of which features are in use, certain powerful settings apply to every RealServer. They are described in this section.

Port Variables

Port settings tell RealServer where to listen for requests. Ports are described in detail in Chapter 2: Overview.

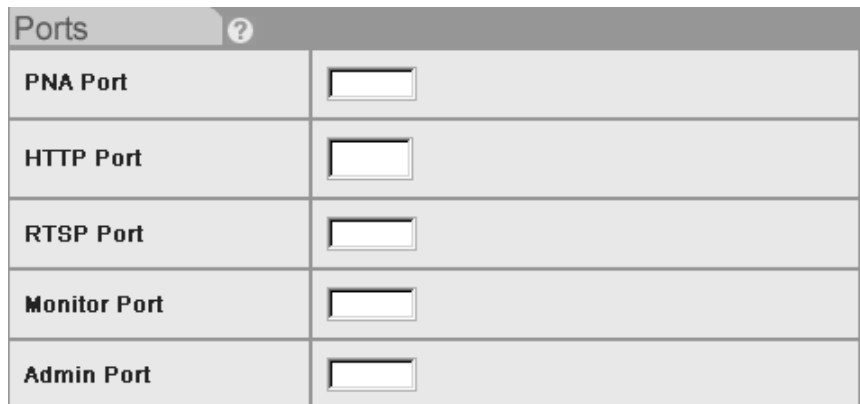
If your RealServer and Web server are on the same machine, you may need to modify the HTTP Port setting. See “Running Web Servers and RealServer on the Same System” on page 47 for additional information.

Warning

If you change the port settings from their default values, you must also change the links to show the port numbers.

► To add port settings:

1. In RealSystem Administrator, click **General Setup**. Click **Ports**.



Ports ?	
PNA Port	<input type="text"/>
HTTP Port	<input type="text"/>
RTSP Port	<input type="text"/>
Monitor Port	<input type="text"/>
Admin Port	<input type="text"/>

2. Tell RealServer where to listen for material requested via PNA (these begin with `pnm://`) by typing the correct value in the **PNA Port** box. The default value is 7070.

Previous versions of RealSystem used this protocol. If you have Ram files from older versions of RealSystem that use URLs that begin with `pnm://`, be sure to include this setting.

3. Tell RealServer where to listen for HTTP requests by typing the correct value in the **HTTP Port** box. The default value for this setting is 8080.
4. Tell RealServer where to listen for RTSP requests (these begin with rtsp://) by typing the correct value in the **RTSP Port** box. 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.

5. Tell RealServer where to listen for RealSystem Administrator connection requests by typing any unique port number in the **Admin Port** box.
6. When you have finished making changes, click **Apply**.

Mount Points

Mount points on this page refer to on-demand clips. For a complete description about the purpose of mount points, see “Mount Points” on page 16. Mount points in other sections, such as for live material, are described in their respective chapters.

Main Mount Point (/)

The mount point shown with a single forward slash in the **Name** list is the mount point for most of the content served by your RealServer.

Warning

If you change the main mount point itself, you will have to modify any links that include the original mount point. Changing the base path only changes where RealServer looks for content.

► **To modify the main mount point:**

1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.
2. In the **Names** list, select /.
3. To change the mount point name, modify the name shown in **Mount Point**. The default name is a forward slash (/).
4. To change the base path, type the new path in the **Base Path** box.
5. Click **Apply**.

Adding Another Mount Point

To add another mount point for on-demand content, use the instructions below.

1. In RealSystem Administrator, click **General Setup**. Click **Mount Points**.
2. Click **Add**. A new browser window appears.
3. Type a description or name for this mount point in the **Description** box.
4. Type the new mount point in the **Mount Point** box. It must be unique.
5. Identify the location of the content by typing the full path in the **Base Path** box.
6. Click **Add**.

Chapter 5

ADVANCED FEATURES

This chapter covers features which are specific to the operating system, as well as reserving IP addresses for RealServer's use, running RealServer on the same system as a Web server, and working with firewalls.

RealServer Caching Features

A media cache is server software usually installed on an intranet and sometimes on a large ISP, that stores streamed media. When a client on the intranet or hosted by the ISP requests a streamed media file, the media cache intercepts the request and sends it on behalf of the client. The media cache then stores the requested media and streams it to any other clients who subsequently request the same material. By sharing the distribution load, media caches conserve bandwidth over the Internet and allow RealServers to send streams to a wider audience.

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

If there is content served by your RealServer which 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.

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

Additional Information

The cache.log file is explained in Chapter 13: Reporting.

Making Your Content Cacheable

Instruct RealServer to permit RealServer to stream clips requested via a media cache by doing the following two things:

- Configure the plug-in which fulfills cached requests
- Indicate which directories should not be cached

Enabling Requests from a Media Cache

Two settings turn on the streaming of cached requests.

► **To allow content to be cached:**

1. In RealSystem Administrator, click **Cache**. Click again on the word **Cache** that appears below it.

Cache ?	
Cache Port	<input type="text"/>
Cache Requests	Disabled ▾
Cache Log	Disabled ▾
Cache Log Path	<input type="text"/>
No-Cache Directories/Files	<input type="text"/>
+ ADD A NO-CACHE DIRECTORY - REMOVE A NO-CACHE DIRECTORY	

2. In **Cache Port**, be sure the number 7802 is shown. This is the port number at which requests sent via media cache will arrive. If you change this value, requests by media caches will not be accepted by RealServer, and therefore will not be cached. If you change the default value of 7802, you will disable

streams to media caches unless you inform the administrators of all media caches that are accessing your streams, and tell them of the new value.

3. In **Cache Requests**, select **Enabled** from the list to turn on this feature.
4. Click **Apply**.

Limiting Caching

Unless you specify otherwise, all material on your RealServer may be cached. You can restrict the virtual directories that are available to such requests. If RealServer receives a request for material included in the No Cache Directories/Files 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.

In addition to indicating specific directories or files that are not cacheable, you can indicate that certain clients or media caches are not allowed to cache any of your material. You can also disable all caching of all material from your RealServer.

► **To prevent media caches from caching material on your RealServer:**

1. In RealSystem Administrator, click on **Cache**. Click again on the word **Cache** that appears below it.
2. In the **No-Cache Directories/Files** section, click the **Add** button.
3. In the **No-Cache Directory** box, type the name of the virtual directory whose content you want to restrict.
4. Click **Add**.

► **To prevent certain media caches from making requests:**

Create an access rule for the media cache 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

See “Limiting Access Via IP Address” on page 93.

► **To prevent all caching of all material from all clients and media caches:**

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**.

Reserving IP Addresses for RealServer's Use

When RealServer starts, it uses the first IP address of the first interface card it detects. If there is more than one IP address on the machine on which RealServer is installed, the operating system assigns an address to RealServer. Because the operating system's assignments may be random, clients attempting to connect to your RealServer may not be able to receive streams.

You can configure RealServer to always use the same IP addresses by setting up the IP Binding list. Within this list, you cite individual addresses to use, or you can reserve all the IP addresses available to the machine on which RealServer is installed.

Additional Information

Instructions on customizing RealServer can be found in Chapter 4: Customizing RealServer Features on page 35.

► To reserve IP addresses for RealServer:

1. In RealSystem Administrator, click **General Setup**. Click **IP Binding**.
2. Click the **Add** button.
3. In the **IP Address** box, type the address or DNS name that you want RealServer to use. Typing an IP address here, rather than the DNS name, allows RealServer to be more efficient.

RealServer will bind to the specified addresses only; it will not bind to localhost.
4. 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.

Warning

Use either 0.0.0.0 or other addresses, but not both. If you use both, RealServer will not start.

5. Click **Add**.

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

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 does not contain a port number, the client supplies a port number—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 the port number in the URL.

Set IP Binding List

You may need to reserve at least one IP address for RealServer's use. See the "Reserving IP Addresses for RealServer's Use" section, above.

Firewalls and RealServer

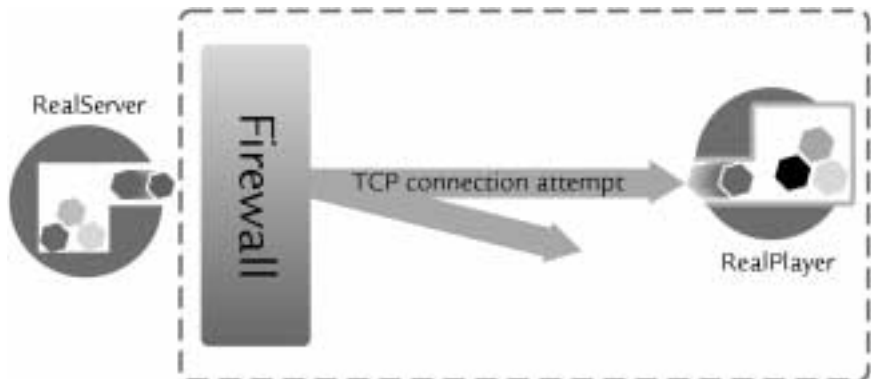
A firewall is a software device located on a network that monitors all transmissions between the organization's network and the Internet. The firewall's role is to ensure that all communication, in both directions, conforms to the organization's security policies.

As shown in “Communication Between RealServer and RealPlayer” on page 12, a client requesting presentations from RealServer first establishes a two-way TCP connection to the RealServer. RealServer uses this connection initially as a means of sending information to the client about the streamed 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.

After the initial connection is established, RealServer then establishes a UDP channel back to the client. The actual media is sent along this channel. The UDP channel is more like a custom radio channel than a telephone call; the client has no way of sending information back to RealServer over this UDP channel.

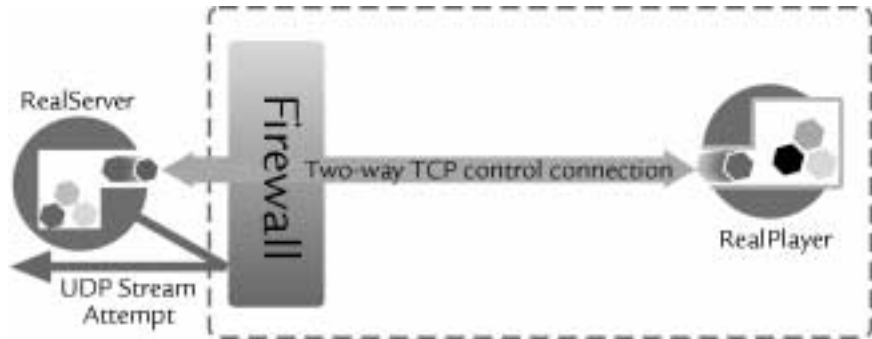
In general, firewalls permit one-way access to the Internet. Because RealServer and the client need to establish two-way communication to stream and receive media content, firewalls may reject a client’s attempt to establish this connection, and the client’s request for a clip will “bounce” off the firewall.

Firewall Rejecting Two-Way Connection Attempt



Other firewalls may be configured to permit the two-way TCP connection, but will then reject the one-way UDP data stream.

Firewall Rejecting One-Way Connection Attempt



Similarly, if RealServer is behind a firewall, it will not be able to open the necessary channels to communicate with clients.

Communicating with Clients Behind Firewalls

Some very strict firewalls allow only HTTP transmissions. If a client that is behind such a firewall attempts to view presentations streamed by your RealServer, they may not receive your content. RealPlayer includes an option to request that all streams be sent in HTTP format.

To receive these clients' requests, verify that the HTTP Port value is 80 or 8080. This works when RealServer is on a different computer than the Web server, or has a separate IP address.

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.

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.

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.

Windows NT

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” on page 26.

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 Chapter 12: Monitoring Activity on page 43.

UNIX

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

Process ID (PID)

RealServer creates a text file that stores the current value of the process ID of the main RealServer file, `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

When you make changes to RealServer using RealSystem Administrator, those changes are saved and RealServer is restarted immediately. If you make changes to the configuration file manually, you will need to restart RealServer yourself. 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 `rmserver.pid` file.

Chapter 6

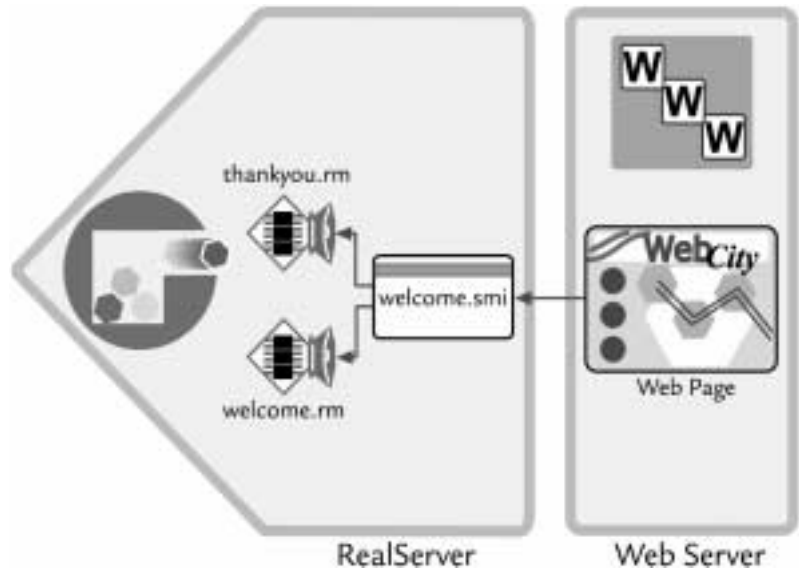
STREAMING PRESENTATIONS ON DEMAND

This chapter describes how to prepare RealServer for streaming on-demand, or pre-recorded, presentations.

Overview

RealServer is ready to stream content when you first install it, and will stream static media that you place in the Content directory.

Location of Files



Notice that the Web pages do not link directly to the media; they link to the SMIL files located on the RealServer. If the Web pages were to link directly to the media files, these files would be downloaded similarly to HTML files, and the visitor would not receive the material quickly.

Streaming On-Demand Clips

Setting up RealServer to stream static content consists of four steps:

1. Making sure RealServer is configured correctly.
2. Placing the content on RealServer.
3. Creating SMIL files that control the order and placement of media files.
4. On the Web pages, linking to the SMIL files.

Neither the content nor the SMIL files go on the Web server. The only things that you'll change on the Web server are the links that point to the SMIL files.

Note

If you have existing content that uses metafiles (.ram files), you'll only modify the links within the metafiles.

► To configure RealServer to stream content:

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. You can change where RealServer looks to find content by changing the base path value for the main mount point.

Additional Information

See "Mount Points" on page 40.

Ensure that the settings for HTTP Port, PNA Port and RTSP Port are correct. These indicate where requests will arrive at RealServer, and the links you create will need to match them.

Additional Information

See "Port Variables" on page 39.

► To place files on RealServer:

Place clips that you've encoded with RealNetworks tools in the RealServer Content subdirectory. If you do not want to use this directory, either place the files in a subdirectory of Content and add a virtual directory to the URL, or create another mount point to use for this content.

For more information about where to place clips, see "Storing Files and Presentations" on page 19.

► **To create SMIL files:**

SMIL files control the synchronization of media files. If you want to stream several clips, create a SMIL file. If you're linking to a single file, you don't have to create a SMIL file.

Additional Information

See "SMIL Files" on page 13. In addition, *RealSystem G2 Production Guide* gives detailed information on SMIL files.

► **To link Web pages to files:**

On the Web pages on your Web server, you'll add links that point to the SMIL files which you just created. See "Links from Web Pages to Streamed Media Clips" on page 20.

Working with SureStream Files

Visitors to a site will view your content via different bandwidth volumes. When a client requests a clip, it sends its bandwidth capabilities to the RealServer. RealAudio® and RealVideo® files encoded with the RealNetworks encoding tools 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. RealServer switches between streams automatically.

If the file to be streamed 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 datatype, such as text, bandwidth negotiation is handled via a SMIL file.

Instructions on doing this are available in *RealSystem G2 Production Guide*.

Chapter 7

BROADCASTING 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

Streaming live content is much the same as streaming static content. The only difference is the live file never actually exists. It is streamed as it's encoded, and a file is never actually created. 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.

RealServer can save a copy of all live broadcasts automatically, or it can save only broadcasts with specific virtual directory names.

Use the Simulated Live Transfer Agent (**G2SLTA**) to broadcast a stored file as if it were live. Events broadcast with **G2SLTA** appear to be live; everyone sees the same part of the broadcast at the same time.

Tip

RealBroadcast Network™ (RBN) provides full services for encoding and broadcasting events to a few or a few thousand viewers. See <http://www.real.com/rbn> for details.

Live Broadcasting

Setting up RealServer to broadcast live files consists of four steps:

1. Configuring RealServer broadcasting mount points.
2. Configuring RealServer port settings.

3. Configuring the encoder.
4. Placing the correct link on the Web page or SMIL file.

Additional Information

Instructions on using RealSystem Administrator to modify RealServer is found in “Customizing RealServer” on page 35.

► To configure G2 encoder mount points:

1. In RealSystem Administrator, click **Broadcasting**. Click **G2 Encoders**.

G2 Encoder ?	
Description:	<input type="text"/>
Mount Point:	<input type="text"/>
Port:	<input type="text"/>
ShortName:	<input type="text"/>
Encoder Authentication Realm:	<input type="text"/>

2. To change the mount point, modify the value in the **Mount Point** box. The G2 encoder typically uses the /encoder/ mount point.
3. Give the port number to which the encoder will send its live data by typing a number in the **Port** box. For RealSystem G2 encoders, a typical value is 4040. If you change this value, be sure to give the new port number to content creators.
4. If you have event files that you want to combine with your live stream, type their location in the **Associated Media Path** box. Any files in this directory with the same name as an incoming live stream will merge their events with the live stream.
5. Content creators using RealSystem G2 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 **Encoder Authentication Realm** 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 10:
Authenticating RealServer Visitors.

6. When you have finished making changes, click **Apply**.

► **To configure pre-G2 encoder live settings:**

1. In RealSystem Administrator, click **Broadcasting**. Click **Pre-G2 Encoders**.
2. To change the mount point, modify the value in the **Mount Point** box. The pre-G2 encoders use the /live/ mount point.
3. Give the port number to which the encoder will send its live data by typing a number in the **Port** box. For encoders created before version G2, a typical value is 5050. If you change this value, be sure to give the new port number to content creators.
4. If you have event files that you want to combine with your live stream, type their location in the **Associated Media Path** box. Any files in this directory with the same name as an incoming live stream will merge their events with the live stream.
5. Encoders developed before RealSystem G2 are able to supply passwords, but no user name. To require a password for older encoders, type the password in the **Password** box. The value was initially established during setup. All encoders connecting to the older encoder port will use this same password.
If you change the password, be sure to tell content creators what password to use.

6. When you have finished making changes, click **Apply**.

► **To configure the ports:**

Ensure that the settings for HTTP Port, PNA Port and RTSP Port are correct. These indicate where requests will arrive at RealServer, and the links you create will need to match them.

Additional Information

See “Port Variables” on page 39.

► **To connect the encoder to RealServer:**

When a content creator sets up the encoder to broadcast a performance, he or she will indicate the RealServer port numbers to which it should send live streams.

If a G2 encoder is in use, and you have typed a value in the **Authentication Realm** box (described in the previous section), the person using the encoder will need to type a user name and password.

If a pre-G2 encoder (such as RealEncoder version 5.0) is in use, and you instructed RealServer to require a password by typing in the **Password** box, the person using the encoder will need to type a password.

Additional Information

Refer to your encoding software documentation for instructions on setting up the encoder.

► **To link the Web page or SMIL file to the live stream:**

Links to live events are similar to links for on-demand clips, with the addition of the mount point.

Follow the format of linking to an individual file, and use the live file mount point, usually /encoder/.

Additional Information

See “Links from Web Pages to Streamed Media Clips” on page 20.

Playing A “Please Stand By...” Message

Should a live stream be 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, and place it under the main base path.
2. In this subdirectory, place a file in the same format as your broadcast (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 in which visitors may live).

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.

Archiving Live Files

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 the next section, “Simulating a Live Broadcast”.

When live archiving is enabled, RealServer examines all arriving live streams, and compares the names of the streams with the list names within the Live Archiving section of RealSystem Administrator or the configuration file. If it contains a list whose name matches the virtual directory 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 Target Directory.

RealNetworks’ encoding products include an option to save a copy of a file while encoding. This setting is 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
One large file	<ul style="list-style-type: none"> • Corporate presentations • Concerts

(Table Page 1 of 2)

Live Archiving Options

Method of archiving	Suggested use
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

(Table Page 2 of 2)

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 `filename01.rm`. When the second archived file maximum size is reached, it is named `filename02.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 renames the existing file with the date and time that the live file was initiated where the date (`filename`) is in the format `MMDDYY`.

Setting Up Live Archiving

In its default configuration, RealServer puts all the files in the same directory, specified by the main mount point's Base Path.

► To archive live streams:

1. In RealSystem Administrator, click **Broadcasting**. Click **Live Archiving**.

2. Click **Add**.
3. In the new browser window that appears, type a description for the live archiving setting in the **Description** box.
4. In the **Virtual Path** box, type the virtual path where the live data will arrive. This is the same text that the content creator will type in the encoder, including the live broadcasting mount point.
If you want to archive all live streams, name this list with an asterisk (*) rather than typing the virtual path.
5. In the **Target Directory** box, name the virtual directory where RealServer should store the files.
Target Directory can also be set to an absolute directory.
6. To limit the files by their size, select **File Size** and type the maximum desired size in megabytes.

To limit the size of the archived files by time, select **File Time**. Select the frequency with which a new archive file will be selected; use the table below.

Example FileTime Values	
FileSize 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

If you give values to both **File Time** and **File Size**, RealServer will use the first, or lower, limit.

To save entire broadcasts without limiting the file size, omit values for both **File Time** and **File Size**.

7. If RealServer 5.0-style bandwidth is in use, select the **Bandwidth Negotiation** box. When this box is selected, and RealServer receives streams from encoders with the .rm extension, RealServer creates a directory named after the filename, including the extension. All the streamed files

go in this subdirectory. For more information on 5.0-style bandwidth negotiation, see “Files Created with Previous Encoder Versions” on page 55.

Note

If you are using bandwidth negotiation to create the files and the **Bandwidth Negotiation** box is clear, RealServer must make a choice as to which file it stores in the archive directory. It will store the first stream that arrives, as *filename.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.

Disabling Live Archiving

To turn off live file archiving, modify the `NoArchive` variable in the directory list which you do not want to archive. Set `NoArchive` to `True`.

Simulating a Live Broadcast

For replaying a pre-recorded stream as if it were live, RealServer includes the **G2SLTA** (Simulated Live Transfer Agent). 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.

Tip

Use **G2SLTA** to test your system in anticipation of an actual live broadcast.

There are three steps to using **G2SLTA**: setting up RealServer, creating a playlist of files to stream, and running **G2SLTA**. You will also need to link the Web page to the broadcast.

Setup for RealServer is minimal; **G2SLTA** uses the G2 Encoder information.

The playlist is a list of files that **G2SLTA** will play. If you want to simulate a live broadcast of only one file, the playlist will refer just to that file. If you have a series of files you want to play during your simulated live broadcast, list them in sequence in the playlist. **G2SLTA** includes an optional command to play files in a playlist in random order. The playlist itself is a text file.

When you start **G2SLTA**, you give a name to the stream, similar to a mount point. This is the name that will be included in the URL. The playlist is not included in the URL.

Tip

Because the URL is linked to the list of streamed files via the **G2SLTA** command line, you can use a different file name for each simulated live broadcast, yet the link on the Web page remains the same.

Setting Up G2SLTA

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

► To configure RealServer to work with G2SLTA:

This program uses the same configuration settings as the encoders. See “To configure G2 encoder mount points:” on page 58.

► 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 they are listed. File paths and names can be absolute, or they can be relative to the directory in which **G2SLTA** is located.

Even if you have only one presentation that you want to broadcast, you must still create a playlist file.

To include title, author, and copyright information for the entire playlist, type the following at the beginning of the file:

Title: *your title*
Author: *your author*
Copyright: *your copyright information*

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

► To run G2SLTA:

From a command line, run **G2SLTA**:

G2SLTA Syntax

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

where:

<i>host</i>	Name of the RealServer system and domain name.
<i>port</i>	Port number specified in the pn-encoder list, usually 4040.
<i>username</i>	Name of the encoder user as defined in the encoder realm.
<i>password</i>	The corresponding user's password. If you are not using authentication for encoders, you can omit both the user name and the password.
<i>livefile</i>	Name of the broadcast that you want to include in the URL that links to this event.
<i>playlist</i>	Name you gave to the playlist.
<i>-r</i>	Indicates that RealServer should randomly play the files in the playlist.
<i>-nN</i>	Gives the number of files in the playlist for RealServer to play. To indicate that RealServer should play seven files, include <i>-n7</i> 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. To play a list of files indefinitely, omit the number from this switch. Use <i>-n</i> only.

An example of a **G2SLTA** command:

```
g2slta realserver.company.com 4040 swordfish annual.rm Annual_Report.txt
```

Tip

You can use both the *-r* and *-nN* switches to cycle randomly through the playlist *N* times.

► **To create the URL that links to the G2SLTA presentation:**

In the Web page, create a link to the content using the format described in “Linking a Web Page to a SMIL File or Individual Clip” on page 21. Use the encoder mount point.

Chapter 8

SPLITTING AND MULTICASTING

RealServer has two methods that reduce the number of streams while distributing high-quality live streams: splitting and multicasting.

Overview

In delivering clips, RealServer sends one stream to each client that requests a clip. This method is called unicasting. A server can easily become overloaded when many clients connect to a single server to receive a presentation. Also, the network can easily become saturated with media data being sent from RealServer to many clients.

Splitting makes efficient use of RealServer networking and resources by redirecting one live stream to another RealServer, which in turn serves the stream to other clients. The load on the original RealServer is thus reduced.

Multicasting is an efficient way of broadcasting a live event over a multicast-enabled network. Multicasting uses less network bandwidth since data packets are sent to all clients via a single transmission. There is no point-to-point connection made between the client and server for the data stream.

Splitting and multicasting are used only for live content.

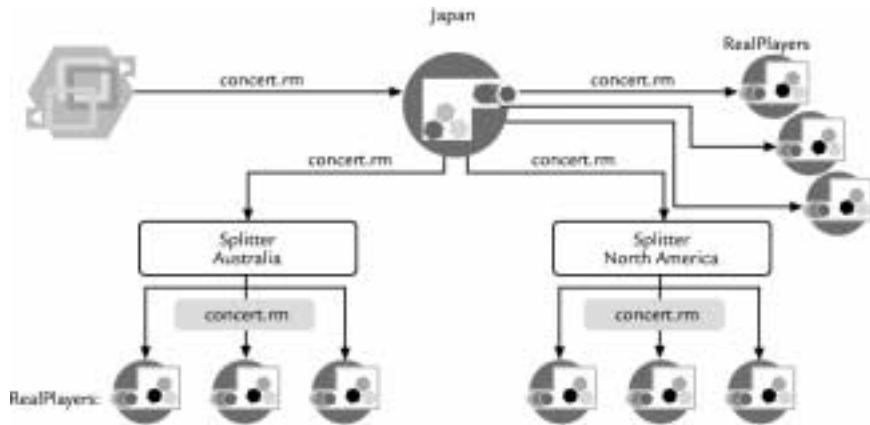
Splitting

Splitting solves the problem of one server getting overloaded by stream requests for live material. In splitting, one or more RealServer servers join the main RealServer to distribute the number of streams. Rather than having all the requests come to one server, the RealServer to which the encoder is attached (also called the source splitter, or just the RealServer) sends one stream to the other RealServer (called the splitter). The splitter can also serve the content, thus multiplying the number of available streams.

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 RealServer closest to them, thereby getting better media quality and performance. While serving content that originated on another computer, a RealServer can simultaneously stream its own content.

Splitting



Web pages listing the event would have different links for different locations:

Sample Web Page



RealServer Splitting Methods

There are two types of splitting: push splitting and pull splitting.

In pull splitting, the link to split material includes the locations of both the source and the splitter. It requires few changes to the configuration file on either computer.

Just as in unicasting, both splitting methods support bandwidth-negotiated files (such as SureStream).

Push Splitting requires more setup time, but it can mean faster connections to the first client requesting a split stream.

Comparison of Push Splitting and Pull Splitting

Push Splitting	Pull Splitting
Pre-establishes a connection, so when first client connects there's no wait time	Does not pre-establish a connection. After the first client connects, the connection remains until the encoder stops the session.
Uses bandwidth while waiting for someone to listen.	The first client to request split material has to wait 30 seconds or so while a splitter connection is built. May use small amount of bandwidth even if no clients are playing streams.

Push Splitting

In push splitting, RealServer and the splitter are in constant communication. When a client requests a file from the splitter, a connection has already been established between the splitter and RealServer and the file is delivered to the client almost immediately. If there is already a connection between splitter and RealServer and a new encoder hooks up, RealServer will immediately feed that stream to the splitter.

You can limit which directories are split by setting a variable in the relevant directory section.

Pull Splitting

In pull splitting, the link on the Web page lists the splitter and the RealServer where the streamed file originates.

When the splitter gets a request for the live file, it opens a stream to the RealServer, and the RealServer streams the file to the splitter, which, in turn, streams it to the client.

This method only uses bandwidth when a client requests a split stream. In addition, the only configuration steps are to make certain that both the RealServer and the splitter have a splitter file system listed.

Controlling Splitter Access to Your RealServer

You can specify the splitters that are allowed to query your RealServer for live streams by adding their IP addresses to the Access Control list. In this list, you indicate the ports to which they can send their requests.

If you do not limit the splitters, any splitter can access your server.

Using Both Push Splitting and Pull Splitting

You can combine these methods to make the best use of your bandwidth.

Setting Up Push Splitting

To set up push splitting, make changes in three places:

1. Edit the source RealServer settings.
2. Edit the splitter's settings.
3. Add splitter URLs to a SMIL file or Ram file, which in turn is linked to the Web pages.

Additional Information

Information on editing the configuration file is found in Chapter 4: Customizing RealServer Features on page 35.

► To set up the source RealServer for push splitting:

1. In RealSystem Administrator, click **Splitting**. Click **Push**.

Splitting - Push ?	
Description	<input type="text"/> <input type="button" value="+ ADD A PUSH SPLITTER"/> <input type="button" value="- REMOVE A PUSH SPLITTER"/>
Mount Point	<input type="text"/>
Host Name	<input type="text"/>

2. In the **Splitter Description** section, click **Add**. A new browser window appears.
3. Type a name for this splitter configuration in the **Description** box.
4. In the **Mount Point** box, type the mount point you want to use in the URLs.
5. Type the name of the machine on which this RealServer is running in the **Host Name** box. RealServer uses this value to identify itself in the resource URL when it feeds a live stream to a splitter.
6. Click **Add**.

You are returned to RealSystem Administrator. Look in the **Options for Splitting Settings** area.

Options for Splitting Sources	
Split All Streams	<input type="button" value="Yes"/>
Resend Buffer	<input type="text"/>
Source Timeout	<input type="text"/>
Sources	<div style="border: 1px solid gray; height: 60px; width: 100%;"></div> <div style="margin-top: 5px;"><input type="button" value="+ ADD A SPLITTER SOURCE"/> <input type="button" value="+ EDIT A SPLITTER SOURCE"/> <input type="button" value="- REMOVE A SPLITTER SOURCE"/></div>

7. If you want all live streams originating from this RealSystem Administrator to be sent to other splitters, select Yes in the **Split All Streams** list.
8. If you want to split only specific live streams, click **Add** to add the virtual path names to the configuration file.
9. In the new browser window that appears, type the name of the virtual path in the **Source Name** box.
10. From the **Splitting** list, select Enabled or Disabled.

11. Click **Add** to return to RealSystem Administrator.
 12. In the **Splitter Resend Buffer** box, type the size of the buffer (in seconds) for UDP resends. It can range from 0 to 32767; the default value is 30.
 13. If you want to limit how many seconds RealServer will wait before it stops sending data to a splitter that is not responding, type a value in the **Splitter Source Timeout** box. The default value is 30.
 14. List the splitters that are allowed to obtain content from this RealServer by adding their IP addresses to the **Access Control** list. Add one rule for each IP address or range of addresses. List the port or ports to which the receive splitters can direct their requests. See “Limiting Access Via IP Address” on page 93 for instructions.
- **To set up the splitter for push splitting:**
1. In RealSystem Administrator, click **Splitting**. Click **Push**.
 2. In the **Splitter Description** section, click **Add**.

Splitting - Push ?	
Description	<input type="text"/> <input type="button" value="+ ADD A PUSH SPLITTER"/> <input type="button" value="- REMOVE A PUSH SPLITTER"/>
Mount Point	<input type="text"/>
Host Name	<input type="text"/>

3. In the new browser window that appears, type a name for this splitter configuration in the **Description** box.
4. In the **Mount Point** box, type the mount point you want to use in the URLs.
5. In the **Port** box, type a port number to which the source RealServer will send its streams. In other words, the value for refers to the port number on the receive splitter. The default value is 11001.

6. Type the name of the machine on which this RealServer is running in the **Splitter Host Name** box. RealServer uses this value to identify itself in the resource URL when it feeds a live stream to a splitter.
7. Indicate where the `encnet.dll` (Windows) or `encnet.so.6.0` (UNIX) file is located by typing its location in the **SupportPathDirectory** box. This is usually the RealServer Lib directory.
8. Click **Add**.

You are returned to RealSystem Administrator. Look in the **Options for Splitters** area.

Options for Splitters	
Port	<input type="text"/>
Buffer Delay	<input type="text"/>
Timeout	<input type="text"/>
Source Probe Interval	<input type="text"/>
Sources	<div style="border: 1px solid gray; height: 60px; width: 100%;"></div> <div style="text-align: right;">+ ADD A SPLITTER SOURCE - REMOVE A SPLITTER SOURCE</div>

9. Define how many seconds of data to store in the buffer by typing a number in the **Splitter Buffer Delay** box. This helps reduce packet losses (dropouts) over a splitter connection. The recommended value is 30 seconds; a minimum of at least 10 seconds should be used.
10. Define how long the splitter will wait before considering a stream inactive. Type the number of seconds, from 0 to 32767 in the **Splitter Timeout** box. The default value is 30.
11. Set how often the splitter will request a stream. Type this number in the **Splitter Source Probe Interval** box. This value is given in seconds, and can range from 0 to 32767. The default value is 30.

12. List the RealServer or RealServers that the splitter should contact for live material to split. In the **Splitter Sources** area, click **Add**.
 13. In the new browser window that appears, type a name for the source in the **Description** box.
 14. In the **Address** box, type the name of the source RealServer. This splitter will contact the RealServer and request its streams. This splitter will, in turn, split those.
 15. In the **Port** box, type the port number of the source RealServer to which this splitter will send its requests.
 16. Click **Apply** to return to RealSystem Administrator. Click **Apply** again.
- **To create the link for push splitting within a SMIL file or Ram file:**
`rtsp://domain_name:RTSPPort/Splitter_MountPoint/SplitterHostName/
 Live_MountPoint/Virtual_Directory/filename.xxx`

RealServer URL Components

Component	Meaning
<code>rtsp</code>	The protocol used for streaming.
<code>domain_name</code>	Domain name of this RealServer. IP address may be substituted.
<code>RTSPPort</code>	Port number where RealServer listens for requests sent via RTSP. This value is usually 554; see “Port Variables” on page 39.
<code>Splitter_MountPoint</code>	The push splitting mount point used on this RealServer, usually <code>/farm/</code> .
<code>SplitterHostName</code>	The source RealServer’s <code>SplitterHostName</code> value. To allow the client to receive streams which differ only by the source server’s <code>SplitterHostName</code> , use an asterisk (*). If multiple connections exist which match the query, this splitter decides which stream the client will receive.
<code>Live_MountPoint</code>	Mount point for live content on the source.
<code>Virtual_Directory</code>	The virtual directory, as defined by the encoder.
<code>filename.xxx</code>	The name of the live stream.

This URL can be used on both the source RealServer and the splitter. Be sure to use the domain name for the system where the link is located: on the source, use the source domain name, and on the splitter, use the splitter domain name.

Setting Up Pull Splitting

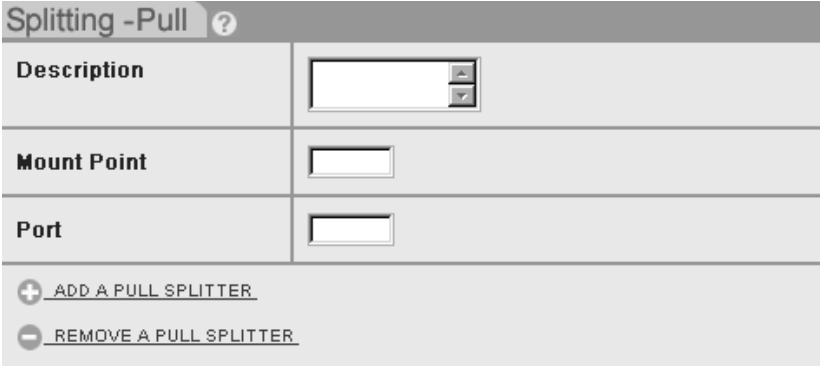
Like push splitting, pull splitting has three steps:

1. Edit the source RealServer settings.
2. Edit the splitter settings.
3. Add splitter URLs to the Web pages.

Configuring the source and the splitter is a quick matter; setting up the URL is more complicated.

► **To set up the source RealServer for pull splitting:**

1. In RealSystem Administrator, click **Splitting**. Click **Pull**.



Splitting -Pull ?	
Description	<input type="text"/>
Mount Point	<input type="text"/>
Port	<input type="text"/>
+ ADD A PULL SPLITTER	
- REMOVE A PULL SPLITTER	

2. Click **Add**. A new browser window appears.
3. In the **Splitter Description** box, type the name for this splitting configuration.
4. In the **Port** box, list the port to which the source RealServer will listen for splitter requests. A typical value is 3030.
5. Click **Add**.

You are returned to RealSystem Administrator.

► **To set up the receive splitter for pull splitting:**

1. In RealSystem Administrator, click **Splitting**. Click **Pull**.
2. Click **Add**. A new browser window appears.
3. In the **Splitter Description** box, type the name for this splitting configuration.

4. In the **Mount Point** box, type the mount point you want to use in the URL.
A typical value is /split/.
5. Click **Add**.

You are returned to RealSystem Administrator.

► **To create the link for pull splitting:**

The link that appears on the Web page that points to the receive splitter looks like this:

```
protocol://splitter[:protocol_port]/splitter_MountPoint/source:[source_port]/
source_MountPoint/filename
```

RealServer URL Components

Component	Meaning
<i>protocol</i>	The protocol used in streaming. For example, rtsp.
<i>splitter</i>	Domain name where the receive splitter is installed. IP address may be substituted.
<i>protocol_port</i>	The protocol port on the splitter, as specified in the splitter's RTSP Port or PNA Port setting. For example, if the streaming method is RTSP, then the port value must match the splitter's RTSP Port setting. The <i>protocol_port</i> is optional, and need only be used if you have changed the port setting from its default value.
<i>splitter_MountPoint</i>	The receive splitter's Pull Splitting mount point, usually /split/.
<i>source</i>	Domain name where the source RealServer is installed.
<i>source_port</i>	The source RealServer's port number as specified by the Port value in the Pull Splitting list, usually 3030. This number is optional, and need only be supplied if you have changed the port value on the source. If you omit it, the splitter will use the default value of 3030.
<i>source_MountPoint</i>	The source RealServer's mount point that is appropriate to the material you're splitting, such as /encoder/.
<i>filename</i>	Name of the file being split.

For example, links in the sample shown at the beginning of this chapter would look like the following (note that the direct link to the Japan server is not in pull splitting format):

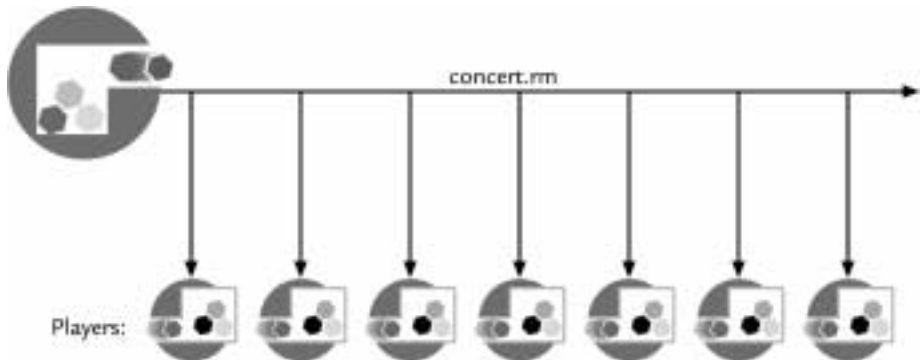
...we hope you enjoy the concert! Choose the link nearest you:

```
<a href="http://Japan.company.com.ja/concert.rm">Japan</a></p>
<a href="http://Australia.company.com.au:8080/split/
Japan.company.com:3030/encoder/concert.rm">Australia</a></p>
<a href="http://NorthAmerica.company.com:8080/split/
Japan.company.com:3030/encoder/concert.rm">North America
</a></p>
```

Multicasting

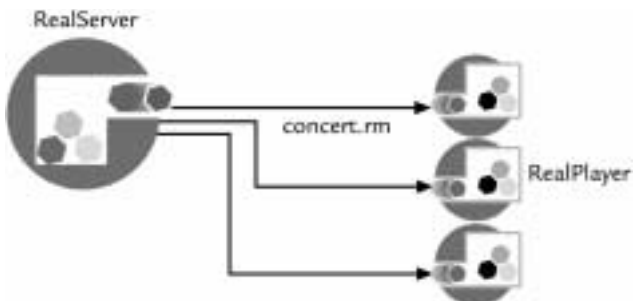
Multicasting is a way of sending a single live stream to multiple clients, rather than sending a stream to each of them.

Multicasting



In contrast, regular unicasting transmission sends a stream to each client who requests it:

Unicasting



To take advantage of multicasting, both RealServer and clients, as well as the routers between them, must be multicast-enabled. For this reason, multicasting is mostly used with intranets where routers can be configured for multicasts. Multicast delivery can be done over the Internet where intermediary network devices have been multicast-enabled, such as UUNET and the Internet Multicast Backbone (Mbone).

RealServer Multicasting Methods

RealServer includes two methods of multicasting: back-channel multicast, which has methods for RTSP and PNA multicast, and scalable multicast. You can use all methods at once.

Back-channel multicast maintains a TCP 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 TCP 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 monitoring graph in RealSystem Administrator will show client activity.

Scalable multicast does not use this TCP control channel. It thus takes up far less bandwidth and administrative overhead. Monitoring tools such as G2 Java Monitor will not track client activity.

Back-Channel Multicast

In both RTSP and PNA multicast, authenticated material and client statistics can be sent because the exchange between the client and RealServer is bi-directional.

RTSP Multicast

This method of multicasting uses RTSP to send control information over a TCP channel. RealServer maintains a control connection for each client. The data channel is multicast to all clients.

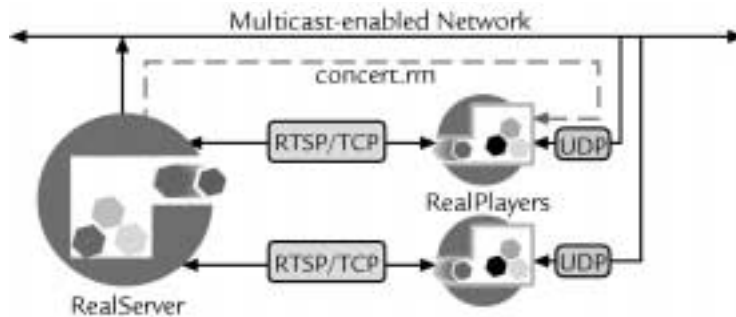
RTSP multicast provides the following features:

- **Authentication**—user name and password for secure content is sent securely.
- **Connection statistics**—RealServer can receive client connection information.

- **SureStream**—these multiply-encoded files are supported.

RTSP multicasting works only with RealSystem G2 clients.

RTSP Multicasting



PNA Multicast

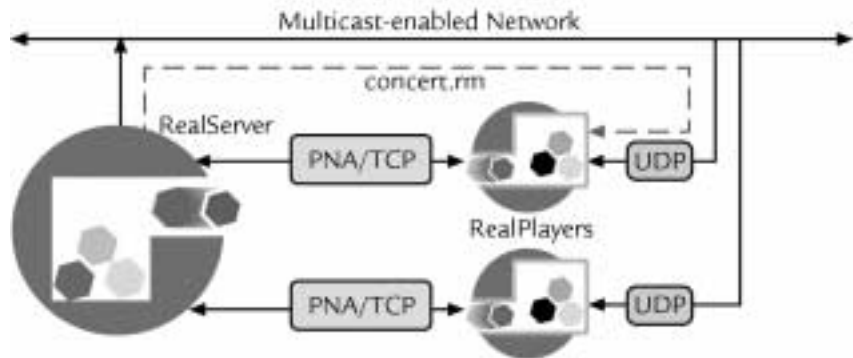
PNA multicast uses the PNA protocol over a TCP connection to exchange information between the client and RealServer.

PNA multicast is used when transmitting to older clients (pre-G2). RealServer maintains a control connection for each client. The data channel is multicast to all clients.

PNA multicast supports the following features:

- **Authentication**—user name and password for secure content is sent securely.
- **Connection statistics**—RealServer can receive client connection information.

PNA Multicasting



Scalable Multicasting

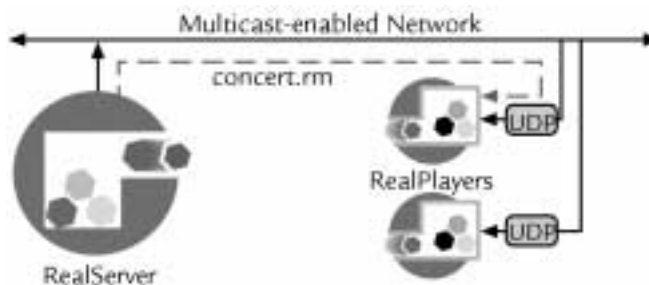
Scalable multicasting allows you to transmit to an unlimited number of clients because the transmission from the RealServer is completely one-way; there is no connection from each client to RealServer at all. All data is multicast on the network once. Each client connected to this multicast receives 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.

This method supports G2 clients only; clients version 5.0 and older will not receive any presentations.

Scalable Multicasting



Summary of Multicast Methods

The following table summarizes the benefits of each multicast method.

Feature	Multicast Methods		
	Back-Channel Multicast	PNA	Scalable Multicast
Control channel	•	•	
Authentication	•	•	•
Client statistics	•	•	
Minimal RealServer resource use			•
RTP enabled			•
SureStream support	•		
Live bandwidth negotiation		•	•

The following table shows the features of the three multicast methods, as they apply to clients

Feature	Back-Channel Multicast		Scalable Multicast
	RTSP	PNA	
RealSystem G2 clients only	•		
Older clients		•	
RealSystem G2 clients or any RTP-enabled clients			•

Logging Multicasts

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 may be identified in the access log by their mount point in the GET statement.

Because the communication between the RealServer and client during scalable multicast is the client's initial request for the multicast, the only statistics RealServer can track are the number of initial requests for the presentation. Information about the client is not available to RealServer. Statistics normally

shown by Stats Mask will not appear for scalable multicasts. (See “StatsMask Results” on page 137 for more information.)

Setting Up Multicasting

To take advantage of multicasting, both RealServer and clients, as well as the routers between them, must be multicast-enabled. For details, consult your network administrator. This section describes only what is required to enable RealServer for multicast broadcasting.

In addition to setting up RealServer, verify with your network administrator that the routers in your network are multicast-enabled and that the system running RealServer is correctly configured for multicast support. Also, ensure that clients are configured to request multicast transmission of live material.

Each type of multicast requires an address range. Values can range from 224.0.0.0 (the lowest usable address) to 239.255.255.255. The network administrator should know which multicast addresses are available on the intranet. On the Internet, certain ranges are reserved for other uses; see RFC 1700, “Assigned Numbers” for a complete list of restricted addresses.

Additional Information

Information on modifying RealServer features is found in “Customizing RealServer” on page 35.

Back-Channel Multicasting

Follow the instructions below to set up back-channel multicasting.

► **To set up back-channel multicasting:**

1. In RealSystem Administrator, click **Multicasting**. Click **Back-Channel**.

Back-Channel ?	
PNA Port	<input type="text" value="7070"/>
RTSP Port	<input type="text" value="554"/>
Address Range	<input type="text"/> to <input type="text"/> <small>Address Range values for back-channel multicasting must not overlap with scalable multicast Address Ranges, and settings must be between 224.0.0.255 to 239.255.255.255.</small>
Time to Live	<input type="text"/> Measured in router hops
Delivery Only	<input type="text" value="No"/> ▼

2. 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 typical value is 7070.
3. 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 typical value is 554.
4. Specify the range of addresses to which you want to multicast streams by filling in the **Address Range** box. 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.
5. In the **IP Address** section, click **Add**.
6. In the new window that appears, type a description for this list in the **Rule Number** box.
7. In the **IP Address** box, type an address to the domain address of the client computer or network for whom RealServer will permit multicast transmissions.

8. In the **Netmask** box, type a netmask that limits the range to a particular subnet.
9. Click **Add**.
10. To require that clients whose addresses you just listed use multicast only, and not unicast, select Yes from the **Delivery Only** list. To remove this restriction and permit unicast, set it to No.
11. To allow clients to request that missing UDP packets be resent, select True from the **Resend** list.
12. 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 will travel.
The default value of 16 is enough to keep multicast packets within a typical internal network.

Time to Live Values

TTL Value	Packet Range
0	Local host
1	Local network (subnet)
32	Site
64	Region
128	Continent
255	World

13. Click **Apply**.

Linking to Back-Channel Multicasts from a Web Page

Links to both RTSP and PNA multicast are identical to links for live unicast transmissions.

A single link can serve clients that are multicast-enabled and those that can only receive unicast transmissions.

Most clients on a multicast-enabled network are usually configured to request material via multicast first.


Links to back-channel multicasts are the same as for links to individual files. Because multicasts are done with live streams, use the live mount point in the URL. See “Linking a Web Page to a SMIL File or Individual Clip” on page 21 for instructions on creating links to individual files.

Scalable Multicasting

After you set up scalable multicasting, be sure to set up special URLs for links to scalable multicast presentations.

► **To set up scalable multicasting:**

1. In RealSystem Administrator, click **Multicasting**. Click **Scalable**.



The image shows a screenshot of a configuration window titled "Scalable". The window has a grey header bar with the word "Scalable" on the left and a question mark icon on the right. Below the header, there are two text input fields. The first field is labeled "Host Address:" and the second field is labeled "Mount Point:". Both fields are currently empty.

Set up the general scalable multicasting information. In the first section on this page, do the following:

2. Type the IP address of this RealServer host computer in the **Host Address** box.
3. Give the mount point that will be included in the links to all scalable multicasts. The default value mount point is /scalable/.

The following instructions describe how to configure certain live sources to be multicast.

4. In the **Live Sources** section, click **Add**.

5. In the new browser window that appears, type a descriptive name for this multicast session in the **Name** box.
6. Turn on scalable multicasting for this virtual path by selecting True from the **Enabled** list.
7. Type the name of the virtual directory in the **Virtual Path** box. The virtual directory is the path typed in the production tool that's 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.
To make all live broadcasts available via scalable multicast, type an asterisk (*) here.
8. Clients listen to a specific range of ports. Indicate this range, to which RealServer will direct its multicasts, by typing port numbers in the **Port Range** boxes. Any valid port numbers are acceptable, in the format *port1-port2*. The first port number must be an even number, and must be

followed by the consecutive port number. (RTP is used to send the data; the RTP standard requires this format.)

9. Specify the range of addresses to which RealServer can send a multicast stream by typing in the **AddressRange** box. RealServer uses the first available address in this range. When typing in this box, use the form address1-address2. To use a single address instead of a range, type the address in the form address1-address1.

Valid ranges are between 224.0.0.0 and 239.255.255.255. However, the addresses between 224.0.0.0 and 224.0.0.255 are reserved; in addition, RFC 1700 (“Assigned Numbers”) lists additional numbers which are restricted.

10. Indicate how far multicast packets can travel on your network in the **Time to Live** box. Use the values in the “Time to Live Values” table on page 84 for TTL.
11. Click **Add**. Click **Apply**.

Linking to Scalable Multicasts from a Web Page, Ram File, or SMIL File

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.

All links to scalable multicast content use the same format. Note that they always begin with `http://` and always ends with the `.sdp` extension:

`http://realserver.company.com:HTTPPort/MountPoint/virtual_path/filename.xxx.sdp.`

RealServer URL Components

Component	Meaning
<code>http</code>	The protocol used for streaming. Always use <code>http</code> in Web pages.
<code>realserver.company.com</code>	Machine and domain name of RealServer
<code>HTTPPort</code>	Port number where RealServer listens for requests sent via HTTP. This value is usually 80 or 8080; see “Port Variables” on page 39.
<code>MountPoint</code>	Scalable mountpoint, usually <code>scalable</code> .
<code>virtual_path</code>	The virtual path is any actual directory, relevant to the base path of the mount point. If the file is located in the base path itself, omit <code>virtual_path</code> .

(Table Page 1 of 2)

RealServer URL Components (continued)

Component	Meaning
<i>filename</i>	The file name itself.
xxx	The file type, such as ra, rm, or rt.
sdp	The final letters are required for scalable multicast. These are not part of the actual live file name; they only appear in the link.

(Table Page 2 of 2)

Using the example in the example for RTSP and PNA multicast, a link would look like the following:

```
<a href="http://realserver.company.com:8080/scalable/vivaldi.ra.sdp">
Click here to listen to today's Vivaldi selection</a>
```

Combining Splitting and Multicasting

To reach large audiences across a network that includes both Internet and intranet connections, use splitters to send data across the Internet to intranet sites, and then use multicast delivery within each target intranet. This can be a powerful method of distributing a live stream while still conserving bandwidth.

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 10: Authenticating RealServer Visitors.

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.

Additional Information

For information on editing the configuration file, see “Customizing RealServer” on page 35.

► To restrict access to HTML pages:

1. In RealSystem Administrator, click **General Setup**. Click **HTTP Delivery**.
2. To add a virtual directory, click **Add**. Type the name of the virtual path that contains material you want to be available for HTTP streaming.

Be sure that Admin and Ramgen are on this list; Admin refers to RealSystem Administrator, which is served via HTTP. Clips streamed with Ramgen may be requested in HTTP format. Also, the mount point used in scaleable multicast must be included; this value is usually scaleable. And push splitting uses HTTP for the initial connection conversation; add the push splitting mount point to this list, usually farm. If you are using Ram files, and they are stored on RealServer, add the virtual directory where they are stored to this list.

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.

3. Click **OK**. Click **Apply**.

Limiting Access by Number of Connections or Bandwidth

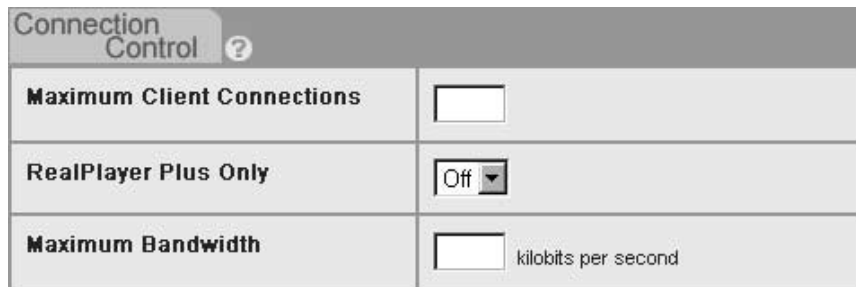
By using the **Maximum Clients** setting (the ClientConnections variable in the configuration file), you can limit the number of clients who connect simultaneously. Once this limit is reached, clients that attempt to connect receive an error message, and will not be able to connect until other clients 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**.



Connection Control ?	
Maximum Client Connections	<input type="text"/>
RealPlayer Plus Only	Off ▼
Maximum Bandwidth	<input type="text"/> kilobits per second

2. In the **Maximum Clients** 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 limiting 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.

3. When you have finished making changes, click **Apply**.

Limiting Access by RealPlayer Version

Two settings restrict access to all RealServer content, based on the client version. RealPlayer Plus Only means that only the RealNetworks RealPlayer Plus software can play presentations.

► **To limit access by player protocol number:**

This variable was used in earlier versions of RealServer and is included here for backwards compatibility. It must be added to the configuration file directly by using a text editor. It denies access to players whose version number is less than the number specified. Use one of the following values for Minimum Player Version:

- 0 All clients are permitted to connect to RealServer
- 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

► **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 **Delivery Only** to Yes in the Back-Channel multicast list, you can require that clients within a certain range of IP addresses connect only in multicast mode.

This feature is described in Chapter 8: Splitting and Multicasting on page 67.

Limiting Access Via IP Address

You can block or permit access to specific RealServer ports based on the IP address of the requesting machine.

For example, you can restrict which encoders can send encoded streams to your RealServer by restricting access to the encoding port (usually 4040).

Entire subnets can be restricted.

If a visitor clicks an URL for which they are denied access via this method, an error message appears in their client indicating that the URL is not valid.

A more selective form of restricting which material users can access (based on the directory or virtual directory where it's stored) is authentication, described in Chapter 10: Authenticating RealServer Visitors.

Setting Up IP Access Control

Add settings or edit the existing settings in the configuration file to limit which IP addresses can access your material, and which format they can use.

► **To limit access via IP number:**

1. In RealSystem Administrator, click **Security**. Click **Access Control**.
2. Click **Add**.
3. In the new browser which appears, type a name for the new access rule in the **Access Rule Name** box.
4. Indicate whether permission is being granted or refused by selecting **Allow** or **Deny** from the **Access** list.
5. In the **To** box, type the IP address of the RealServer machine or network card which is hosting the requested content. To refer to any IP address, type Any.
6. In the **From** box, type the IP address of the machine that is accessing RealServer. To restrict access from all IP addresses, type the word Any. This is the machine or range of addresses you want to restrict. Requests from the clients with the IP addresses in this variable are restricted in the method of content reception they can use.

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 bits for the subnet mask. For example, the following are equivalent and acceptable in the **From** box:

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.

7. Finally, list the RealServer port numbers to which you want to restrict access. In the **Restricted 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, HTTP port, and the port value used by the encoder.

8. Click **Add**. Click **Apply**.

RealServer authentication provides a way for you to control what or who can access your RealServer, whether it is an encoder sending a stream or a colleague perusing RealSystem Administrator or a user viewing content for which they've paid.

Overview

Authentication verifies the identity of a user or RealPlayer that makes a request for streamed media. 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:

- **Encoders**—Limiting which content creators can use their encoders to send live streams to RealServer.
- **RealSystem Administrators**—Allowing only certain administrators in your organization to use RealSystem Administrator.
- **Individual users**—Restricting which users can view certain content, both on-demand and live.

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 will identify 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 content.

Authenticating Encoder Connections

When a user sets up an encoder to send a stream to RealServer, you can require that she supply a user name and password. In this way, only authorized people can send streams to your RealServer.

Authenticating RealSystem Administrator Users

To protect your RealServer from changes made by unauthorized users, RealServer is installed with authentication turned on for RealSystem Administrator access. RealServer maintains a separate data store of user names and passwords of people who are authorized to make changes to RealServer via RealSystem Administrator.

Limiting User Access to Content

The most popular use of all is limiting user access to individual presentations or directories of clips.

Like the other methods, one database stores the names and passwords of the users who are authorized to view content. But an additional database can be used to list which content each user can view, and what type of access they have. The default method uses one database for all this information.

The different types of access to an individual clip include watching it a limited number of times, or watching it indefinitely while RealServer merely notes the number of viewings. Other methods are available; they are described in “Clip and Directory Permission Types” on page 106.

Two “levels” of authentication are a name and password requirement (user authentication), or a transparent type (player validation) that allows you to track visitor activity.

Additional Information

To limit visitors to RealServer via bandwidth, connection volume, client version, or IP address, use the methods described in Chapter 9: Limiting Access to RealServer.

Compatible Client Versions

RealPlayer versions 3.0 and earlier do not work with authentication and may display an error message. RealPlayer version 4.0 works with player validation only. RealPlayer version 5.0 and RealPlayer G2 support both player validation and user authentication.

Authentication Components

Authentication of encoders and RealSystem Administrator users has two components:

- **Realms**—authentication protocol to use in verifying user identity
- **Databases**—databases where names and passwords are stored

Authentication of content users—also known as the commerce feature—adds another piece:

- **Secure virtual paths**—URLs which should be authenticated

In addition, if you are using player validation, RealServer requires another list.

In the configuration file, 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 virtual 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 allows different types of content to share the same list of authorized users.

The components are covered in greater detail below.

Realms

A realm contains information about the type of authentication protocol and the database where the authenticated users' names will be stored. If you will be using Windows NT to authenticate users, the realm lists the type of NT authentication and the NT administrator-defined group name.

RealServer has three methods of authenticating the identity of visitors:

- Basic
- RealSystem 5.0
- Windows NT LAN Manager

Each realm can use only one authentication method.

If the clients that will be accessing content on your RealServer are RealPlayer version 5.0 and earlier, be sure to use the RealSystem 5.0 style for content authentication.

Authentication Protocols

PluginID Value	Authentication Protocol	Password Tool Used	Authenticates
rn-auth-basic	Basic	No	Encoders, RealSystem Administrator
rn-auth-rn5	RN5	Yes	Encoders, content
rn-auth-sspi	Windows NTLM Challenge/Response	No	RealSystem Administrator users, content (on intranets only)

Basic Authentication Protocol

The Basic Authentication protocol 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.

RN5

RN5 authentication is RealNetworks' own authentication protocol, developed for RealServer version 5.0.

If your material will be served to users working with RealPlayer version 5.0 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.

Using the Password Tool to Change Passwords Under RN5 Authentication

In RN5 authentication, RealServer stores all passwords in an encrypted format. Passwords can be entered and changed through the RealServer Administration page.

To manually change a user's password, the new password must first be encrypted using the **mkpnpass** tool with the Realm variable from the

appropriate list within AuthenticationRealms, then copied into the appropriate field in the authentication database or text file.

The password tool is a command line utility. It is located in the RealServer Bin directory.

► **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 G2 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
```

Windows NTLM Challenge/Response

For sites that use an NT-based security model, popular on corporate intranets, this method allows RealServer to use the existing NT database of user groups and permissions. It also allows access control of content via NTFS file permissions. The NTLM Authentication protocol uses Windows NT authentication.

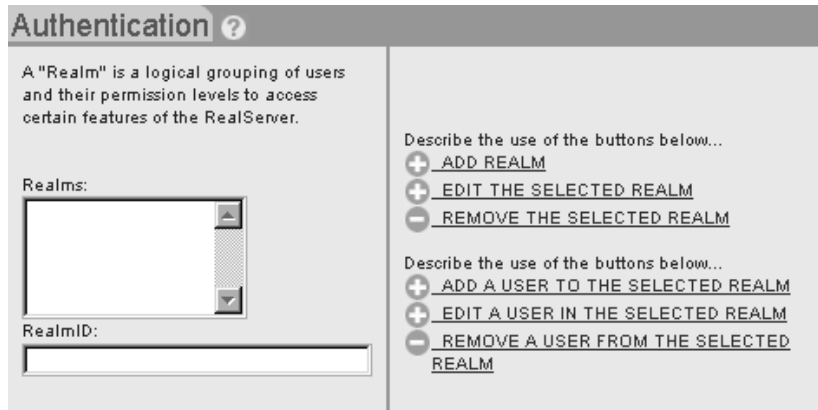
This method is only available to systems using Windows NT, and requires that RealServer itself be installed on an NT Server. For authenticating content, it also requires Microsoft Internet Explorer and RealNetworks RealPlayer.

Setting Up a Realm

Use the instructions below to create a realm.

► **To create a realm:**

1. In RealSystem Administrator, click **Security**. Click **Authentication**.



2. Click **Add Realm**. A new browser window appears.
3. Type a name for the realm in the **Realm ID** box.
4. In the **Realm** 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.
5. Choose the authentication method you want to use for this realm.
 - a. If you choose Basic or RN5, you will also need to select a database in which the names and passwords of authenticated users will be stored.

Additional Information

Information on setting up databases is in “Setting Up a Database” on page 102.

- b. If you choose Windows NT authentication, you do not need to select a database—instead, RealServer will use the NT list of names. Select the appropriate provider from the **Provider** list (there may be only one choice available). Type the Group name in the **Group** box.
6. Click **OK** to return to RealSystem Administrator.

Adding User Names to Realms

Use the following instructions to add to the list of authorized users in a particular realm.

Note

NTLM users must be managed using tools supplied by Windows NT.

► To add a user name to a realm:

1. In RealSystem Administrator, click **Security**. Click **Authentication**.
2. In the **Realms** list, select the name of the realm to which you want to add a user.
3. Click **Add**.
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. Click **Add**. A message appears; click **OK**.

Databases

The list of databases groups database interfaces and the locations of databases. RealServer includes three database interfaces:

- ODBC
- MySQL
- Text file

They are described in greater detail below.

Database Interfaces

The authentication package contains templates for common databases, including mSQL and common ODBC-compliant databases. 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.

Text File

The text file method is enabled during installation, as it allows the greatest insight into the access permission structure, but the text file method lacks the flexibility of a full database application.

Tip

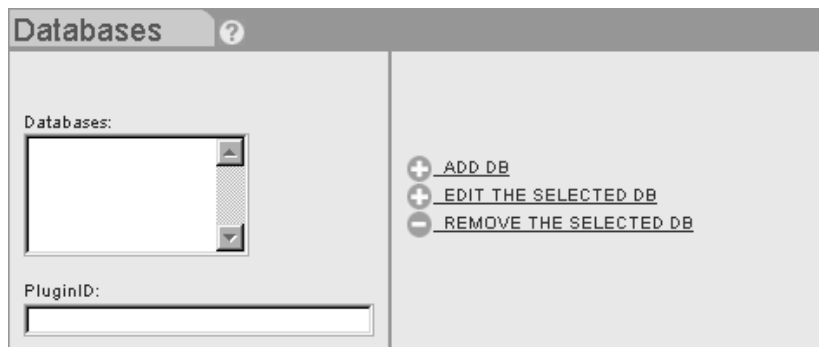
It's best to use the text file method only for simple tracking or for troubleshooting the system before linking a full-fledged database to RealServer. The text file method is also faster than a full-fledged database.

Setting Up a Database

Follow the instructions below to create the databases list.

► To set up a database:

1. In RealSystem Administrator, click **Security**. Click **Databases**.



2. Click **Add**.
3. In the new window that appears, type a description for the new database in the **Description** box.
4. From the **PluginID** list, select **rn-db-wrapper**.

Based on what you selected, more options may become available. If the database you're using is **rn-db-wrapper**, fill out the following boxes:

- a. In the **5.0 Plugin Path** box, type the location of the file you want to use. RealServer G2 includes the following files:

Function	Windows Name	UNIX Name
Basic	ppvb3260.dll	ppvbasic.so.6.0
ODBC	ppvo3260.dll	ppvodbc.so.6.0
MSQL	ppvm3260.dll	ppvmsql.so.6.0

- a. In the **Name** box, type the location where you want RealServer to store the database.
- b. In the **UserName** box, type the user name which the database requires.
- c. In the **Password** box, type the password which the database requires.
5. Click **Apply**. Click **Apply** again.

Secure Virtual Paths

The virtual path in the URL tells RealServer that this request should be authenticated before allowing access to the clip or presentation.

To protect access to content, you add the virtual path to the list of Authenticated Mount Points.

The links to on-demand authenticated content are just like the links to other on-demand content, with the addition of the Protected Virtual Path substituting for the virtual path.

The Protected Virtual Path refers to a virtual path. Consider the following directory structure:

```
(RealServer main directory)
    Content
        Speeches
            President
                Executives_only
```

In this example, if you want to authenticate the final directory on the list, `Executives_only`, add the following virtual path to the Protected Virtual Path list (assume that the main mount point is `/` and is defined as the RealServer Content directory):

```
/Speeches/President/Executives_only
```

Encoder User Authentication

When you install RealServer, you supply a user name and password. These are added to the administrator database and the encoder database. Users of RealSystem G2 encoders must supply this user name and password to connect. You can add or change user names and their password.

► **To use authentication for G2 encoders:**

1. In RealSystem Administrator, click **Broadcasting**. Click **G2 Encoders**.
2. Select the mount point for which you want to require authentication. In the **Encoder Authentication Realm** box, type the name of the 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 “Setting Up a Realm” on page 100.

3. Click **Apply**.
4. Next, add each user and assign a password; use the instructions in “Adding User Names to Realms” on page 101.

Older encoders can supply a password, and it must be the same for everyone. If you change the password, be sure to tell everyone who will be connecting what the new password is.

► **To use authentication for pre-G2 encoders:**

1. In RealSystem Administrator, click **Broadcasting**. Click **Pre-G2 Encoders**.
2. In the **Password** box, type the password which all pre-G2 encoders must supply in order to connect to RealServer.

During installation, you are prompted for a word to use for this password. If you change the value of the password here, be sure to inform everyone who is sending live data to you.

3. Click **Apply**.

RealSystem Administrator User Authentication

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.

► **To require authentication for RealSystem Administrator users:**

Authentication is enabled at installation. To turn it off, you must modify the configuration file directly. See Appendix B: Configuration File Contents.

To add other users to the database of names, use the instructions in “Adding User Names to Realms” on page 101.

Content User Authentication

There are several more options in setting up content authentication than for encoder or RealSystem Administrator user authentication.

To Use Passwords or Not?

To levels of verification are available: player validation requires a user name the first time the user registers. Thereafter, RealServer does not ask the user for a user name or password. The player ID is associated with the original user name, no matter who is using the player.

User authentication requests the user’s name and password each time the user clicks a link to secure material.

User Authentication

When you want to verify user identity before permitting access to a clip or directory, choose user authentication. With user authentication, it does not matter which computer a visitor uses to connect to the Web site. User authentication access privileges can be set by the administrator before the visitor views the secure media. User authentication is best suited to applications like pay-per-view, executive briefings, and distance learning.

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. This type authentication involves less viewer interaction, but each client must be registered individually by the viewer or central administrator. Player validation is the best way to track requests for specific types of material, such as fan clubs, premium groups, microcommerce, intranet, and demographic tracking.

► **To set up user authentication or player validation:**

Step 4 in “Creating Secure Virtual Paths” on page 109 gives instructions on choosing user authentication. This is done on a per-Protected Virtual Path basis.

Give Access to Everything or Specific Clips?

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 False, it allows general access to all authenticated users or players. When set to True, 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 directory, 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:**

On the **Commerce** page, clear the **Evaluate Permissions** box. This setting applies to the rules; you can use a different setting for each rule.

If you selected this box, you must 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.

Clip and Directory Permission Types

Access control features determine how long a user can view a particular presentation. These are indicated in the data storage. There are three types of access, discussed in greater detail below.

Permission Types

Name	Access to presentation or presentations in a directory	Permission Number
Event	Unlimited viewings of a given clip.	0
Calendar	Permission expires on a certain date.	1

(Table Page 1 of 2)

Permission Types (continued)

Name	Access to presentation or presentations in a directory	Permission Number
Duration	User gets a finite amount of time to view clips. All viewing time is deducted from this amount.	2
Credit	RealServer tracks how much time the user has spent viewing content.	3

A single RealServer can simultaneously deliver multiple types of access for different clips or directories of clips. (Table Page 2 of 2)

Event Access

In event access, the visitor is granted, in advance, unlimited access to one or more specific media clips.

Calendar Access

The process for expiration access follows that of event access, but permission is granted through a certain date (for example, unlimited viewing of any or all of some number of specified videos during the next week).

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 Access

In this type the user receives a fixed amount of viewing time (given in seconds) and RealServer subtracts all viewing time from this amount.

Credit Access

Like a taxi meter, this merely counts the number of times the user has played a presentation to which he has been given access. 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.

Changing Permission Types

► **To change permission types:**

1. In RealSystem Administrator, click **Security**. Click **Commerce**.
2. Click **Grant Permission**.
3. Follow the instructions on the page.

If you are using your own databases, you can modify them directly, without using RealSystem Administrator.

Note

Give only one type of access to a clip or directory. More than one type causes conflicts.

Creating Secure Virtual Paths

Identify which directories contain material to which you want to restrict access.

You can have multiple directories that contain secure material, and they can be in different physical locations.

► **To set up authentication for on-demand or live content:**

1. In RealSystem Administrator, click **Security**. Click **Commerce**.
2. In the **Commerce Rules** section, click **Add**.

The screenshot shows the 'Commerce' configuration window. On the left side, there is a 'Rules' list box, a 'Protected Path' text field, a 'Realm' text field, and three checkboxes: 'UsePlayerValidation', 'EvaluatePermissions', and 'AllowDuplicateIDs'. Below these are 'DatabaseID' and 'Registration Prefix' text fields. On the right side, there is a list of actions: '+ ADD RULE', '+ EDIT THE SELECTED RULE', '- REMOVE THE SELECTED RULE', a sub-header 'Manage user level permissions for granting and revoking access to content.', '+ GRANT PERMISSION TO A USER', '+ EDIT A PERMISSION', '- REVOKE PERMISSION FROM A USER', '- REVOKE ALL PERMISSIONS FROM A USER', '+ REDIRECT AN URL', and '- REMOVE A REDIRECTED URL'.

3. In the **Rules** box, type the name of the mount point or virtual directory for which you want to require authentication.

The default configuration creates one directory which contains all material to be authenticated, usually named **Secure**. If the secure directory contains subdirectories, append these to the mount point in Protected Virtual Path. For example, the subdirectory of the **Secure** directory called **Earnings** would be added to a Protected Virtual Path as `/Secure/Earnings/`. (Be sure you have added the single mount point as a Protected Virtual Path, or anything you put in the main secure directory will not be authenticated!)

4. Choose the level of authentication you want to use for this rule:
 - If you want this mount point to have user authentication, click **Realm** and select the database that will store the information.
 - If you want to use player validation, select the **UsePlayerValidation** box.
5. To allow access to all content in the virtual path, set **Evaluate Permissions** to **Off**. To look up permission information in the database, select **On**.
6. If you want a user to be able to log in at more than one location, set **Allow Duplicate IDs** to **True**. Otherwise, when this is set to **False**, and a user who tries to log in from a different location receives an error message, and must log out at the first location before he or she will be permitted to log in at the second location.
7. In the **DatabaseID** box, type the name of the database in which to store permission information.
8. If you will be using player validation, type the registration prefix in the **Registration Prefix** box. The word you use here must be unique—none of the registration prefixes that RealServer uses can be the same.
9. Click **Apply**. Click **Apply** again.

Allowing Users to Self-Register

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.

In its default state, RealServer requires that you add the names of users or clients to the appropriate databases before they can receive secure content.

This is feasible if you are administering RealServer over an intranet 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. Check the `readme.txt` file for more information.

Linking to Authenticated Content

Links to individual on-demand or live streams are the same as their non-authenticated counterparts. See “Linking to Files and Presentations” on page 20 for more information.

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.

Combining Authentication with Other RealServer Features

Authentication works with all other RealServer features. There are few special considerations for each feature, however.

Media Caches

Media cache software makes requests on behalf of clients, and caches the streams it receives. Although the media cache 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

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.

Splitting

If you are sending a stream to a RealServer which is acting as a splitter, you must put copies of all the databases that store authentication information on the splitter. This distributes the authentication load.

Splitting is described in Chapter 8: Splitting and Multicasting.

Multicasting

In both back-channel and scalable multicasts, the user or client is authenticated through the initial control channel connection. Be sure the multicast virtual path is on the list of Commerce Rules.

Both types of multicasting are described in Chapter 8: Splitting and Multicasting.

STORING AUTHENTICATION DATA

After a visitor has been granted access by the authentication feature, RealServer can check to see whether they have special permissions for viewing specific presentations or directories of presentations. You can use this information for applications such as pay-per-view.

Overview

The commerce feature works in conjunction with the authentication feature. It consists of an additional database which stores permissions to individual clips.

Working with the authentication feature, permission information is stored in a separate database.

After you've set up authentication, use the RealSystem Administrator to modify access for each user.

RealServer Data Storage

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.

Two methods are supplied with RealServer: text file and database. The text file storage method uses a combination of directory structure and text files to achieve a sensible data storage method. It is the default method. The database templates included with RealServer use a similar structure to the text file method, in a more familiar database format.

Using Text Files

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.

Note

If you manually edit the files, be sure that any blank (or unused) fields use an asterisk (*) and semi-colon (;) as a placeholder. Spaces are not allowed.

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;uuid;uuid_writeable
```

where:

<i>password</i>	When user authentication is in use, this stores the password. Otherwise shows an asterisk (*). Note: Passwords are encrypted. See “Using the Password Tool to Change Passwords Under RN5 Authentication” on page 98.
<i>uuid</i>	In player validation, stores playerID. In user authentication, an asterisk (*) appears in this field.
<i>uuid_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 106 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;*
```

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 version 3.0 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 <i>redirect.txt</i>).

Accesslog.txt

Each line of *accesslog.txt* describes the result of an attempt to view a clip.

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 "Permission Types" 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

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 121.

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 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 version 3.0 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 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 < textfilename`
Be sure to include the less-than sign (<).

MONITORING 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, whether viewers are being turned away. RealServer includes the following methods for monitoring real-time activity: G2 Java Monitor and NT Performance Monitor (for Windows NT users). To generate reports of historical activity, see Chapter 13: Reporting.

G2 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 one or more RealServers, making server management easy. It shows how the server is being used, who is using it, when it is most used, and which files are the most requested.

Use feedback from G2 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 G2 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.

Using G2 Java Monitor

There are several ways you can control what G2 Java Monitor displays. This section describes the commands present on the G2 Java Monitor display area and their functions.

► **To start G2 Java Monitor:**

In RealSystem Administrator, click **Monitor**. G2 Java Monitor appears.

Options Menu

Select the drop-down **Options** menu in the upper left hand corner of G2 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. G2 Java Monitor continues to receive data, but the graphical display of data does not change. Click Unpause 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 collapse the tab information and show only the tab name, leaving more room for the monitor. To show the contents of the tabs 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 Splitters.
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.

G2 Java Monitor Modes

The G2 Java Monitor can run as an applet or application. When you select **New Window** from the **Options** menu, the new G2 Java Monitor runs as an applet. Another method is available for running G2 Java Monitor as a separate application.

Review the considerations below before choosing which mode you want the G2 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 G2 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

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 G2 Java Monitor in applet mode:**

Applet mode is the default method for G2 Java Monitor when you click **New Window** from the **Options** menu.

► **To run G2 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/products/jdk/>.
Follow the installation instructions on the Web site.
2. Once you have installed the Java Development Kit on your system, change to the directory where the newly installed Java class files are located.
3. At a system prompt, type the following:
Java Monitor
The Monitor and a logon screen appear.
4. In the logon screen, type the RealServer name, port number, and password. Click **OK**.
5. G2 Java Monitor starts.

Configuring G2 Java Monitor Settings

G2 Java Monitor uses just two variables from RealServer.

► **To change G2 Java Monitor Settings:**

Indicate which port G2 Java Monitor should use in connecting to RealServer by setting the `MonitorPort` variable. The default value for `MonitorPort` is 9090.

The password which G2 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.

These values must be changed by directly modifying the configuration file. See Appendix B: Configuration File Contents for instructions.

Using 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 splitters 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 are 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 Performance Monitor.

Chapter 13

REPORTING

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.

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. At installation, Logging Style is set to 3 and Stats Mask is 0. 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.

Additional Information

Read about customizing RealServer settings in “Customizing RealServer Features”.

Once you know the values of these two variables, view the access log by opening `raccess.log` (Windows) or `raccess` (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 116.

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_id] [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.								
<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]								
<i>"GET filename</i>	File name (and path) requested by the client. Path is relative to BasePath in local file system. If the client requests a file that doesn't exist, UNKNOWN appears in place of a file name.								
<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: <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">(blank)</td> <td>UDP connection</td> </tr> <tr> <td>T</td> <td>TCP connection</td> </tr> <tr> <td>H</td> <td>HTTP connection</td> </tr> <tr> <td>M</td> <td>Multicast</td> </tr> </table> <p>For example, PNAT means that the clip was sent using the PNA protocol over a TCP connection.</p> <p>The version number indicates the edition of the protocol.</p>	(blank)	UDP connection	T	TCP connection	H	HTTP connection	M	Multicast
(blank)	UDP connection								
T	TCP connection								
H	HTTP connection								
M	Multicast								
<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 1 of 4)

Access Log Format (continued)

Access Log Field	Description																
[<i>client_info</i>]	Describes the version and type of client being used. Client information appears in the following format, [<i>platform version client type dist_code language CPU</i>]																
	<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>dist_code</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. For example: Win95_4.0_3.0.0.19_play32_PN01_EN_586</td> </tr> </tbody> </table>	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>dist_code</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. For example: Win95_4.0_3.0.0.19_play32_PN01_EN_586
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>dist_code</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. For example: Win95_4.0_3.0.0.19_play32_PN01_EN_586																
	RealAudio Player version 1.0 shows only two fields for [<i>client_info</i>]. They are <i>platform</i> and <i>client</i> .																
<i>client_id</i>	Unique ID generated during RealPlayer installation that enables you to track details for individual clients. Included when Logging Style is set to 2 or higher.																
[Stat1] (see the “Statistics Type 1 Information” table below)	Connection statistics sent by the client when it completes playing a clip. When the client blocks connection statistics, or when the RealServer is a splitter, 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 below)	Extended connection statistics sent by the client when it completes playing a clip. When the client blocks connection statistics, or when the client is a splitter, 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.																

(Table Page 2 of 4)

Access Log Format (continued)

Access Log Field	Description
[Stat3] (see the “Statistics Type 3 Information” table below)	Actions taken by the visitor while playing the clip. 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. 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 resent because of transmission errors. Included when Logging Style is set to 1 or higher.
<i>failed_resends</i>	Number of packets not successfully resent in time to correct transmission errors. Included when Logging Style is set to 1 or higher.
<i>stream_components</i>	Type of material sent, indicated in the following pattern: RealAudio RealVideo Event RealImage 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 RealImages would appear in the access log as: 1 1 0 0. Included when Logging Style is set to 3 or higher.
<i>start_time</i>	Timestamp of start time. Included when Logging Style is set to 3 or higher.
<i>server_address</i>	IP address of RealServer supplying the clip. If a splitter is in use, the receive splitter will be indicated here, rather than the source splitter’s name. Included when Logging Style is set to 3 or higher.
<i>average_bitrate</i>	Average bitrate of clip. Included when Logging Style is set to 4 or higher.

(Table Page 3 of 4)

Access Log Format (continued)

Access Log Field	Description
<i>packets_sent</i>	Number of packets sent. Included when Logging Style is set to 4 or higher.
<i>presentation_id</i>	Number used by other clips in a SMIL presentation. All elements from the same presentation use the same number. The SMIL file itself is also included in the log, and shares the number as well. The number is assigned by RealServer at the time of transmission. Included when Logging Style is 5.

(Table Page 4 of 4)**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</i> - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_id] [StatsMask results]
1	<i>client_IP_address</i> - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_id] [StatsMask results] file_size file_time sent_time resends failed_resends
2	<i>client_IP_address</i> - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_id] [StatsMask results] file_size file_time sent_time resends failed_resends
3	<i>client_IP_address</i> - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_id] [StatsMask results] file_size file_time sent_time resends failed_resends stream_components start_time server_address
4	<i>client_IP_address</i> - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_id] [StatsMask results] file_size file_time sent_time resends failed_resends stream_components start_time server_address average_bitrate packets_sent
5	<i>client_IP_address</i> - - [timestamp] "GET filename protocol/version" HTTP_status_code bytes_sent [client_info] [client_id] [StatsMask results] file_size file_time sent_time resends failed_resends stream_components start_time server_address average_bitrate packets_sent presentation_id

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 5.0 formats dnet RealAudio 3.0 formats 28.8 RealAudio 2.0 28.8 format lpcJ RealAudio 2.0 14.4 format cook RealAudio G2 format

Stat2 Syntax

Statistics Type 2 provide details about the success of clip delivery, giving information about bandwidth requests. Resent packets are described in detail here. 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.
<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 resent packets received by the client.
<i>late</i>	Number of resent 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.

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>	The visitor's or client's behavior, where values are the following:
ABORT	Abnormal client stop (not the natural end of clip play).
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.
RECEM	RealPlayer Plus stopped recording the clip.

Customizing Information Reported by the Access Log

To gather information with the access log, you must first decide what types of information you want to gather. Then make the appropriate changes to Logging Style and Stats Mask.

Information about RealServer activity is collected by Logging Style, and Stats Mask gathers statistics about what arrived at the client and viewer behavior while playing the clips.

Placing the Access Log

At installation, RealServer is configured to place log files in the Logs subdirectory of the main RealServer directory, Logging Style is set to 3, and Stats Mask is set to 0.

► **To indicate where to store the access log file:**

1. In RealSystem Administrator, click **General Setup**. Click **Logging**.

Access Log ?	
Logging Style	<input type="text" value="0"/>
Stats Mask	<input type="text" value="0"/>
Log Rolling Frequency	<input type="text"/>
Log Rolling Interval	<input type="text"/> Enter a Number
Log Rolling Size	When file reaches <input type="text"/> MB
Access Log Path	<input type="text"/> The access log file name will be raccess.log.yymmddhhmmss, where yymmdd is the year, month, and day, while hhmmss is the hour, minute, and second the access log file was created.

2. Type the name you want to use in the **Access Log Path** box. The default name of the access log file is raccess.log (Windows) or raccess (UNIX), and it is usually placed in the Logs subdirectory of the main RealServer directory. The directory (if any) typed here can be absolute or relative to the base path of the main mount point.

The name of the access file will be different if Log File Rolling is enabled; see “Log File Rolling” on page 143.

3. When you are finished, click **Apply**.

If **Access Log Path** is blank, RealServer records access information in the `rmaccess.log` or `rmaccess` file located in the same directory as the RealServer executable file.

Gathering Information with Logging Style

To configure RealServer to collect access information, configure Logging Style. There are six options, styles 0 through 5. Each logging style 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. If you omit this variable, RealServer uses the default style of 0.

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 132.

Information Collected by Logging Style

To gather this information...	...set LoggingStyle to this value
Bytes sent	0
Clip name including path	0
Client IP address and platform information	0
Timestamp	0
File size (in bytes)	1
File time (total file length in seconds)	1
Packets successfully and unsuccessfully resent	1
Protocol (RTSP or PNA)	1
Send time (total media sent in seconds)	1
Transport method (TCP, UDP) and version	1
Client ID	2
Server IP Address	3
Stream components	3
Timestamp for start time	3
Average bitrate	4

(Table Page 1 of 2)

Information Collected by Logging Style

To gather this information...	...set LoggingStyle to this value
Packets sent	4
Common presentation identifier	5

(Table Page 2 of 2)

Gathering Information 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 137, the “Statistics Type 2 Information” table on page 138, and the “Statistics Type 3 Information” table on page 139.

Collecting Combinations of Stats Mask Information

To gather this information...	...set Stats Mask to this value	Statistics Type 1	Statistics Type 2	Statistics Type 3
No additional statistics	0			
Statistics type 1 only	1	•		
Statistics type 2 only	2		•	
Both statistics types 1 and 2	3	•	•	
Statistics type 3 only	4			•
Both statistics types 1 and 3	5	•		•
Both statistics types 2 and 3	6		•	•
All statistics (types 1, 2, and 3)	7	•	•	•

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.

Not all versions of RealPlayer supply the information requested by Stats Mask; Statistics type 2 is supplied by RealAudio Player versions 3.0 and later, and Statistics type 3 is supplied by RealPlayer versions 5.0 and later.

Log File Rolling

Access 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.

Log file rolling applies only to access log files.

► **To set up log file rolling:**

1. In RealSystem Administrator, click **General Setup**. Click **Logging**.
2. Indicate where log files should be stored by giving the path and file name in the **Access Log Path** box. This is described in “Placing the Access Log” on page 140.
3. Decide whether to limit the log files by time period or by size.

- To limit by time period, choose the period from the **Log Rolling Frequency** list. You can save by the hour, day, week, or month.

In the **Log Rolling Interval** box, type the number of time periods. For example, if you chose **Days** from the **Log Rolling Frequency** list and typed 4 in the Log Rolling Interval box, RealServer will start a new access log every 4 days.

- To limit by file size, type a number in the **Log Rolling Size** box. Specify the size in megabytes.

If you have values in all three boxes, RealServer will use the size or time period that is reached first.

4. When you're done, click **Apply**.

Rolled log files are named with the following format:

name.log.datestamp

where:

name Name of the regular log file. The name for access logs is taken from the LogPath setting (usually raccess).

log The log file extension.

datestamp The date stamp, in the following format:

YYYYMMDDHHMMSS

where:

YYYY Year.

<i>MM</i>	Two digits of 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

Choose **Never** from the **Log Rolling Time Period** list, and type 0 (zero) for the **Log Rolling** size.

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.

Error messages relating to RealServer activity appear in the error log. The error log is created when the first error occurs.

For a list of error messages that can appear in this file and what to do about them, visit the RealNetworks technical support page at <http://service.real.com>. If you have an entry that refers to a fatal error, contact the RealNetworks Technical Support Department for assistance.

► **To customize where RealServer creates the error log:**

1. In RealSystem Administrator, click **General Setup**. Click **Logging**.
2. In the **Error Log Path** box, type the path and name you want to use for the error log. The default location is the Logs directory of the main RealServer directory, and the default file name is `rmerror.log`.
3. When you have finished making changes, click **Apply**.

Error Log 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
<code>servername(process_ID)</code>	The server name, followed by the process ID in parentheses.
<code>error_message</code>	Text of error message

Example Error Log

A sample error message looks like this:

```
***15-Nov-96 14:13:30.488 myserver(1556): 6220: No such user: joe
```

Cached Requests Log

Whenever RealServer sends a stream, it records that information in the access log. In addition, if RealServer sends a stream to a media cache, it creates an entry in the `ts.log` file. Requests that will be stored in media caches are identified by the port number to which they send the request.

Reading a Cached Requests Log

The entries in the `ts.log` file use one of two formats.

Note

As with other log files, the brackets within the `ts.log` file always appear and do not indicate optional material.

General Information Format

[day month date time year] message

where:

day is a three-letter abbreviation, such as Thu for Thursday

month is a three-letter abbreviation

date is the one or two-digit date

time is in twenty-four hour format: hours:minutes:seconds

Clips Served Format

[day month date time year] IP_address path_filename

where *IP_address* and *path_filename* refer to the stored location of the content.

Setting Up Cache Request Logging

Configure RealServer to create records of content requested for caches.

► To set up the media cache log file:

1. In RealSystem Administrator, click **Cache**. Click again on the word **Cache** that appears below.
2. From the **Cache Log** list, select **Enabled**.
3. In **Cache Log Path**, give the path and filename where the cached requests log file should be created. The default location is the Logs directory, and the default name is `ts.log`.
4. Click **Apply**.

Disabling Cache Request Logging

To disable the log file of cache requests, change **Cache Log** to Disabled.

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 G2 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).

A graphic for Appendix B featuring the word "Appendix" in a bold, italicized font, slanted upwards to the right. To its right is a large, bold, black letter "B". The background consists of several thin, light gray lines that create a sense of depth and perspective, resembling a grid or a set of converging lines.

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.

If you are going to modify the configuration file directly, please read the following sections:

- **Appendix A: Configuration File Syntax**—explains the structure of this file
- **“Editing the Configuration File with a Text Editor” in Chapter 4**—contains instructions on modifying the configuration file with a text editor

It is recommended that you first use RealSystem Administrator to make changes, and then examine the configuration file to learn how changes are made. Noticing how lists are created and changed will be especially instructive.

Warning

Exit RealSystem Administrator before opening the configuration file with a text editor or unexpected changes may result.

Elements of the Configuration File

Settings are grouped into like categories. Variables that are not part of lists can appear anywhere in the configuration file, but are grouped here for clarity.

Most configuration file variables closely match names in RealSystem Administrator. Differences are noted here.

Ports

Port settings for RTSPPort, PNAPort, and HTTPPort are described in Chapter 4: Customizing RealServer Features. MonitorPort is described in Chapter 12: Monitoring Activity.

<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 G2 Java Monitor) connect to RealServer.
<code><Var AdminPort="7845"/></code>	Port number for RealSystem Administrator connection.

Paths

LogPath and ErrorLogPath are described in Chapter 13: Reporting. PIDPath is described in Chapter 5: Advanced Features. PluginDirectory is described on Chapter 4: Customizing RealServer Features. LicenseDirectory is given on Chapter 3: Starting and Stopping RealServer.

Windows NT Variables

Path variables, along with typical paths used in Windows NT, are shown here.

<code><Var LogPath="C:\Program Files \Real\RealServer\Logs\rmaccess.log"/></code>	LogPath indicates where and with what name the access log 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.

```
<Var PluginDirectory="C:\Program Files Shows where the plug-in files are stored.
Real\RealServer\Plugins"/>
<Var LicenseDirectory="C:\Program File Gives the location of the license files.
\Real\RealServer\License"/>
```

UNIX

One additional setting is found on RealServer running on a UNIX system: PIDPath.

```
<Var LogPath="/usr/bin/RealServer LogPath indicates where and with what
/Logs/rmaccess.log"/> name the access log will be stored. If
omitted, RealServer places rmaccess.log in
the Logs directory.

<Var ErrorLogPath="/usr/bin/RealServer ErrorLogPath gives the path and name of the
/Logs/rmerror.log"/> error log file. If this setting is omitted,
RealServer places rmerror.log in the Logs
directory.

<Var PluginDirectory="/usr/bin Shows where the plug-in files are stored.
/RealServer/Plugins"/>

<Var LicenseDirectory="/usr/bin Gives the location of the license files.
/RealServer/License"/>

<Var PidPath="/usr/bin/RealServer In UNIX systems, the location of the process
/Logs/rmserver.pid"/> id file.
```

Passwords

MonitorPassword is described in Chapter 12: Monitoring Activity.

```
<Var MonitorPassword="letmein"/> Password used by G2 Java Monitor in
connecting to RealServer.
```

MIME Types

Setting up RealServer to send correct MIME type information with clips is described in Chapter 3: Starting and Stopping RealServer.

```
<List Name="MimeTypes">
  <List Name="audio/x-pn-realaudio">
    <Var Extension_01="ra"/>
    <Var Extension_02="ram"/>
  </List>
  <List Name="application/x-pn-realmedia">
```

```
<Var Extension_01="rm"/>
</List>
<List Name="application/x-pn-realmedia">
  <Var Extension_01="rt"/>
</List>
<List Name="application/x-pn-realmedia">
  <Var Extension_01="rp"/>
</List>
<List Name="application/x-pn-realmedia">
  <Var Extension_01="smi"/>
</List>
<List Name="application/sdp">
  <Var Extension_01="smi"/>
</List>
<List Name="text/html">
  <Var Extension_01="html"/>
  <Var Extension_02="htm"/>
</List>
<List Name="image/gif">
  <Var Extension_01="gif"/>
</List>
<List Name="image/jpg">
  <Var Extension_01="jpg"/>
  <Var Extension_02="jpeg"/>
</List>
</List>
```

Caching

This section allows media caches to request and cache streams on behalf of clients. Caching is described in Chapter 5: 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.

<code><Var TSEnable="True"/></code>	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 .)
<code><Var TSPort="7802"/></code>	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 .)
<code><Var TSLog="1"/></code>	Turns on the log of requests made by media caches. (In RealSystem Administrator this is changed with Cache Log .)
<code><Var TSLogPath="C:\Program Files\Real\RealServer\Logs\cache.log"/></code>	Path and file name of cache request log. The default location is the Logs directory, and the default name is ts.log. (In RealSystem Administrator this is changed with Cache Log Path .)
<code><List Name="NoCacheDir"></code>	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.
<code><Var Directory_01="/nocache1"/></code>	For each directory you want to restrict, create another Directory variable. Each directory must start and end with a forward slash.
<code><Var Directory_02="/nocache2"/></code>	
<code></List></code>	

IP Binding

The ability to reserve specific addresses for RealServer's use is explained in Chapter 5: 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 IPBinding list, RealServer binds to the host IP address and localhost. It does not bind to any others.

```
<List Name="IPBinding">
```

```
<Var Address_01="0.0.0.0"/>
```

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.

```
</List>
```

Live Archiving

The live archive feature is described in Chapter 7: Broadcasting 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 (*).

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 FileSize, and FileTime, and BandwidthNegotiation.

```
<List Name="LiveArchive">
```

```
<List Name="*">
```

An asterisk for a list name indicates the main content directory.

```
<Var TargetDirectory="/Archive"/>
```

The virtual directory where RealServer will create the archive files.

```
<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 5.0-style bandwidth negotiation is in use.

```
</List>
```

<pre><List Name="concerts"> <Var TargetDirectory="/Archive"/> <Var FileTime=1h"/></pre>	<p>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 FileTime. RealServer requires that the units be in the mhd order, so if you specify a subset, be sure to use the correct order.</p>
<pre></List> <List Name="secure"> <Var NoArchive="True"/></pre>	<p>When set to True, disables archiving of live files for the given directory.</p>
<pre></List> </List></pre>	

Allowance

Settings in this section refer to the allowance plug-in. They are described in Chapter 9: 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 On, ValidPlayerOnly sends a message to any clients other than RealNetworks RealPlayer version 5.0 or RealNetworks RealPlayer G2 directing them to upgrade to the latest version of RealPlayer. If set to Off, all clients can receive all clips. In Basic Server and Basic Server Plus, this is set to On and cannot be changed.

<pre><Var ClientConnections="25"/></pre>	<p>Limits the number of connections 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.</p>
<pre><Var MaxBandwidth="64"/></pre>	<p>Limits the amount of bandwidth in use by RealServer. The value is given in kilobits per second.</p>

<Var ValidPlayersOnly="True"/>	Allows only RealPlayer version 5.0 and RealPlayer G2 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 On and cannot be changed.
<Var MinPlayerVersion="2"/>	Sets the minimum RealPlayer version that can access the content. To limit to version 2.0 and later, set MinPlayerVersion to 2, and so on. To allow only RealPlayer G2, set it to 6.
<Var MinPlayerProtocol="0"/>	Limits access by protocol number. Use one of the following values for MinPlayerVersion: 1 RealAudio Player version 1.0 2 RealAudio Player version 2.0 3 RealAudio Player version 3.0 4 RealPlayer version 4.0 5 RealPlayer version 5.0 6 RealPlayer G2
<Var PlusOnly="False"/>	When set to True, PlusOnly allows only RealPlayer Plus to play content.

HTTP Support

This feature, which indicates the virtual directories whose content can be streamed via HTTP, is explained in [Chapter 9: Limiting Access to RealServer](#). Each Path variable gives the name of a virtual directory whose content can be streamed via HTTP.

Be sure that Admin and Ramgen are on this list; Admin refers to RealSystem Administrator, which is served via HTTP. Clips streamed with Ramgen may be requested in HTTP format. Also, the mount point used in the Scalable Multicast list must be included; this value is usually scalable. And push splitting uses HTTP for the initial connection conversation; add the push splitting mount point to this list, usually farm.

```
<List Name="HTTPDeliverable">
```

```

<Var Path_01="/admin"/>
<Var Path_02="/localadmins"/>
<Var Path_03="/ramgen"/>
<Var Path_04="/farm"/>
<Var Path_05="/scalable"/>
</List>

```

Each Path variable gives the name of a mount point, directory or virtual directory whose content can be streamed via HTTP.

Access Control

Restricting access to RealServer content via the requesting client's IP address is described in Chapter 9: 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.

```

<List Name="AccessControl">
  <List Name="100">
    <Var Access="Allow"/>
    <Var Transport="TCP"/>
    <Var To="127.0.0.1"/>
  </List>
</List>

```

Whether access is allowed or denied: set to Allow or Deny.

Transmission method being accessed. TCP is the only option for this list.

Address of the host RealServer or network card of hosting machine. Use specific address or Any.

<pre><Var From="any"/></pre>	<p>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 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.</p>
<pre><List Name="Ports"> <Var Port_01="554"/> <Var Port_02="4040"/> <Var Port_03="5050"/> <Var Port_04="7070"/> <Var Port_05="8080"/> <Var Port_06="9090"/> </List> </List> </List></pre>	<p>List of ports to which access is restricted. 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.</p>

File Systems

The FSMount section of the configuration file 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 G2 SDK Developer's Guide provides developers with the public interfaces used to extend and customize RealSystem G2 to stream new datatypes, create new

clients, or to customize RealServer by building a new plug-in.

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.

RealNetworks Plug-in Names

ShortName	Windows Filename	UNIX Filename	Description
pn-local	smp13260.dll	smp1fsys.so.6.0	Local File System
pn-admin	admi3260.dll	adminfs.so.6.0	Admin File System
pn-encoder	enco3260.dll	encoplin.so.6.0	G2 Live Encoder Plug-in
pn-live3	liv33260.dll	liv3plin.so.6.0	Pre-G2 Live Encoder Plug-in
pn-ramgen	ramp3260.dll	ramplin.so.6.0	Ramgen File System
pn-splitter	pull3260.dll	pullplin.so.6.0	Pull Splitting File System
pn-farmsplit	push3260.dll	pushplin.so.6.0	Push Splitting File System
pn-scalable	pply3260.dll	pplyplin.so.6.0	Scalable Multicasting File System

Local File System

The local file system, which handles requests for nearly all streamed media content, is described in Chapter 2: Overview.

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.

```
<List Name="RealContent">                               Identifies this list as the main content list.
  <Var ShortName="pn-local"/>                           The short name indicates which file system
                                                         handles requests directed to this mount
                                                         point.
```

```
<Var MountPoint="/" />
```

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.

```
<Var BasePath="C:\Program
Files\Real\RealServer\
Content" />
```

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.

```
</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 4: Customizing RealServer Features.

```
<ListName="RealAdministrator_Files">
```

```
<Var ShortName="pn-admin">
```

RealSystem Administrator uses the pn-admin plugin.

```
<Var MountPoint="/admin/" />
```

The default value for MountPoint is /admin/. If you change this, you will need to type a new URL to connect to RealSystem Administrator.

```
<Var BaseMountPoint=
"/localadmin/" />
```

This special form of mount point reflects the mount point of the RealAdministrator list.

<Var Realm="AdminRealm"/>	The Realm variable identifies which AuthenticationRealm settings will be used with requests sent to the RealSystem Administrator mount point.
</List>	
<List Name="RealAdministrator">	
<Var ShortName="pn-local"/>	RealSystem Administrator uses the local file system.
<Var MountPoint="/localadmin"/>	Mount point, used when RealAdministrator_Files list requests files from this plugin. The default value is /localadmin/. 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>	

Ramgen

Ramgen is described in Chapter 2: Overview and in *RealSystem G2 Production Guide*. There are only two variables associated with Ramgen: ShortName and MountPoint.

<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>	

Encoders

G2 Encoders

Receiving streams from both RealSystem G2 encoders and earlier versions are explained in Chapter 7: Broadcasting Presentations. These variables are used in this list: ShortName, MountPoint, Port, AssociatedMediaPath, and Realm.

Unlike other plug-ins, encoder lists cannot have multiple mount points.

<code><List Name="G2_Encoders"></code>	
<code><Var ShortName="pn-encoder"/></code>	Short name of G2 live encoder plugin. See “RealNetworks Plug-in Names” table for values.
<code><Var MountPoint="/encoder"/></code>	Portion of URL that indicates the type of request and therefore which file system will handle the request.
<code><Var Port="4040"/></code>	Port to which G2 encoders will send their live streams. Default value is 4040.
<code><Var AssociatedMediaPath="events"/></code>	Name of directory containing event files. A live stream whose name matches a file in this directory will merge its live streams with the events.
<code><Var EncoderRealm="EncoderRealm"/></code>	List of authentication protocols and databases. See AuthenticationRealms list.
<code></List></code>	

Pre-G2-Encoders

The list for encoders such as RealEncoder and RealPublisher versions 5.0 and earlier uses these variables: ShortName, MountPoint, Port, AssociatedMediaPath, Realm, and Password.

Unlike other plug-ins, encoder lists cannot have multiple mount points.

<code><List Name="Pre_G2_Encoders"></code>	
<code><Var ShortName="pn-live3"></code>	Short name of 5.0 and older live encoder plugin. See “RealNetworks Plug-in Names” table for values.
<code><Var MountPoint="/live"/></code>	Portion of URL that indicates the type of request and therefore which file system will handle the request.
<code><Var Port="5050"/></code>	Port to which older encoders will send their live streams. Default value is 5050.
<code><Var AssociatedMediaPath="events"/></code>	Name of directory containing event files. A live stream whose name matches a file in this directory will merge its live streams with the events.
<code><Var Password="letmein"/></code>	Password used by encoders to connect to RealServer.
<code></List></code>	

Splitting

Push Splitting

The push splitting method is described in Chapter 8: Splitting and Multicasting, beginning on page 70. Settings that go on both the source RealServer and the splitter are ShortName, MountPoint, Port, SplitterHostName, and the FarmSplitSources list.

The source uses the SplitterResendBuffer and SplitterSourceTimeout variables.

The splitter uses the SplitterSourceList with one or many sublists, and the SplitterBufferDelay, SplitterTimeout, SplitterSourceProbeInterval, and SupportPathDirectory variables.

You can create another push splitting list by copying the list structure and changing the name of the overall list from Splitter_Farm to something different.

<code><List Name="Splitter_Farm"></code>	Push splitting list.
<code><Var ShortName="pn-farmsplit"/></code>	The short name of pn-farmsplit indicates the plugin to use.
<code><Var MountPoint="/farm/"></code>	Mount point used in URLs.
<code><Var SplitterHostName="myhost.domain.com"/></code>	Domain and name of this RealServer.
<code><!-- source variables --></code>	The next two variables are necessary only on the source.
<code><List Name="FarmSplitSources"></code>	Identifies which directories will not be split.
<code><List Name="/live/concerts/"></code>	Each sublist of FarmSplitSources names a directory. To indicate all directories at once, use an asterisk (*) for the list name. The
<code><Var NoSplit="True"/></code>	NoSplit variable indicates whether the
<code></List></code>	directory will be split. To allow streams from
<code></List></code>	all directories to be split, set NoSplit to False (or set Split All Streams to Yes in RealSystem Administrator).
<code><Var SupportPathDirectory="c:\Program Files\Real\RealServer\Lib"/></code>	Gives the location of the encnet.dll (Windows) or encnet.so.6.0 (UNIX) file, usually the RealServer Lib directory.
<code><Var SplitterResendBuffer="60"/></code>	Size of the buffer for UDP resends, in seconds. Permitted values are from 0 to 32767.

<code><Var SplitterSourceTimeout="60"/></code>	Limits how many seconds the source RealServer will wait before sending data to a splitter that is not responding.
<code><!-- splitter variables --></code>	The following settings are necessary on the splitter only.
<code><Var Port="11002"/></code>	Port number on the receive splitter which will receive splitter connections.
<code><List Name="SplitterSourceList"></code>	List of source RealServers that this splitter should contact for live streams.
<code><List Name="Host1"></code>	Names each source RealServer from which this splitter will be splitting streams, one list per source.
<code><Var Address="host.domain.com"/></code>	Name or IP address of the RealServer to contact for streams.
<code><Var Port="8080"/></code>	Port number on the source RealServer to which this splitter will direct its probes.
<code></List></code>	
<code></List></code>	
<code><Var SplitterBufferDelay="60"/></code>	Seconds of data to store in the buffer, thus reducing dropouts over a splitter connection.
<code><Var SplitterTimeout="60"/></code>	Seconds a splitter will wait before considering a stream inactive. Range is from 0 to 32767.
<code><Var SplitterSourceProbeInterval="60"/></code>	Frequency with which the splitter requests a stream from a source. Given in seconds.
<code></List></code>	

Pull Splitting

The second splitting method, pull splitting, is described in Chapter 8: Splitting and Multicasting, beginning on page 75.

Only three variables appear in the pull splitting: ShortName, MountPoint, and Port. The source RealServer and the splitter have the same information in their Splitter_DoubleURL sections, but each system is interested in different

information: the RealServer looks at the Port value, and the splitter looks at the mount point.

```
<List Name="Splitter_DoubleURL">
  <Var ShortName="pn-splitter"/>
  <Var MountPoint="/split"/>
  <Var Port="3030"/>
</List>
```

Short name of the pull splitting plugin. Default is pn-splitter.

Mount point. Used in URLs that reference pull splitting streams. Default is /split/.

Port number to which the source RealServer will listen for pull splitting requests.

Multicasting

Two methods of multicasting are available: scalable and back-channel.

Scalable Multicasting

Scalable multicasting is described in Chapter 8: Splitting and Multicasting. Located within the FSMount list, scalable multicasting uses the following variables: ShortName, Enabled, HostAddress, MountPoint, PortRange, AddressRange, VirtualPath, and TTL.

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.

```
<List Name="FSMount">
...
  <List Name="Scalable Multicast">
    <Var ShortName="pn-scalable"/>
    <Var HostAddress="" />
    <Var MountPoint="/scalable"/>
  </List Name="Sources">
    <List Name="Concerts">
```

Gives the short name of the plugin file.

Indicates name of RealServer host computer.

Mount point used in all URLs for scalable multicasts.

Each list within this list represents a virtual directory that is to be streamed via scalable multicast.

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 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.
<code><Var PortRange="7300-7320"/></code>	Range of addresses to which RealServer can send a multicast stream. RealServer uses the first available address.
<code><Var TTL="16"/></code>	Time To Live for multicast packets travelling over the network.
<code></List></code>	
<code><List Name="Live Concerts"></code>	Name of this list.
<code><Var VirtualPath="Liveconcerts"/></code>	Virtual directory which will be available via scalable multicast.
<code><Var Enabled="True"/></code>	See description earlier in this section.
<code><Var AddressRange="231.1.1.1-231.1.1.10"/></code>	See description earlier in this section.
<code><Var PortRange="7300-7320"/></code>	See description earlier in this section.
<code><Var TTL="16"/></code>	See description earlier in this section.
<code></List></code>	
...	
<code></List></code>	

Back-Channel Multicasting

Unlike scalable multicasting, back-channel multicast settings are not located within the `FSMount` list. Back-channel multicasting, both RTSP and PNA methods, is described in Chapter 8: Splitting and Multicasting.

Settings used with this list are `AddressRange`, `DeliveryOnly`, `PNAPort`, `RTSPPort`, `Resend`, and `TTL`.

```
<List Name="Multicast">
```

```
  <Var AddressRange="" />
```

Range of addresses to which you want to send streams, in the form of *address-address*. 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.

```
  <List Name="ControlList">
```

The `ControlList` list gives the addresses of clients allowed to receive multicast transmissions.

```
    <Var Allow=  
      "164.16.2.24:255.0.0.0" />
```

Address and netmask, separated by a colon, of clients allowed to receive multicast transmissions. Uses same format as `From` variable in `AccessControl` list.

```
  </List>
```

```
  <Var DeliveryOnly="False" />
```

Requires clients listed in `ControlList` to receive only multicast transmissions from RealServer. When `DeliveryOnly` is `False`, clients on `ControlList` can receive both multicasts and unicasts.

```
  <Var PNAPort="7070" />
```

Port on client machines to which RealServer sends PNA streams. Default value is 7070.

```
  <Var RTSPPort="554" />
```

Port on client machines to which RealServer sends RTSP 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.

```
</List>
```

Authentication and Commerce

Authentication is described in Chapter 10: Authenticating RealServer Visitors.

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.

<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.
<List Name="RN5Authenticator">	User-defined description of authentication to use in this realm. Use only one type of authentication per realm.

<code><Var PluginID="rn-auth-rn5"/></code>	Plug-in which performs the authentication. For a list of options, see the “AuthenticationRealms PluginID Settings” table below.
<code><Var DatabaseID="Encoder_RN5"/></code>	Identifies which database to look in for authentication data. Refers to a list name within the Databases list.
<code></List></code>	
<code></List></code>	
<code><List Name="SecureContent"></code>	A realm.
<code><Var Realm="ContentRealm"/></code>	See description earlier in this section.
<code><List Name="NTLMAuthenticator"></code>	User-defined description of authentication to use in this realm. Use only one type of authentication per realm.
<code><Var PluginID="rn-auth-sspi"/></code>	Plug-in which performs the authentication. For a list of options, see the “AuthenticationRealms PluginID Settings” table below.
<code><Var Provider="NTLM"/></code>	See description earlier in this section.
<code></List></code>	
<code></List></code>	
<code></List></code>	

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 plugin 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.

```
<List Name="CommerceRules">
  <List Name="SecureContent">
    <Var ProtectedVirtualPath=
      "/secure"/>
    <Var Realm="ContentRealm"/>
    <Var EvaluatePermissions="True"/>
    <Var DatabaseID="Content_RN5"/>
    <Var AllowDuplicateIDs="False"/>
  </List>
  <List Name="SecureLiveContent">
    <Var ProtectedVirtualPath=
      "/encoder/secure"/>
    <Var UseGUIDValidation="True"/>
    <Var EvaluatePermissions="True"/>
    <Var DatabaseID="Content_RN5"/>
    <Var AllowDuplicateIDs="True"/>
```

Name of the mount path, virtual directory, or actual directory, used in URLs, that you want RealServer to authenticate.

This points to the realm names in AuthenticationRealms list. Sets up user authentication. Don't use if UseGUIDValidation is also in use.

Instructs RealServer whether or not to look at the permissions list, or to just allow access to all content.

Points to the database identifiers in the Databases list.

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.

Name of the mount path, virtual directory, or actual directory, used in URLs, that you want RealServer to authenticate.

Sets up player authentication. Gathers the client's ID, but not the user's name. Don't use if Realm is in use.

See description earlier in this section.

Points to the database identifiers in the Databases list.

See description earlier in this section.

```
</List>
```

```
</List>
```

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.

```
<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.

```
<Var DatabaseID="Content_RN5"/>
```

Name of database, from Databases list, that will store this type of data.

```
</List>
```

```
<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"/>
```

Name of database, from Databases list, that will store this type of data.

```
</List>
```

```
</List>
```

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 plugins with location information.

The options available to each sublist are PluginID, Path, DBName, DBLoginPassword, and DBLoginPassword. The last two are only required if the PathToDBPlugin is set to ppvm3260 or ppvo3260.

```
<List Name="Databases">
  <List Name="Admin_Basic">
    <Var PluginID="rn-db-flatfile"/> Name of plugin that will interact with the
    database. See "Databases PluginID Values"
    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.
  </List>
  <List Name="Encoder_RN5">
    <Var PluginID="rn-db-wrapper"/> Name of plugin that will interact with the
    database. See "Databases PluginID Values"
    table for a list of options.

    <Var Path="C:\Program Files
    \Real\RealServer\enc_r_db"/> Location where the database files are stored
    or will be stored.
  </List>
  <List Name="Content_RN5">
    <Var PluginID="rn-db-wrapper"/> Name of plugin that will interact with the
    database. See "Databases PluginID Values"
    table for a list of options.

    <Var PathToDBPlugin="C:\Program Files\Real\realserver\Plugins
    \ppvb3260.dll"/> Location of plugin file.

    <Var DBName="C:\Program Files
    \Real\RealServer\con_r_db"/> Location where the database files are stored
    or will be stored.

    <Var DBLoginUsername="amanda"/> User name required by database. (Optional.)

    <Var DBLoginPassword="q27df"/> Password required by database. (Optional.)
  </List>
</List>
```

Databases PluginID Values

PluginID Value	Data Store Method	Associated Variables
rn-db-wrapper	Interface to work with older protocols	PathToDBPlugin (required), DBName (required)

Logging

Logging and reporting features are described in Chapter 13: Reporting. Variables which control the locations of the access and error log files are described in “Paths” on page 152 of this chapter.

<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 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.
<code><Var LogRollSize="50"/></code>	Creates a new access log when the indicated file size is reached. See also <code>LogRollFrequency</code> . If you include both <code>LogRollFrequency</code> and <code>LogRollSize</code> , RealServer uses the variable it finds first.

Disable log file rolling by changing the `LogRollFrequency` and `LogRollSize` variables to 0.

CONFIGURATION FILE EQUIVALENTS

Earlier versions of RealServer used a different file format. Some of the configuration variables have different names or syntax in this version of RealServer.

If you are upgrading from a previous version of RealServer, it is recommended that you use RealSystem Administrator to customize your new RealServer, rather than editing the configuration file directly.

Additional Information

Instructions on customizing the configuration file can be found in “Customizing RealServer” on page 35.

RealServer Configuration File Equivalents

5.0 Variable	G2 List or Variable
AuthAllowDuplicateIDs	AllowDuplicateIDs variable in CommerceRules list.
AuthDBName	DBName in Databases list.
AuthDBPassword	DBLoginPassword in Databases list.
AuthDBPlugin	Not used.
AuthDBUserID	DBLoginUsername in Databases list.
AuthMode	Within CommerceRules list, use Realm variable to indicate user authentication, implement UseGUIDValidation variable to indicate player validation.
AuthPath	Not used.
AuthRegPrefix	GUIDRegistrationPrefix list.
BandwidthEncoding	Not used.
BasePath	BasePath variable in local file system section of FSMount list.
BindToAllInterfaces	No specific variable exists; instead, set address to 0.0.0.0 in IPBinding list.

(Table Page 1 of 4)

RealServer Configuration File Equivalents (continued)

5.0 Variable	G2 List or Variable
ClientConnections	ClientConnections variable.
ConnectControllList	AccessControl list.
CustomerName	License information is stored in license files. LicenseDirectory gives the location of the license files.
DefaultErrorFile	Not used. Instead, see “Playing A “Please Stand By...” Message” on page 60.
EncoderControllList	Done with AccessControl list.
EncoderPassword	Password in Pre_G2_Encoders list within FSMount list.
EncoderTimeout	Not used.
ErrorLogPath	ErrorLogPath variable.
Group	Group variable.
HTTPPort	HTTPPort variable.
InputFile	Not used.
IOBufferSize	Not used.
IPBindingList	IPBinding list.
LicenseKey	License information is stored in license files. LicenseDirectory gives the location of the license files.
LiveFileBandwidthNegotiation	BandwidthNegotiation variable in directory name section of LiveArchive list.
LiveFilePassword	See Chapter 10: Authenticating RealServer Visitors for new method of storing individual passwords for each encoder.
LiveFileSize	FileSize variable in <i>directory name</i> section of LiveArchive list.
LiveFileTarget	TargetDirectory variable in <i>directory name</i> section of LiveArchive list.
LiveFileTime	FileTime variable in <i>directory name</i> section of LiveArchive list.
LocalHost	Not used.
LoggingStyle	LoggingStyle variable.
LogPath	LogPath variable.

(Table Page 2 of 4)

RealServer Configuration File Equivalents (continued)

5.0 Variable	G2 List or Variable
MaxBandwidth	MaxBandwidth variable.
MaxThreads	Not used.
MinPlayerProtocol	MinPlayerProtocol variable or MinPlayerVersion variable.
MonitorConnections	MonitorConnections variable.
MonitorPassword	MonitorPassword variable.
MonitorPort	MonitorPort variable.
MulticastAddressRange	AddressRange variable in back-channel and scalable multicast lists.
MulticastControllist	Controllist list in RTSP or PNA lists within Multicast list.
MulticastDeliveryOnly	DeliveryOnly variable in RTSP or PNA lists within Multicast list.
MulticastPort	Port variable in RTSP or PNA lists within Multicast list.
MulticastTTL	TTL variable in RTSP, PNA, or Scalable list within Multicast list.
OutputFile	Not used.
PidPath	PidPath variable.
PnaPort	PnaPort variable.
Realm	See “Realms” in “Authenticating RealServer Visitors”.
ResolverPort	Not used.
RestoreOriginalPrivilege OnReload	RestoreOriginalPrivilegeOnReload variable.
ServerHost	Not used.
ServerPassword	Not used.
ServerPort	Not used.
SplitterAnnouncePort	Not used.
SplitterBufferDelay	SplitterBufferDelay variable.
SplitterControllist	Use the AccessControl list.
SplitterMaxResendPPS	Not used.
SplitterResendBuffer	Not used.
SplitterSourceList	SplitterSource list.

(Table Page 3 of 4)

RealServer Configuration File Equivalents (continued)

5.0 Variable	G2 List or Variable
SplitterSourceProbeInterval	SplitterSourceProbeInterval variable.
SplitterSourceTimeout	SplitterSourceTimeout variable.
SplitterTimeout	SplitterTimeout variable.
StatsMask	StatsMask variable.
Timeout	Not used.
URL	Not used.
User	User variable.
UserDir	Not used.
UserList	Not used.

(Table Page 4 of 4)



INDEX

- A**
- about *See* stopping RealServer
 - ABORT, in access log, 139
 - Access
 - in RealSystem Administrator, 93
 - variable, 159
 - Access Control
 - in RealSystem Administrator, 93
 - limiting splittter access, 70
 - list, 154, 159
 - access log, 81, 131, 143, 153
 - customizing, 140
 - format, 132, 136
 - reading, 133, 145
 - rolling, 143
 - Access Log Path
 - in RealSystem Administrator, 140, 143
 - Access Rule Name
 - in RealSystem Administrator, 93
 - access_log table, 118, 121
 - accesslog.txt, 114, 116, 117
 - Address
 - in RealSystem Administrator, 12, 132
 - variable, 166
 - Address Range
 - in RealSystem Administrator, 83, 87
 - variable, 167, 168, 169
 - Address_01 variable, 155, 156
 - Admin Port
 - in RealSystem Administrator, 40
 - variable, 152
 - adminfs mount point, 161, 162
 - alerts, in NT performance monitor, 129
 - Allow Duplicate IDs
 - variable, 172
 - Allow variable, 169
 - allowance plugin, 157
 - announcement file *See* SMIL file
 - archiving live broadcasts, 61
 - Associated Media Path
 - in RealSystem Administrator, 58, 59
 - variable, 163, 164
 - Auth Allow Duplicate IDs
 - variable, 177
 - Auth Reg Prefix
 - in RealSystem Administrator, 177
 - variable, 177
 - AuthDBName variable, 177
 - AuthDBPassword variable, 177
 - AuthDBPlugin variable, 177
 - AuthDBUserID variable, 177
 - Authentication
 - in RealSystem Administrator, 170
 - variable, 162
 - authentication, 95
 - in G2SLTA, 66
 - in license, 32, 33
 - of content users, 96
 - of encoder users, 96, 104
 - of RealSystem Administrator users, 96, 104
 - telling content creators, 24
 - Authentication Realm
 - in RealSystem Administrator, 60
 - variable, 170, 172
 - AuthMode variable, 177
 - Authorized_User_Group variable, 162
 - AuthPath variable, 177
 - AuthRegPrefix variable, 177
- B**
- back-channel multicasting, 78
 - Bandwidth Negotiation
 - in RealSystem Administrator, 63, 64

- variable, 63, 64, 156
 - bandwidth negotiation, 55
 - bandwidth, limiting, 90
 - BandwidthEncoding variable, 177
 - Base Path
 - described, 16
 - in local file system, 161
 - in RealSystem Administrator, 40, 41
 - variable, 162, 177
 - in RealAdministrator list, 163
 - BaseMountPoint
 - variable, 162, 163
 - BindToAllInterfaces variable, 177
- C**
- Cache Log
 - in RealSystem Administrator, 146, 155
 - cache log, 146
 - Cache Log Path, 146, 155
 - in RealSystem Administrator, 146
 - Cache Port, 44, 155
 - Cache Requests, 44, 45, 46, 155
 - chart *See* graph
 - CLICK, in access log, 139
 - Client Connections
 - variable, 90, 157, 178
 - ClientConnections
 - variable, 157
 - clustering *See* splitting
 - comment tag, 147
 - configuration file
 - components, 147
 - editing with text editor, 37
 - starting with, 29
 - ConnectControlList variable, 178
 - connectionless multicasting, 78
 - setting up, 85
 - connections, limiting, 90
 - CustomerName variable, 178
- D**
- Database ID
 - variable, 171, 172, 173
 - Databases
 - in RealSystem Administrator, 102
- DB Name
 - variable, 174
 - DBLoginPassword
 - variable, 174
 - DBLoginUsername
 - variable, 174
 - DBName
 - variable, 174
 - DefaultErrorFile variable, 178
 - Delivery Only
 - in RealSystem Administrator, 84, 92
 - variable, 169
 - De-Militarized Zone (DMZ) *See* firewalls, 49
 - Directory_01 variable, 155
 - disabling features
 - cache request log, 146, 155
 - live archiving, 64
 - log file rolling, 144
- E**
- e-mail
 - NT Performance Monitor, 129
 - Enabled
 - in RealSystem Administrator, 86
 - variable, 168
 - in Scalable Multicast list, 167
 - Enabled variable, 168
 - Encoder Authentication Realm, 104
 - in RealSystem Administrator, 58
 - encoder mount point, 164
 - EncoderControlList variable, 178
 - EncoderPassword variable, 178
 - EncoderRealm
 - variable
 - in G2_Encoders list, 164
 - EncoderTimeout variable, 178
 - error log, 144, 153
 - format, 145
 - Error Log Path
 - in RealSystem Administrator, 144
 - variable, 152, 153, 178
 - error messages
 - "File not found", 47
 - "Please stand by...", 60
 - Evaluate Permissions, 106

- variable, 172
 - EvaluatePermissions variable, 172
 - Extensible Markup Language (XML) *See* XML
 - Extension_01 variable, 149, 153, 154
- F**
- farm mount point, 165
 - FarmSplitSources
 - list, 165
 - FarmSplitSources list, 165
 - features in RealServer, 4
 - "File not found" error message, 47
 - File Size
 - in RealSystem Administrator, 63
 - variable, 63, 156
 - file systems, 160
 - File Time
 - in RealSystem Administrator, 63
 - variable, 63, 156, 157
 - firewalls, 47
 - authentication, 111
 - From
 - in RealSystem Administrator, 93
 - variable, 159, 160
 - FSMount list, 160, 161, 167, 169, 170
- G**
- G2 Java Monitor, 152
 - applet mode, 128
 - application mode, 128
 - described, 123
 - G2_Encoders list, 164
 - G2SLTA, 57, 64, 65, 66
 - GET, appearance in access log, 133
 - graph of RealServer activity
 - G2 Java Monitor, 123
 - Windows NT Performance Monitor, 129
 - Group
 - in RealSystem Administrator, 100
 - variable, 171, 178
 - in AuthenticationRealms list, 170
 - GUID Registration Prefix
 - variable, 173
- H**
- Host Name
 - in RealSystem Administrator, 71
 - HostAddress variable, 167
 - HTTP Deliverable
 - list, 158
 - HTTP Port, 39, 47
 - and firewalls, 49
 - and Web servers, 15
 - in Access Control list, 94
 - in AccessControl list, 160
 - in live broadcasting, 59
 - in on-demand streaming, 54
 - in RealSystem Administrator, 40
 - in URL, 21, 22, 87
 - variable, 87, 152, 160, 178
 - Web server and RealServer on same system, 15, 47
- I**
- InputFile variable, 178
 - Internet Multicast Backbone, 78
 - IOBufferSize variable, 178
 - IP Address
 - in RealSystem Administrator, 83
 - IP Binding
 - list, 47, 156
 - IPBindingList variable, 178
- J**
- Java class files, 127, 128
 - Java Monitor, 123
- L**
- License Directory
 - in RealSystem Administrator, 32
 - variable, 32, 152, 153
 - license information, 4
 - LicenseKey variable, 178
 - limiting access by bandwidth, 90
 - limiting access by player version, 92
 - limiting access to HTML pages, 90
 - limiting connections, 90
 - links *See* URL
 - list tag, 148
 - Live Archive
 - list, 61, 156
 - live mount point, 164
 - LiveFileBandwidthNegotiation variable, 178
 - LiveFilePassword variable, 178
-

- LiveFileSize variable, 178
 - LiveFileTarget variable, 178
 - LiveFileTime variable, 178
 - local file system, 161
 - localadminfs mount point, 163
 - LocalHost variable, 178
 - location of files, 54
 - Log Path, 143
 - variable, 51, 145, 149, 152, 153, 178
 - Log Roll Frequency
 - variable, 175
 - Log Roll Size
 - variable, 175
 - Log Rolling, 144
 - Log Rolling Frequency, 143
 - Log Rolling Interval, 143
 - Log Rolling Time Period, 144
 - Logging Style
 - default value, 131
 - format, 136
 - options, 141
 - variable, 175, 178
 - logs
 - access log, 131
 - customizing, 140
 - format, 132
 - rolling, 143
 - accesslog.txt, 117
 - and with NT Performance Monitor, 129
 - cached requests log, 145
 - directory, 116
 - error log, 144
 - media cache requests log, 145
 - multicast, 81
 - reglog.txt, 116
- M**
- Max Bandwidth
 - in RealSystem Administrator, 91
 - variable, 157
 - MaxBandwidth
 - variable, 179
 - MaxBandwidth variable, 157
 - Maximum Bandwidth
 - in RealSystem Administrator, 91
 - MaxThreads variable, 179
 - media cache
 - authentication, 111
 - MIME Types
 - configuring on Web server, 30
 - in RealSystem Administrator, 30, 31
 - list, 148, 149, 153
 - Min Player Version
 - variable, 158
 - Minimum RealPlayer Version, 158
 - MinimumPlayerProtocol, 92
 - MinPlayerProtocol
 - variable, 158, 179
 - MinPlayerVersion variable, 158
 - Monitor Password
 - in RealSystem Administrator, 153, 179
 - variable, 128, 153, 179
 - Monitor Port
 - variable, 128, 152, 179
 - MonitorConnections variable, 179
 - mount points
 - multiple, 16
 - mSQL, 122
 - Multicast Control
 - list, 92
 - MulticastAddressRange variable, 179
 - MulticastControlList variable, 179
 - MulticastDeliveryOnly variable, 179
 - multicasting
 - authentication, 111
 - back-channel, 78
 - in URL, 84, 87
 - log files, 81
 - PNA, 79
 - requiring use of, 92
 - RTSP, 78
 - scalable, 80
 - MulticastPort variable, 179
 - MulticastTTL variable, 179
- N**
- NoArchive variable, 157
 - No-Cache Directories/Files, 45
 - NoSplit variable, 165
 - NT *See* Windows NT

- O**
 - ODBC compliance, 121
 - OutputFile variable, 179
- P**
 - Password
 - in RealSystem Administrator, 103, 104
 - variable, 164
 - password
 - for content users, 96
 - for encoder users, 24, 96, 104
 - for RealSystem Administrator users, 36, 96, 104
 - mkpnpass, 98
 - Path
 - variable, 174
 - Path To DB Plugin
 - variable, 174
 - Path variable, 174
 - Path_01 variable, 159
 - PathToDBPlugin
 - variable, 174
 - PathToDBPlugin variable, 174
 - PAUSE, in access log, 139
 - permissions table, 119
 - Pid Path
 - variable, 30, 51, 152, 153, 179
 - placing files, 53, 54
 - Plugin Directory
 - variable, 152, 153, 160
 - Plugin ID
 - authentication protocol options, 98
 - in AuthenticationRealms list, 170, 171
 - options, 171
 - in Databases list, 174
 - options, 174
 - in RealSystem Administrator, 102
 - Plus Only
 - in RealSystem Administrator, 158
 - variable, 158
 - PNA multicast, 79
 - PNA Port
 - in live broadcasting, 59
 - in on-demand streaming, 54
 - in push splitting link, 76
 - in RealSystem Administrator, 39, 83
 - variable, 23, 152, 169, 179
 - PNA protocol, 14, 15
 - pn-admin, 161, 162
 - pn-encoder, 161, 164
 - pn-farmsplit, 161, 165
 - pn-live3, 161, 164
 - pn-local, 161, 163
 - pn-ramgen, 161, 163
 - pn-splitter, 161, 167
 - Port
 - described, 15
 - in RealSystem Administrator
 - G2 Encoders, 58
 - Pre-G2 Encoders, 59
 - pull splitting, 75
 - push splitting, 72, 74
 - variable
 - in G2_Encoders list, 163, 164
 - in Pre_G2_Encoders list, 164
 - in pull splitting list, 76, 166, 167
 - in push splitting list, 165, 166
 - in Splitter_DoubleURL list, 167
 - in Splitter_Farm list, 166
 - Port Range
 - variable, 167, 168
 - Port_01 variable, 160
 - Ports
 - list, 160
 - Ports variable, 159
 - ppvbasic.txt
 - defined, 114
 - warning, 115
 - Pre_G2_Encoders list, 164
 - Process ID, 51
 - ProtectedVirtualPath variable, 171, 172
 - Provider
 - in RealSystem Administrator, 100, 171
 - variable, 170, 171
 - pull splitting, 69, 70
 - push splitting, 69, 70
- R**
 - Ramgen
 - and content creators, 24
 - described, 19

- in configuration file, 163
 - in HTTP Deliverable list, 90, 158
 - in URL, 21
 - mount point, 16, 18, 163
 - plug-in name, 161
 - Real Time Streaming Protocol *See* RTSP
 - RealAdministrator list, 163
 - RealAdministrator_Files list, 162
 - RealContent list, 161
 - Realm
 - defined, 97
 - in RealSystem Administrator, 100
 - variable, 179
 - in AuthenticationRealms list, 170, 171
 - in CommerceRules list, 172
 - in Pre_G2_Encoders list, 164
 - in RealAdministrator_Files list, 162, 163
 - RealPix, 15
 - RealPlayer Plus, 92
 - RealPlayer Plus Only, 92
 - RealPlayer version, 92
 - RealSystem Administrator, 35, 162
 - starting, 35
 - RECENT, appearance in access log, 139
 - recording live broadcasts, 61
 - RECSTART, in access log, 139
 - redirect directory, 117
 - register_log table, 120
 - Registration Prefix
 - in RealSystem Administrator, 109
 - reglog.txt, 114, 116
 - reports
 - access log, 131
 - customizing, 140
 - rolling, 143
 - accesslog.txt, 117
 - error log, 144
 - media cache requests log, 145
 - reglog.txt, 116
 - Windows NT Performance Monitor, 129
 - Resend
 - in RealSystem Administrator, 84
 - variable, 169
 - ResolverPort variable, 179
 - RestoreOriginalPrivilegeOnReload variable, 179
 - Restricted Ports, 94
 - restricting access, 90
 - RESUME, in access log, 139
 - rm files
 - bandwidth negotiation, 55
 - rmserver.pid, 51
 - RTSP multicasting, 78
 - RTSP Port
 - in Access Control list, 94
 - in live broadcasting, 59
 - in on-demand streaming, 54
 - in push splitting URL, 76
 - in RealSystem Administrator, 23, 40, 83
 - in URL, 74
 - variable, 22, 23, 152, 160, 169
 - in Multicast list, 169
 - use in on-demand streams, 54, 59
 - RTSP protocol, 15
 - Rule Number
 - in RealSystem Administrator, 83
- S**
- saving live broadcasts, 61
 - scalable mount point, 85
 - scalable multicasting, 80
 - SEEKSTART, in access log, 139
 - ServerHost variable, 179
 - ServerPassword variable, 179
 - ServerPort variable, 179
 - ShortName variable
 - described, 161
 - in G2_Encoders list, 163, 164
 - in Pre_G2_Encoders list, 164
 - in Ramgen list, 163
 - in RealAdministrator list, 163
 - in RealAdministrator_Files list, 162
 - in RealContent list, 161
 - in Scalable Multicast list, 167
 - in Splitter_DoubleURL list, 166, 167
 - in Splitter_Farm list, 165
 - SIGHUP command, 51
 - simultaneous content creation *See* live ar-

- chiving
- SMIL file
 - and content creators, 24
 - defined, 13
 - in access log, 136
 - in URL, 21, 55, 60
 - multicasting and, 83, 169
- SMIL files
 - authentication, 110
- split mount point, 167
- Splitter Buffer Delay
 - in RealSystem Administrator, 73
 - variable, 165, 166, 179
- Splitter Host Name
 - in RealSystem Administrator, 71, 73
 - variable, 165
- Splitter Resend Buffer
 - in RealSystem Administrator, 72
 - variable, 165, 179
- Splitter Source List
 - list, 165, 166, 179
 - variable, 179
- Splitter Source Probe Interval
 - in RealSystem Administrator, 73
 - variable, 165, 166, 180
- Splitter Source Timeout
 - in RealSystem Administrator, 72
 - variable, 165, 166, 180
- Splitter Timeout
 - in RealSystem Administrator, 73
 - variable, 165, 166, 180
- Splitter_DoubleURL list, 167
- Splitter_Farm list, 165
- SplitterAnnouncePort variable, 179
- SplitterControlList variable, 179
- SplitterMaxResendPPS variable, 179
- splitting, 67
 - authentication, 111
 - limiting splitter access, 70
 - pull splitting, 69
 - in URL, 76
 - push splitting, 69
 - in URL, 74
- starting RealServer, 25
- Stat1
 - location in access log, 132
 - syntax, 137
- Stat2
 - location in access log, 132
 - syntax, 137
- Stat3
 - location in access log, 132
 - syntax, 138
- statistics
 - displaying in G2 Java Monitor, 123
 - See also* reports
- statistics type 1
 - gathering with StatsMask, 142
 - syntax, 137
- statistics type 2
 - gathering with StatsMask, 142
 - syntax, 137
- statistics type 3
 - gathering with StatsMask, 142
 - syntax, 138
- statistics, collecting in access log, 131
- Stats Mask
 - and RealPlayer, 142
 - default value, 131
 - options, 137, 142
 - scalable multicasting, 82
 - variable, 4, 132, 175, 180
- STOP, in access log, 28, 139
- stopping RealServer
 - UNIX, 30
 - Windows NT, 28
- streaming, 11, 53
- SupportPathDirectory
 - in RealSystem Administrator, 73
 - variable, 165
- SureStream
 - defined, 55
 - multicasting, 79, 81
 - RTSP, 15
 - splitting, 69
- Synchronized Multimedia Integration Language, *See* SMIL file

- T**
- tables
 - access_log, 118, 121
 - permissions, 118, 119
 - redirect, 118, 120
 - register_log, 118, 120
 - users, 118
 - Target Directory, 61
 - in RealSystem Administrator, 63
 - variable, 156, 157
 - TCP control channel, 78
 - text files
 - authentication data storage structure, 114
 - Timeout variable, 180
 - To
 - in RealSystem Administrator, 93
 - variable, 75, 159
 - Transport
 - in RealSystem Administrator, 159
 - variable, 159
 - ts.log file, 43, 145
 - TSEnable variable, 154, 155
 - TSLog variable, 155
 - TSLogPath variable, 155
 - TSPort variable, 155
 - TTL variable, 84, 87, 167, 168, 169
- U**
- unicasting, 77
 - UNIX
 - PID, 51
 - SIGHUP, 51
 - special features, 50
 - starting RealServer, 29
 - stopping RealServer, 30
 - URL
 - authentication, 110
 - for on-demand streams, 55
 - for pull splitting, 76
 - for push splitting, 74
 - for SMIL files, 55
 - from a Ram file to a clip, 22
 - from a SMIL file to a clip, 22
 - from a Web page to a clip, 21
 - from a Web page to a Ram file, 22
 - from a Web page to a SMIL file, 21
 - from a Web page to scalable multicast, 87
 - in RealPlayer, 23
 - parts of, 13
 - port values, 15
 - protocols, 20
 - SMIL files, 13
 - telling content creators, 24
 - URL variable, 180
 - UseGUIDValidation
 - variable, 172
 - user authentication vs. player validation, 105
 - User variable, 180
 - UserDir variable, 180
 - UserList variable, 180
- V**
- Valid Player Only, 92
 - variable, 158
 - ValidPlayerOnly variable, 158
 - variable tag, 148
 - Virtual Path
 - in RealSystem Administrator, 86
 - variable, 167, 168
- W**
- Web server
 - and firewalls, 49
 - and RealServer, 15, 39, 47
 - location of files, 54
 - log format, 133
 - MIME types on, 30
 - Windows NT
 - Performance Monitor, 50, 128
 - running multiple RealServers, 28
 - special features, 50
 - stopping RealServer, 28
- X**
- XML
 - configuration file, 8, 37, 147
 - license files, 32
 - XML declaration tag, 147