RealServer™ Administration Guide

Version 5.0

RealNetworks, Inc.

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

Printed in the United States of America. Part #: 307423

Contents

Chapter 1 Overview

The RealSystem™ 5.0 family of products provides a complete solution for streaming media and allows your Internet or intranet users to easily access multimedia content in real time.

Superb audio and video quality, high transmission reliability, great broadcast scaleability, and flexible modes of delivery have made RealNetworks software the market leader for professional live and on-demand streaming media on the Internet or your intranet. RealSystem provides a proven, reliable platform for multimedia streaming and allows you to grow as your needs and use expand.

An updated version of this document is available online at the RealNetworks Web site at:

www.real.com

Using This Guide

This *RealServer*™ *Administration Guide* explains how to install, configure, and operate the RealServer, part of every RealSystem 5.0 product. Step-by-step installation, configuration, and customization procedures for the server and each feature allow you to tailor your system and integrate it seamlessly with your network. Content and template samples are included when you install RealServer.

This guide is intended for the technical system administrator who is familiar with the Internet or intranet and who understands how to work with .html and .rm files. IS managers, server administrators, web masters and others providing Web pages for the Internet and intranet may also find this document useful.

Conventions

This manual uses the following conventions:

`Command` Monospace font represents commands to be typed or information displayed on the screen.

`<file name>` Angle brackets show where to insert information, such as the name and location of a file. **Note**: the brackets are not required when typing the command.

`[ ]` Square brackets show optional command arguments. **Note**: the brackets are not required when typing the command.

Bold Bold font is used for names of files, directories, commands, and options.

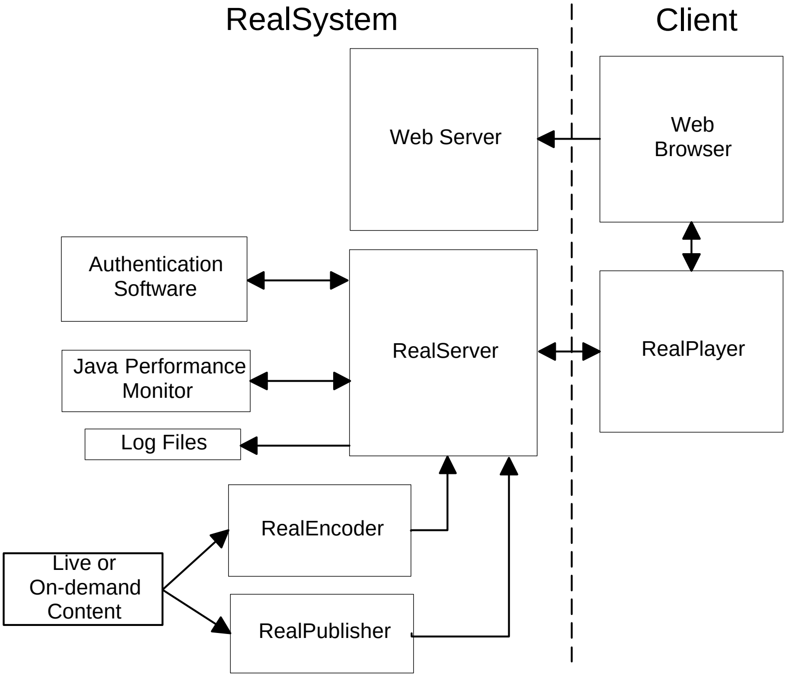
RealServer 5.0 New Features

RealServer 5.0 continues to provide award-winning RealAudio® and RealVideo® technology and offers many exciting new features:

- **New media types:** RealServer now streams RealFlash animation in addition to RealAudio and RealVideo.
- **Ad Insertion:** RealServer now allows you to insert advertisements anywhere in a content clip with no buffering delay.
- **Authentication:** RealServer now offers the capability to make some content available to only select Internet and intranet users.
- **Web-based Administration Page:** Administer the RealServer remotely over the Internet or your intranet using advanced server management features. The Java Performance Monitor enables you to configure any server via a Web browser.
- **Fault Tolerance System:** RealServer keeps going in the event of problems, and logs problems such as faulty streams, memory corruption, and product incompatibilities without shutting down the server.

Conceptual Overview

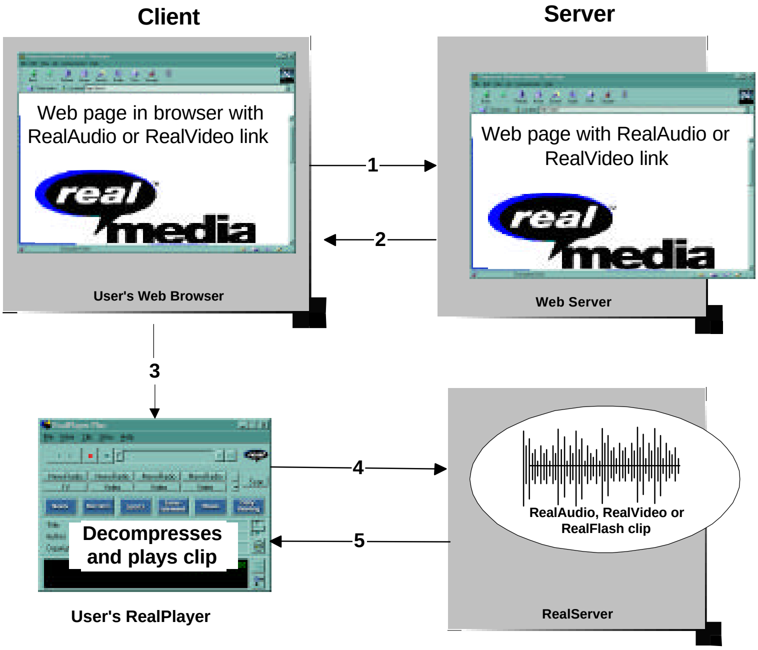
The following diagram shows how RealServer components work together in relation to the RealSystem family of products.



RealServer employs a client/server architecture to provide features such as live broadcasting, interactive stream control (fast-forward and seek options), efficient use of network bandwidth, and scaleability that can support large numbers of concurrent users.

Sophisticated compression, buffering, and transmission techniques allow RealServers to stream audio and video animation to players, which continuously decompress and play back the stream in real time. Users can watch or listen to entire programs or navigate on-demand clips to experience what they want when they want it.

The following figure shows how RealServer components work together to deliver a typical RealAudio or RealVideo clip to a user (client).


Client
Server
Web page in browser with RealAudio or RealVideo link
User's Web Browser
Web page with RealAudio or RealVideo link
Web Server
1
2
3
4
5
Decompresses and plays clip
User's RealPlayer
RealAudio, RealVideo or RealFlash clip
RealServer

1. When a user visits your Web page and clicks a link to a RealAudio or RealVideo file, the Web browser requests the metafile from your Web server.
2. Your Web server sends the metafile to the user’s Web browser and based on the metafile extension, sets the MIME type.
3. Based on the MIME type, the Web browser starts the user’s RealPlayer as a helper application and passes it the metafile.
4. The RealPlayer reads the first URL from the metafile and requests that clip from the RealServer at your Web site.
5. Your RealServer streams the clip to the user’s RealPlayer, which plays the clip in real-time.

No messages pass between your RealServer and your Web server.

Components of RealSystem 5.0

A RealSystem includes a RealServer and one or more tools for publishing, editing, and serving media files, for monitoring and managing your secured Web site.

The following table lists the components of a RealSystem:

Component	Description
Tools	The tools for modifying RealAudio and RealVideo clips: RMTools, RMMerge, RMEdit, RMPaste, RMCut, and RMDump.
RealPublisher™	The program that creates RealAudio and RealVideo from a digitized audio clip, live audio input, digitized video clip, or live video input. RealPublisher allows the creation of synchronized multimedia presentations. RealPublisher is included with all RealSystem products. RealPublisher creates RealAudio and RealVideo clips, creates HTML, and can publish your Web page to a remote server.
RealEncoder™	The program that creates RealAudio and RealVideo from a digitized audio clip, live audio input, digitized video clip, or live video input. RealEncoder allows the creation of synchronized multimedia presentations. RealEncoder is available for free download from the RealNetworks Web site. It can create clips but it cannot automatically publish them to your Web site.
RealServer™	The server program that delivers live and on-demand RealAudio, RealVideo, and RealFlash clips over a network. One RealServer can simultaneously deliver different clips to many RealPlayers.
Java Performance Monitor	The program that allows you to track current usage of the server, configure the server, determine the most popular clips, and view the server loads in real time, all via a Web browser.

Component	Description
System Manager	The program that allows you to track current usage of the server, configure the server, determine the most popular clips, and view the server loads in real time.
Ad Insertion and Content Rotation	The feature that seamlessly inserts a rotated list of ad clips into a content clip.
Authentication	The files necessary for the Authentication front-end and back-end, as well as one database application and templates for other databases.
Log Files	All RealServer activity is recorded in log files. Log file data can be analyzed for reporting purposes.

Other Software

RealSystem can be used as a standalone Internet or intranet system, but works with other software to provide a complete Web-based solution.

Web server The server program that delivers Web pages to clients. Typically, RealAudio and RealVideo clips are accessed from a Web browser when a user clicks a link on a Web page. The Web server also delivers the visual part of Synchronized Multimedia presentations.

RealSystem 5.0 works with all popular Web servers.

Programming software A programming tool for creating, modifying, and compiling CGI files used with Authentication. C and Perl are examples.

ODBC-compliant database Used by Authentication. An mSQL database is provided with this release of RealSystem 5.0. This package also has templates for other common databases.

Client Software

RealSystem interfaces with the following software on your clients' machines to deliver media.

RealPlayer The client program that enables users to view and listen to RealAudio, RealVideo, and RealFlash clips. The RealPlayer can be used as a standalone application or as an ActiveX or Netscape Plug-in.

Web browser The client program that enables users to view Web pages. The Web browser is also used to display the visual part of Synchronized Multimedia presentations and can contain integrated the RealPlayer Plug-in.

RealSystem works with all popular Web browsers.

RealSystem Media Files

RealSystem uses several file types, each identified by a specific file extension. A brief overview of the files and their file extensions is listed below.

RealAudio clip (.ra) Audio encoded to RealAudio format. This type of file is delivered by RealServer and is played on a RealPlayer.

RealVideo clip (.rm) Audio and video encoded to RealVideo format. This file can contain multiple streams, including audio, video, image maps, and events. This type of file is delivered by RealServer and is played on a RealPlayer.

RealAudio or RealVideo metafile (.ram) Connects a Web page to one or more RealAudio or RealVideo clips. This metafile is located on your Web server and is linked to from your Web page. A metafile contains the URL(s) for one or more clips stored on your RealServer.

RealPlayer Plug-in metafile (.rpm) Like a RealAudio or RealVideo metafile (above), but used with RealPlayer Plug-in for Netscape Navigator and Internet Explorer 3.0 or later.

RealFlash clip (.swf) Animation in RealFlash format.

Each of these file types is described in greater detail in the following sections.

RealVideo and RealAudio Clips

RealAudio and RealVideo are highly compressed files that are formatted to deliver the best possible sound and video over a limited-bandwidth connection.

- A RealAudio clip is a file (.ra) or live broadcast containing sound encoded in one of the RealAudio formats.
- A RealVideo clip is a file (.rm) or live broadcast containing sound and/or video encoded in one of the RealVideo formats.

Because there is no single best format for delivering audio and video, several formats are available. You can use RealPublisher and RealEncoder to provide formats that are optimized for different types of content and connection. You can choose to provide a clip in one or more formats based on the type of content and the available bandwidth. For example, you would use a different format to deliver speech over a 14.4 Kbps modem than you would to deliver a music video over an ISDN connection.

RealFlash Animation Clips

RealFlash is a highly compressed file that is formatted to deliver the best possible animation over a limited bandwidth connection. RealServer provides the only streaming solution available for RealFlash synchronized with audio. Combining RealFlash with RealAudio for fully synchronized low-bandwidth animation-and-audio allows content providers to provide long-form animated presentations over their Internet or corporate intranet.

Customizable Features of RealAudio and RealVideo Files

Image Maps

RealVideo allows users to interact with video content using clickable image maps.

An image map is an active area that overlays the RealVideo display. When clicked by the user, this “hot spot” sends a URL to the user’s Web browser. The Web browser goes to the new Web page without interrupting the RealPlayer.

Image maps are fully customizable; actions can be connected to rectangular, circular or polygonal content regions and can be varied over time intervals that you define when you encode the clip.

Encoding RealAudio and RealVideo Clips

Media files need to be encoded into RealPlayer format using one or more of the many codecs available with any installation of RealPublisher or RealEncoder.

Encoding a RealAudio or RealVideo clip is a one-way process; you cannot convert a RealAudio or RealVideo file back into the original source format. When you encode a RealAudio or RealVideo clip, the original file is not modified. If you want to be able to encode media in more than one format, it is important that you archive (save) the original source.

It is possible to modify various attributes of RealAudio and RealVideo clips without reencoding them. For example, you can override the title, author, and copyright of an encoded clip. For more information, see the customization sections in Chapter 8, “Configuring Your Web Site” or review the *RealAudio and RealVideo Content Creation Guide*.

Title, Author and Copyright

RealAudio and RealVideo clips include text strings for the title, author, and copyright. This text is displayed by RealPlayer when a clip is played. Although the player usually labels the text as title, author, and copyright, the text you supply when you encode the clip is displayed. After encoding, you can override encoded values using the metafile or RMEdit and RMTools.

Live Encoding and Live Broadcasting

The source for a RealAudio or RealVideo clip can be prerecorded file or live input. The encoded RealAudio or RealVideo clip can be stored as a file for later use, broadcast live over a network, or simultaneously stored as a file and broadcast live. If simultaneously encoding and broadcasting, you can save (or “archive”) the live broadcast in the encoded format.

All versions of RealEncoder and RealPublisher can encode a live input source. For details about encoding, refer to the *RealAudio and RealVideo Content Creation Guide*.

System Requirements

This section describes the hardware and software required to run RealServer.

Hardware Requirements

Ensure that the machine on which you install RealServer is secure and is not accessible to unauthorized users.

Memory Requirements

RealServers require approximately 6 MB of available RAM plus 40 KB RAM for each simultaneous stream. For example, to support 100 simultaneous connections—without Authentication or Ad Insertion—requires approximately 10 MB of available memory.

CPU Requirements

RealServer has a modest CPU impact. A 120 MHz or faster processor is required. For example, a 100-stream RealServer operating on a 120 MHz Pentium computer consumes less than 30% of the CPU cycles. With enough network bandwidth, the same computer can deliver at least 400 28.8 Kbps streams simultaneously.

Disk Space Requirements

The RealServer program files require about 3 MB of disk space. You also need disk space for the content files you are serving. See also the *RealAudio and RealVideo Content Creation Guide*.

Bandwidth Requirements

Bandwidth requirements for audio and video depend on the encoding/decoding formats used. A T1 line is recommended for all installations.

Encoding Requirements

See the *RealAudio and RealVideo Content Creation Guide* for a list of hardware requirements for encoding media clips.

Other Requirements

You’ll need a Web server for presenting your content. If your RealSystem product includes Authentication, you can also use your own database for tracking user information; if your RealSystem includes Ad Insertion, you can use a third-party ad server for scheduling ad placement and rotation.

Compatible Web Servers

Although a Web server is required to make the best use of RealAudio and RealVideo, it need not be installed on the same machine as RealServer.

To make the best use of RealAudio and RealVideo, you must have a Web site and a registered domain.

To ensure that the media files are accessible from links embedded in your Web pages, your Web server must be configured to recognize the various MIME types of your anticipated audience. RealServer can be configured to work with any Web server that supports configurable MIME types.

RealServer is compatible with the following Web servers

- Apache 1.1.1 and later versions
- CERN HTTPD version 3.0
- EMWC HTTPS version 0.96
- HTTPD4 Mac
- Mac HTTP
- Microsoft Internet Information Server
- NCSA HTTPD versions 1.3 or 1.4
- Netscape Netsite and Netscape Enterprise Server
- O'Reilly Website NT
- Spinner versions 1.0b12 through 1.0b15
- Webstar and Webstar PS

To view a current list of Web servers that are compatible with RealServer 5.0, visit the RealNetworks Web site at:

www.real.com

Compatible Databases for Authentication

RealServer comes with a database (mSQL), but can also be used with any ODBC-compliant database. Templates for common databases are included.

Compatible Advertisement Servers for Ad Insertion

RealServer integrates with several third-party advertisement servers. Please check the RealNetworks site at **www.real.com** for the latest additions.

Chapter 2 Installing RealServer

This chapter explains how to install your RealServer, how to start, stop and test your server, and how to get help if you have installation problems.

There are three options for installing RealServer:

- Installing on Windows platforms
- Installing on UNIX platforms
- Upgrading from a previous version

Procedures for each option can be found in subsequent sections of this chapter, but before turning to the appropriate procedure, please review the installation notes.

Installation Notes

The RealServer setup program creates directories and copies the program, configuration, and sample content files into them. The Windows and UNIX installation sections list the default file and directory structures created during setup.

In general, you should not store media files in directories on your Web server. The RealServer is independent of and does not communicate with the Web server.

Smart Networking

RealServer 5.0 can broadcast to players behind firewalls by transmitting all data in HTTP over port 80 after more efficient transports have failed. Setup asks if you would like to enable Smart Networking on port 80. The default answer is “yes.” It is strongly recommended that you take full advantage of this feature to maximize the viewership of your content.

Normally you cannot use Smart Networking if your RealServer is installed on the same machine as your Web server. The second bullet item below describes a solution.

If you enable this feature, the setup program checks port 80 for the presence of a Web server. If a conflict is detected, you can enable Smart Networking in one of two ways:

- You can move the RealServer to another machine and activate Smart Networking by setting the **HTTPPort** setting in the server configuration file (**server.cfg**).
- You can configure the computer to have two IP addresses (consult your system documentation for details), then configure RealServer to use the alternate IP address by setting the **IPBindingList** and **HTTPPort** settings in the **server.cfg** file. Note that Microsoft Internet Information Server (IIS) version 3.0 will bind to all IP addresses, regardless of this setting. IIS version 4.0 does allow binding to multiple IP addresses.

Testing RealServer with a Sample Media Clip

After installing RealServer, it is recommended that you test your installation by playing a sample media clip. You can do this easily from the Administration page, or you can use a Web browser to manually test the clip:

To play a test clip manually:

1. In a Web browser, type the following URL:

```
http://<servername>:<port>/ramgen/<filename>.rm
```

where **`<servername>`** is the name of your Web server, **`<port>`** is the port number it’s using, and **`<filename>`** is the name of the file to be served. The key word “**ramgen**” tells RealServer to simulate .ram file behavior (.ram files are described in Chapter 8, “Configuring Your Web Site”). Be sure to type “**ramgen**” in lowercase. You can also include optional arguments, as described in “Customizing Calls to Video and Audio Content” on page 179. For example, if the URL you would reference in the .ram file or the RealPlayer is:

```
pnm://www.server1.com/hello.rm
```

the URL you would type in the browser to test the file is

```
http://server1:7070/ramgen/hello.rm
```

2. If the clip plays correctly, the server is working properly.

If the media file does not play at all, or if the performance or playback quality is poor, check your log files for clues. To learn about log files, see Chapter 9, “RealServer Log Files.”

Installing RealServer on Windows Platforms

When installing RealServer, close all programs and turn off virus protection software to prevent installation conflicts.

To install from the Web

RealServer is available on the RealNetworks Web site at:

www.real.com

Follow the download instructions there.

To install RealServer from a CD-ROM

1. Insert the RealServer CD-ROM in your computer.
2. In Windows Explorer, click the icon for your CD-ROM directory and double-click the Install RealServer icon.
3. Follow the screen prompts on the RealServer Setup screen.

   When prompted for Customer Name and License Key, use cut and paste. Enter them exactly as provided (via email) by RealNetworks or your RealNetworks reseller.

RealServer License Agreement

The setup program displays information about your server license. For example:

```
Your License has the following features:
Valid License
Version:                           2
Platform:                          NT
Start Date:                        5/23/1997
Expiration Date:                   5/23/1999
Licensed Streams:                  10
User Streams:                      0
Multi-Media:                       Enabled
Live:                              Enabled
ISP Hosting:                       Disabled
ISP Only:                          Disabled
User Max Limit:                    Disabled
Platform Checking:                 Disabled
Intranet Only:                     Disabled
Remote License:                    Disabled
Remote Streams w/ Local Lic:       Disabled
Local Streams:                     Enabled
Remote Streams w/ Remote Lic:      Disabled
Dynamic ISP licensing:             Disabled
Restricted IP Access:              Disabled
Ad Insertion:                      Disabled
Client Authentication:             Disabled
RealFlash:                         Disabled
```

RealServer Windows Directories and Files

The RealServer setup program creates the following directories and copies files into them:

`Real\Server` (default server directory)

```
\bin
\commerce
\content
\docs
\logs
\plugins
```

Refer to the following sections for descriptions of the files that are copied to each of these directories during installation.

Real\Server Directory

The **Real\Server** is the default root directory for RealServer. This directory contains the following ASCII text files:

File	Description
server.cfg	The configuration settings for the server.

Bin Directory

The **bin** directory contains the following executables and libraries:

File	Description
adli3250.dll	Ad Insertion plug-in library.
crtsvc.exe	The service utility for installing RealServer (for Windows NT only).
delsvc.exe	The service utility for uninstalling RealServer (for Windows NT only).
pnserver.exe	The RealServer program.
raconv.exe	The bandwidth negotiation file converter, which converts RealAudio and RealVideo files to a bandwidth

File	Description
	negotiation naming scheme.
rmfile.exe	The live file creation utility for creating (archiving) files from live broadcasts.
srvrctrl.exe	Server Control Application for starting and stopping RealServer.
srvrctrl.ico	Server Control Application icon.
swftune.exe	A command-line utility for altering the bit rate of a RealFlash file.

Commerce Directory

The **Commerce** directory contains the Authentication files and directories.

Content Directory

The **Content** directory contains sample media files. This directory is also the location in which you should save content files you create.

Note: You should not store media files in directories on your Web server. The RealServer is independent of and does not communicate with the Web server.

The **Content** directory also contains the **Admin** directory, which contains files for administering the RealServer. Also in this directory is the **JavaMonitor** directory, which contains all the Java classes you need to run the Java Performance Monitor.

Docs Directory

The **docs** directory contains the following ASCII files:

File	Description
license.txt	Server license agreement.
problem.txt	A list of known server problems.

File	Description
readme.txt	Information about server installation and operation.

Logs Directory

Until you launch the RealServer, this directory contains only the **log.txt** file. The **logs** directory contains the log files in text format. The .log files are dynamically created when you launch the RealServer.

File	Description
log.txt	This file is a placeholder for the log directory.
pnaccess.log	The access log, which logs information about clients that have connected to the server. If deleted or renamed, this file is dynamically created when you start RealServer.
pnerror.log	The error log, which logs informational and error messages about server operation. If deleted or renamed, this file is dynamically created when you start RealServer.

Plugins Directory

The **plugins** directory stores any plug-in libraries used by RealServer.

File	Description
swf3250.dll	RealFlash plug-in library.
ppvb3250.dll	Authentication data text file
ppvo3250.dll	Authentication ODBC database
jmon3250.dll	Java Performance Monitor page
ppvf3250.dll	Administration page
stat3250.dll	Status page
tstfs3250.dll	Test sample content page

Installing RealServer on UNIX Platforms

Server installation files are available for several UNIX platforms. After selecting the appropriate installation file, you need to launch the setup program. Detailed instructions follow:

Accessing the RealServer Distribution File from the Web

RealServer is available from the RealNetworks Web site at

www.real.com

Follow the instructions on the download page.

Uncompress the file with the following command, then refer to the section titled “Running the Setup Program” on page 21:

```
gunzip -c <filename>.tar.gz | tar xfv -
```

RealNetworks may send your Customer Name and License Key via e-mail, depending on the server package you’ve chosen. If so, cut and paste these values from the email to avoid typographical errors.

Accessing the RealServer Installation Files on CD-ROM

UNIX-based operating systems require you to mount a new file system or device before running the Setup program. The commands needed to mount a CD-ROM differ slightly depending on the system. Follow the appropriate mounting instructions below and then start the setup program.

To mount the CD-ROM on Sun Solaris

1. Insert the CD-ROM and wait for the operating system to mount the CD-ROM.

   If you are running File Manager, a window displays the disk contents.
2. If you are not running File Manager, in a shell type:

   ```
   cd /cdrom/pn_server
   ```

To mount the CD-ROM on all other UNIX-based systems

1. Insert the CD-ROM in the drive.
2. Log in as super-user.
3. From a shell, check for a **/cdrom** directory to mount the CD on. If one does not already exist, type:

   ```
   mkdir /cdrom
   ```

4. Type the appropriate command to mount the CD-ROM:

Operating System	**Command**
Sun SunOS	mount -rt hsfs /dev/sr0 /cdrom
DEC UNIX	mount -t cdfs -o noversion /dev/rz3c /cdrom
SGI IRIX	mount -rt iso9660 /dev/scsi/sc0d7l0 /cdrom
IBM AIX	mount -rv cdrfs /dev/cd0 /cdrom
Hewlett-Packard HP-UX	mount -rF cdfs /dev/dsk/c0t2d0 /cdrom
FreeBSD	mount -rt cd9660 /dev/cd0a /cdrom
BSD/OS	mount -rt cd9660 /dev/sd1 /cdrom
Linux	mount -rt iso9660 /dev/hdc /cdrom

Running the Setup Program

After mounting the CD-ROM and uncompressing the distribution file, you need to launch it as an install program. During installation, the setup program can set some default options, such as passwords and default ports, for you or you can customize these configuration settings.

To start the install program

1. Change directory to the CD-ROM:

   ```
   cd /cdrom
   ```

   Sun Solaris only: Change directory to the **pn_server** directory:

   ```
   cd /cdrom/pn_server
   ```

2. Change directory to the server directory:

```
cd server
```

HP-UX only: HP-UX computers cannot run **setup** from the CD-ROM. Copy the **server.tgz** file to a directory on the computer and uncompress it, then extract the resulting file using the following commands:

```
gunzip -c <filename>.tar.gz | tar xfv -
```

3. From the directory in which you untarred the distribution file, run the **./setup** program. The following list of navigational controls are displayed:

```
Key Behavior
N   Next
P   Previous
X   Exit
F   Finish (Express Installation)
H   Help
```

4. Type the **Customer Name** as provided (via e-mail or in the application packaging) by RealNetworks or your RealNetworks reseller.
5. Type [**N**]ext to continue.
6. Type the **License Key** exactly as provided (via e-mail or in the application packaging) by RealNetworks or your RealNetworks reseller. Use cut and paste if possible to avoid typographical errors.
7. Type [**N**]ext to continue.
8. Read the Server License agreement. Type [**N**]ext to accept the terms of the license, or any other key to cancel the installation.
9. The setup program displays information about your server license. Type [**N**]ext to continue.
10. All the information necessary for installation has been provided.
11. Press Express [**F**]inish to accept the default installation options or press [**N**]ext to set the options yourself.

If you are setting the options yourself:

1. Type a monitor password, or accept the default (letmein).
2. Type [**N**]ext to continue.
3. Type a Live Encoder password, or accept the default (letmein).
4. Type [**N**]ext to continue.
5. Type a Live File Archive password, or accept the default (letmein).
6. Type [**N**]ext to continue.

7. Type the TCP port on which the RealServer should listen, or accept the default (7070).
8. Type [**N**]ext to continue.
9. Type the e-mail address to which the RealServer should send system error messages.
10. Type [**N**]ext to continue.
11. Type the e-mail server hostname that RealServer should use for e-mail messaging service.
12. Type [**N**]ext to continue.
13. Type [**Y**]es to enable Smart Networking. For more information, see “Smart Networking” on page 13.
14. Type [**N**]ext to continue.
15. Type the user ID for running the server.
16. Type [**N**]ext to continue.
17. Type the group ID for running the server.
18. Type [**N**]ext to continue.

    A screen displays the configuration settings you selected.

    Verify the RealServer configuration settings you chose.
19. Type [**N**]ext to begin copying files to your directory.

    A screen displays information about how to contact RealNetworks for help.
20. Type [**N**]ext to continue.
21. Type E [**X**]it to complete the installation.

After installing RealServer, it is recommended that you test your installation by playing a sample media clip. For instructions, see “Testing RealServer with a Sample Media Clip” on page 14 of this chapter.

Note: Setup asks you to register online. RealServer will not start unless you register.

RealServer License Agreement

The setup program displays information about your server license. For example:

```
Your License has the following features:
Valid License
Version:                       2
Platform:                      UNIX
Start Date:                    5/23/1997
Expiration Date:               5/23/1999
Licensed Streams:              10
User Streams:                  0
Multi-Media:                   Enabled
Live:                          Enabled
ISP Hosting:                   Disabled
ISP Only:                      Disabled
User Max Limit:                Disabled
Platform Checking:             Disabled
Intranet Only:                 Disabled
Remote License:                Disabled
Remote Streams w/ Local Lic:   Disabled
Local Streams:                 Enabled
Remote Streams w/ Remote Lic:  Disabled
Dynamic ISP licensing:         Disabled
Restricted IP Access:          Disabled
Ad Insertion:                  Disabled
Client Authentication:         Disabled
RealFlash:                     Disabled
```

RealServer Directories and Files on UNIX Platforms

The RealServer setup program creates the following directories and copies files into them.

```
pnserver (default root directory)
        /bin
        /commerce
        /content
        /docs
        /logs
        /plugins
```

Refer to the following sections for descriptions of the files that are copied to each of these directories during installation.

The Root Installation Directory

The default root installation directory contains the following ASCII text files:

File	Description
server.cfg	The configuration settings for the server.

The bin Directory

The **bin** directory contains the following executable files:

File	Description
adlist.so.5.0	The Ad Insertion executable.
pnserver	The RealServer executable.
raconv	The bandwidth negotiation file converter, which converts RealAudio files to a bandwidth negotiation naming scheme.
rainfo	Utility that displays header information (title, author, copyright) about a clip.
rmcut	Cuts a specific portion of a RealVideo file without changing the original source.
rmedit	Modifies the title, author, copyright, comment, MIME type, or stream name. It can also be used to print the current values for the file or stream.
rmfile	The live RealMedia file creation utility, which creates (archives) files from live broadcasts.
rmpaste	Combines two or more RealVideo files.
rssm	The System Manager executable.
swftune	A command-line utility for altering the bit rate of a RealFlash file.

The commerce Directory

The **commerce** directory contains the Authentication files and directories.

The content Directory

The **content** directory contains sample media files. This directory is also the location in which you should save content files you create.

Note: You should not store media files in directories on your Web server. The RealServer is independent of and does not communicate with the Web server.

The **content** directory also contains the **Admin** directory, which contains files for administering the RealServer. Also in this directory is the **JavaMonitor** directory, which contains all the Java classes you need to run the Java Performance Monitor.

The docs Directory

The **docs** directory contains the following ASCII files:

File	Description
PROBLEM	A list of known server problems.
README	Information about server installation and operation.
License.txt	Server license agreement.

The logs Directory

Until you launch the RealServer, this directory is empty. The **logs** directory contains the log files in ASCII text format. These files are dynamically created when you launch the RealServer.

File	Description
pnaccess	The access log contains information about clients that have connected to the server.
pnerror	The error log lists information and error messages about server operation.
pnserver.pid	When the server is running, this file contains the server's current process ID (PID) When the server is not running, this file contains the server's last-used PID.

The plugins Directory

The **plugins** directory stores any plug-in libraries used by RealServer.

File	Description
swf3250.dll	RealFlash plug-in library.
ppvbasic.so.5.0	Authentication data text file
ppvmsql.so.5.0	Authentication mSQL database
jmonfs.so.5.0	Java Performance Monitor page
ppvfs.so.5.0	Administration page
statfs.so.5.0	Status page
tstfs.so.5.0	Test sample content page

Upgrading RealServer From a Previous Version

If you are upgrading from a previous version of RealServer, you should install the new version in addition to, and not instead of, the original version. Once you have installed, configured, and tested the new RealServer, you can then replace your old version with the new installation.

Installing with a Previous Version

To install without replacing an existing RealServer

1. Install the new version in a new directory.
2. In your original RealServer configuration file (**server.cfg**) note the **PnaPort** entry. (For information on editing the settings in this file, see Chapter 4, “Configuring and Maintaining RealServer.”) If the setting is not present, RealServer is using the default port 7070. You need to use a different port number for testing the new RealServer installation. To use a different port number, add or edit the following line in the **server.cfg** file for the new installation:

   ```
   PnaPort   7072
   ```

   Be sure to specify a port that is not used by another configuration parameter such as **ResolverPort**.
3. Send a test URL to the new RealServer. Add :**7072** (as in step 2, above) to the pnm URL.

Note: You cannot run two versions of RealServer simultaneously as a service on Windows NT. Run the new version of RealServer from a command window until you are ready to replace the original version with the new version.

```
pnm://<my.server>:7072/144.rm
```

Configuring the New Version

After you have tested your new version of RealServer, configure it to match the original version.

To configure the new RealServer

1. Copy the **BasePath** setting from the **server.cfg** file of your original RealServer to the **server.cfg** file for the new version. It is best to use an absolute base path for the **BasePath** entry.
2. Send a test URL to the new RealServer. Remember to add :7072 to the pnm URL:

   ```
   pnm://<my.server>:7072/welcome.rm
   ```

Replacing the Old Version with the New Version

When you are satisfied that the new RealServer installation supplies existing content correctly, stop the original server and move the new server into production.

To replace the old version with the new version

1. Stop both instances of RealServer. For instructions on how to do this, see Chapter 3, "Using RealServer."
2. Rename the directory of original installation to something like **/old**.
3. Rename the directory for the new installation to the name of the original installation.
4. Copy all appropriate settings from the original configuration file into the new configuration file.
5. Set the **PnaPort** entry to its original value (default 7070).
6. Change the Web page links back to the original port number.

Troubleshooting RealServer Installation

This section describes the most common installation problems.

“Incorrectly Entered Customer Name or License Key”

If you enter the wrong customer name or license key, the setup program quits.

If you typed in the customer name or license key, ensure you did not make any typing errors. If you cut and pasted in the customer name and or license key from the e-mail message you received from RealNetworks, make sure that you pasted the entire name and license key and that you did not introduce any extra spaces.

If you have verified that the customer name and license key are correct, and you still receive this error message, contact Customer Service at RealNetworks.

“The Assigned User Name or Group is Invalid”

If you assign a user name (UID) to the server, it must have already been defined on the computer. If you assign a group (GID) to the server, it must have already been defined on the computer. If you assign a UID or GID that does not exist on the computer, RealServer will not start.

Re-installing the RealServer as an NT Service

If you are upgrading the Windows NT version of the RealServer and want to install it as an Windows NT Service, you may find it easier to manually install the service. Before you begin, make sure you stop the existing **pnserver** service by opening the Windows NT services window from the Windows NT control panel, highlight the Pnserver service and select the stop button.

The following two procedures can be used to accomplish this.

There are two command line utilities located in the RealServer bin directory of your NT Server:

Delsvc.exe is used to manually delete the service.

Crtsvc.exe is used to manually install the RealServer as an NT service

To delete the service:

1. Open a DOS Window and move to the Bin directory of your RealServer installation. For example:

   ```
   c:\real\server\bin
   ```

2. Type **delsvc.exe** and press Enter.

To restore the service:

From the command line type the following:

```
<path>crtsvc.exe <path>pnserver.exe <path>server.cfg
```

For example:

```
crtsvc.exe c:\real\server\bin\pnserver.exe
c:\real\server\server.cfg
```

Chapter 3 Using RealServer

RealServer 5.0 comes with four graphical applications that you can use to start and stop RealServer and to monitor performance: the Administration Page, the Java Performance Monitor, the System Manager, and the Control Center. In addition, command line utilities that perform a subset of these functions are available.

The Administration Page, the Java Performance Monitor and the Server Manager are primarily for viewing server status; Control Center is used to start and stop the server. The Administration Page can stop the server, but cannot start it.

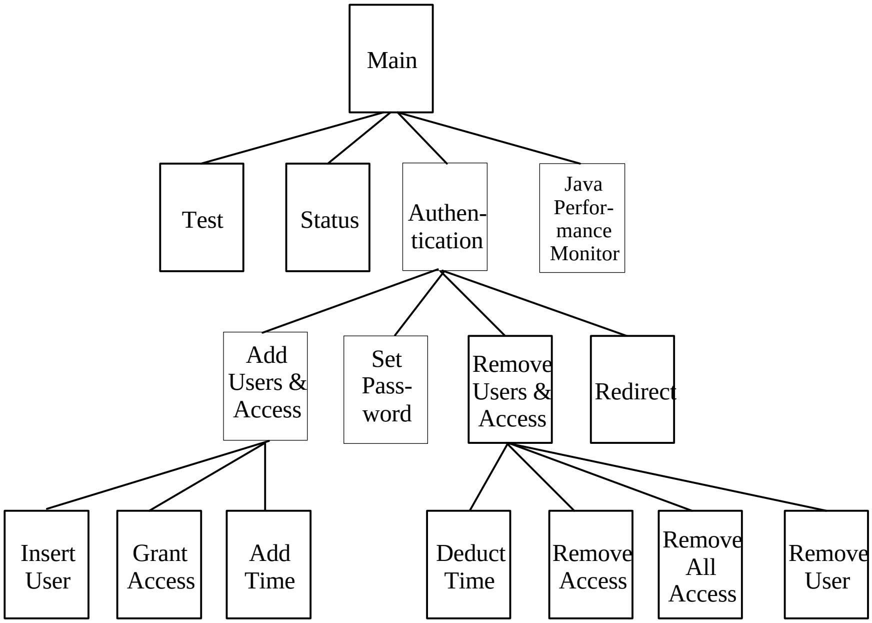
The following table lists the capabilities of the three applications:

	Admin-istration Page	**Java Performance Monitor**	**System Manager**	**Control Center**
Start and stop server	Stop server only	No	No	Yes
Configure server settings	No	Yes	Yes	No
Track number of visitors connected	Yes	Yes	Yes	No
Show number of listeners per clip, bandwidth	No	Yes	Yes	No
View remotely via Web browser on any computer	Yes	Yes	No	No
Must be run on same computer as server	No	No	No	Yes
Operating system	All	All	Windows 95, Windows NT; under UNIX as command line	Windows 95, Windows NT

Windows users can take advantage of all four applications and the command line utilities to perform all functions. On the UNIX platform, the Java Performance Monitor can be used to observe server activity and the command line utilities can be used to start and stop the server.

Using the Administration Page

The Administration Web page contains links to pages that perform the major administrative functions; use this page after you start RealServer.



To use the Administration Page:

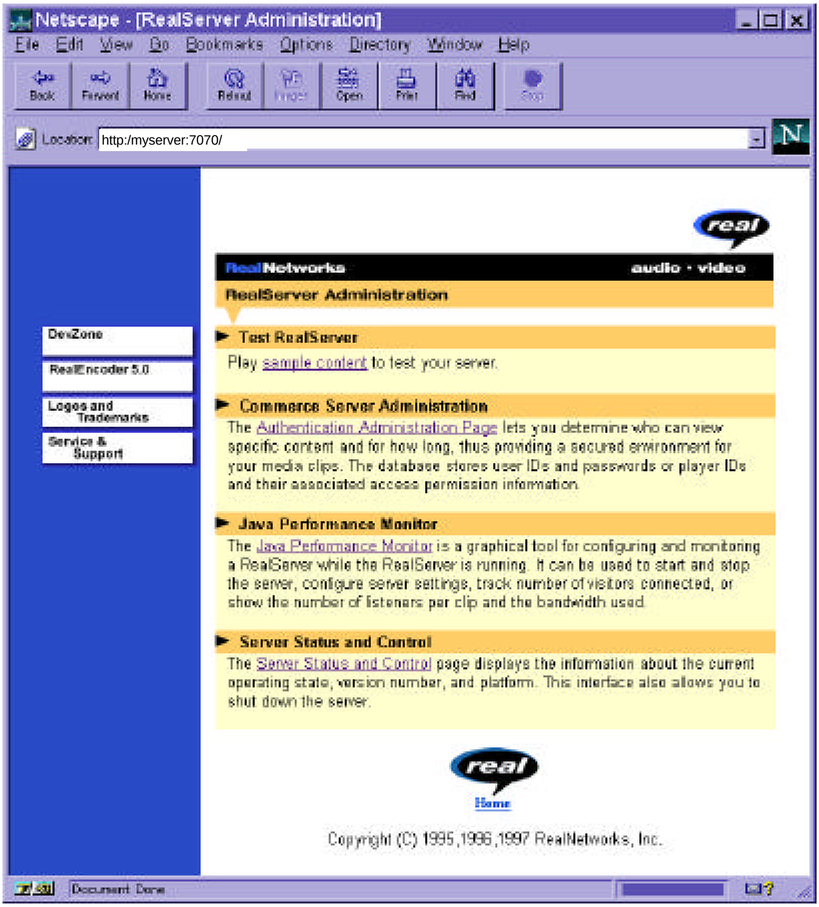
1. Start the RealServer using a method described in the “Starting and Stopping RealServer” section of this chapter.
2. In a Web browser, type the following URL:

```
http://<myserver>:<port>/
```

where **`<my.RealServer.com>`** is the name of your RealServer, and **`<port>`** is the port number as defined in the **server.cfg** file, such as 7070. For information on the **server.cfg** file, see Chapter 4, “Configuring and Maintaining RealServer.”

The user name is “Admin” and the password is defined in the **MonitorPassword** setting in the **server.cfg** file.

3. RealServer displays the Administration Page:


Netscape - [RealServer Administration]
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
DevZone
RealEncoder 5.0
Logos and Trademarks
Service & Support
real
RealNetworks audio · video
Document: Done

Location: http://myserver:7070/

RealServer Administration

Test RealServer

Play sample content to test your server.

Commerce Server Administration

The Authentication Administration Page lets you determine who can view specific content and for how long, thus providing a secured environment for your media clips. The database stores user IDs and passwords or player IDs and their associated access permission information.

Java Performance Monitor

The Java Performance Monitor is a graphical tool for configuring and monitoring a RealServer while the RealServer is running. It can be used to start and stop the server, configure server settings, track number of visitors connected, or show the number of listeners per clip and the bandwidth used.

Server Status and Control

The Server Status and Control page displays the information about the current operating state, version number, and platform. This interface also allows you to shut down the server.

Home

Copyright (C) 1995,1996,1997 RealNetworks, Inc.

Using the Java Performance Monitor

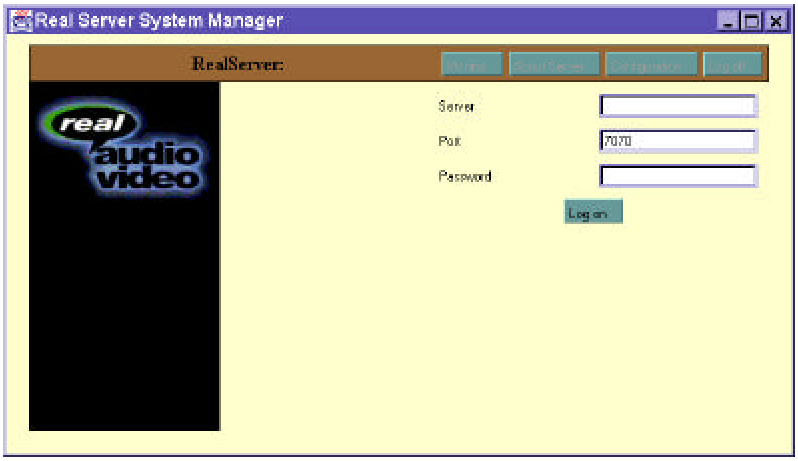
The Administration Page contains a link to the Java Performance Monitor. If you use the Administration Page, you do not need to follow the steps for configuring a Java Performance Monitor, unless you want to be able to monitor any RealServer or want to monitor a RealServer from a computer that doesn’t have a Web browser.

The Java Performance Monitor can be run two ways: as a stand-alone application or as an applet that you view within a Web browser.

If you want to monitor performance on any server, run the Java Performance Monitor as a stand-alone application. If you’re only interested in viewing server activity on a specific Web site, or if you want users to be able to view that site, use the Java Performance Monitor as an applet.

Monitoring Activity of Any Server

Two sets of Java files enable you to use the Java Performance Monitor to monitor activity on any server. One such file, **jview.exe**, is installed with Microsoft Internet Explorer versions 3.0 and later. If you have this file on your computer, you can monitor server activity immediately. If you do not have this file, you’ll need to download files from Sun Microsystems.



Server

Port 7070

Password

Internet Explorer

To use the jview program

1. Locate the **jview.exe** file on your computer. It is typically found in the Windows directory. The Windows directory is typically in the path, but if **jview.exe** is not in a directory that's in the path, you'll need to type its full path in step 3.
2. At a command line, change to the JavaMonitor directory.
3. Type the following:

```
jview monitor
```

The Java Performance Monitor runs as an application. To connect to the server, follow the instructions in "Connecting to the Server," below.

All Others

To use the Java Runtime Environment

1. Download the Java Runtime Environment (JRE) from Sun Microsystems at the following address:

   http://java.sun.com

   Follow the installation instructions on the Web site.

2. At a command prompt, change to the directory where JRE is installed.
3. Type the following:

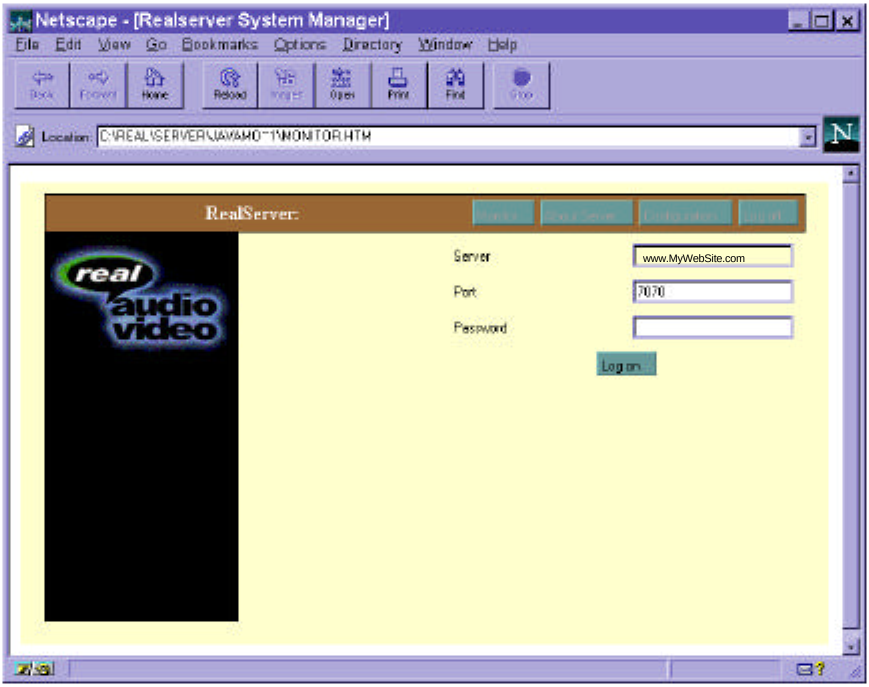
```
jre -cp <path to JavaMonitor directory> monitor
```

The Java Performance Monitor starts as an application. To connect to the server, follow the instructions in “Connecting to the Server,” below.

Monitoring Activity of a Specific Server

RealServer includes the file **monitor.htm** which is preconfigured to display server status when viewed from the RealServer computer.

If you want to view the RealServer computer from any other computer, you must edit **monitor.htm** and ensure that the files in the **JavaMonitor** directory are available to the Web server or ftp server.



Netscape - [Realserver System Manager]

Location: C:\REAL\SERVER\JAVAMO~1\MONITOR.HTM

Server www.MyWebSite.com

Port 7070

Password

Note: If you open the supplied **monitor.htm** without having made any changes, the **Server** box will be blank.

To view this server's performance from any other computer:

1. Ensure that all the **.class** files in the RealServer **JavaMonitor** directory are available to Web browsers pointing to your site. You can copy the entire **JavaMonitor** directory, or you can copy just the **.class** files to a Web server directory.

2. In a text editor, open the file **monitor.htm**, located in the RealServer **JavaMonitor** directory.
3. Look for the APPLET tag:

   ```
   <applet code = "Monitor.class" width = 700 height =
   400>
   ```

4. Add the CODEBASE variable after the CODE variable:

   ```
   CODEBASE = "http://ServerName/JavaClassDirectoryPath/"
   ```

   where “ServerName” is the name of your Web or ftp server and “JavaClassDirectoryPath” is the directory where you copied the .class files in step 1. If your server is an ftp server, you can substitute “ftp” for “http.”

   Note: If you are using an ftp server, you must configure an anonymous account that permits directory read access to the location you specified in the CODEBASE variable.

5. The tag should now look similar to the following example:

   ```
   <applet code = "Monitor.class" CODEBASE =
   "http://www.YourWebSite.com/YourJavaFilesDirectory/"
   width = 700 height = 400>
   ```

6. Save the file under a different name.
7. Put the new .htm file on your Web server.

Connecting to the Server

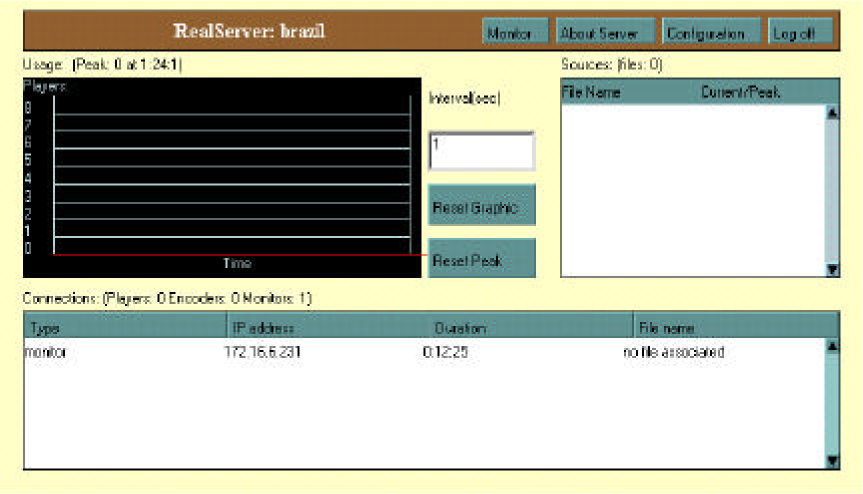
Once you have started the Java Performance Monitor, you’ll use it to view and edit information about your server.

1. In the **Server** box, type the name of the computer running the RealServer you want to monitor.

2. In the **Port** box, type the port number to use to reach RealServer. This is the port specified in the PnaPort setting for the RealServer (the default is 7070).
3. Type the password for the RealServer. (For more information, see **MonitorPassword** in Chapter 4, “Configuring and Maintaining RealServer.”)
4. Click **Log on**.

Monitoring Performance

In the Java Performance Monitor, double-click the **Monitor** button to see a display of your server. The Monitor window appears.



Usage: (Peak: 0 at 1:24:1)

Interval(sec): 1

Souces: (files: 0)

File Name	Current/Peak

Connections: (Players: 0 Encoders: 0 Monitors: 1)

Type	IP address	Duration	File name
monitor	172.16.6.231	0:12:25	no file associated

Usage Window

This area displays the number of players viewing content over time on the RealServer listed at the top of the window.

Interval Window

The Interval Window specifies how often information in the Monitor window is updated. This information is given in minutes.

Reset Graphic Button

The Reset Graphic button acts as a “refresh” or “reload” button, to clear the Usage screen.

Reset Peak Button

To update the peak shown in the Usage and Sources window to its current value, click **Reset Peak**.

Sources Window

The Sources window tells what files are being accessed and the number of times each file is being played. This helps you determine which files are most and least popular, which could help you decide what new fields to add or remove to improve the popularity of your site.

Connections Window

The Connections window indicates how many clients are connected to your site simultaneously.

Using the System Manager

The System Manager can monitor server status of any RealServer for which the **server.cfg** file is accessible.

Windows

In the Windows environment, System Manager provides a graphical display of server information.

Starting the System Manager

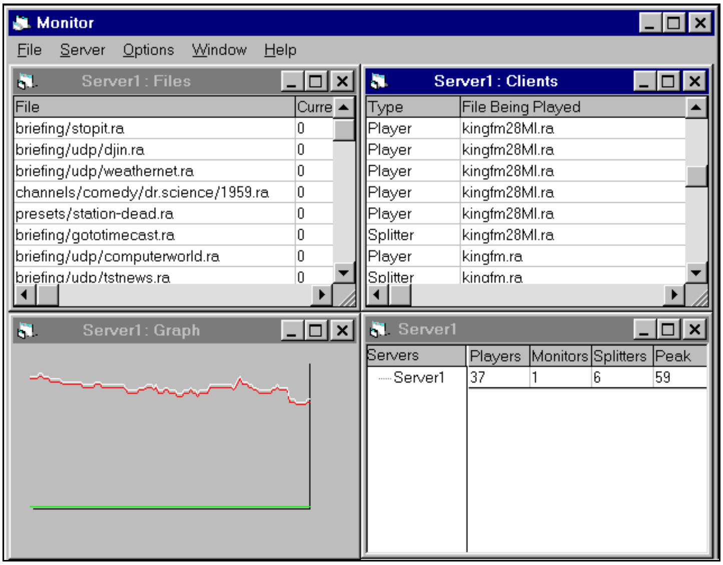
1. **Windows 95 and NT 4.0**: Click the **Start** button, point to **Programs**, point to **Real Server**, and click **RealServer System Manager**.

   Windows NT 3.51: Double-click the Real Server program group and double-click the **System Manager** icon.

2. On the **File** menu, click **Open**.

3. Click the name of the Server that you want to monitor and click **OK**.

4. On the Server menu, click **Clients**, **Files**, or **Graph** to display the type of information you want.



File	Curre
briefing/stopit.ra	0
briefing/udp/djin.ra	0
briefing/udp/weathernet.ra	0
channels/comedy/dr.science/1959.ra	0
presets/station-dead.ra	0
briefing/gototimecast.ra	0
briefing/udp/computerworld.ra	0
briefing/udp/tstnews.ra	0

Type	File Being Played
Player	kingfm28MI.ra
Player	kingfm28MI.ra
Player	kingfm28MI.ra
Player	kingfm28MI.ra
Player	kingfm28MI.ra
Splitter	kingfm28MI.ra
Player	kingfm.ra
Splitter	kingfm.ra

Servers	Players	Monitors	Splitters	Peak
Server1	37	1	6	59

Monitoring Performance

System Manager includes windows that display clients currently connected, files being played, and a graph of all connections for the past two minutes.

Clients Window

You can view the following information in the Clients window:

Column	Description
Type	The type of client connected: Player, Monitor (System Manager), or Encoder.
File Being Played	Name of the file being played from your RealServer.
Domain Name	The domain name or IP address of the client computer. To toggle between IP address and domain name, click **Preferences** on the **Options** box, select the **Client View** tab, and check the **Do DNS Lookups** box.

Column	Description
Elapsed Time	The length of time that the client has been connected to that file since the System Manager has been attached to the Server. This information is also available in the Access Log.

Use the Clients window in System Manager to determine how many clients connect to your site simultaneously. Multiply this number by 10 to 20 Kbps to determine how much bandwidth your RealServer is using.

If you want the Clients window to update continuously, select **Preferences** from the Options menu, select the **Client View** tab and check the **Update Continuously** box.

Files Window

The Files window tells which files are being accessed and the number of times each file is being played. This helps you determine which files are most and least popular, which could help you decide what new files to add or remove to improve the popularity of your site.

Column	Description
File	Name of the file currently being played.
Current	Number of clients currently connected to that file.
Total	Total number of connections made to this file since the System Manager was started.

If you want the Files window to update continuously, click **Preferences** on the **Options** menu, select the **File View** tab and check the **Update Continuously** box.

Graph Window

The Graph window gives a graphical interpretation of selected connections made to your Server in the past two minutes. To control what information appears on the graph, select **Preferences** on the **Options** menu, select the **Graph View** tab and check the boxes for the statistics you want to display.

Resetting the Peak Usage

The **Peak** value in the System Manager display is maintained until you restart the Server or manually reset the value.

To reset the Peak value, click **ResetPeak** on the Server menu.

UNIX

System Manager is a command line program (**rssm**) under UNIX. It does not have a graphical interface. System Manager can monitor a Server running on any platform. Information provided by System Manager includes the number and status of Player connections, System Manager connections, Unknown connections (connections currently being negotiated with the Server), and Total connections. System Manager can also be used to edit configuration file settings.

Using the rssm Program

System Manager runs in two modes: interactive and non-interactive. When the System Manager is in the non-interactive mode, information is automatically appended to **STDOUT** every 5 minutes, unless that time span is modified by the **-l** command. The System Manager accepts commands from the command line; however, it does not prompt for them.

Syntax

```
rssm [-v] [-l <update>] [-p <password>] [-c] [-i]
[-k] <hostname[:port]>
```

where:

-v	Displays the version information of the System Manager. This includes the platform, build and release tags used to identify a particular release.
-l <update>	Sets the update period for output to the screen to update seconds.
-p <password>	Provides the password required by System Manager

to connect to the Server. If this option is not used the System Manager prompts for the password. This feature is not secure. The password is easily accessible to knowledgeable searchers. The password is required each time you want to start monitoring a Server. You can include `-p <password>` in the **rssm** command line in automatic monitoring scripts to avoid having to enter the password interactively.

`-c` Connects to the server to verify it is still accepting connections and then exits. Prints a message if the connection fails and the exit status is non zero.

`-i` Starts interactive mode and permits entry of the commands listed in the command section.

`-k` Does DNS lookups on incoming IP addresses to translate them to full domain names. This command can slow down responses on System Manager. If you are experiencing delays in System Manager information or in response to commands, make sure that this feature is turned off.

`<hostname>` To connect System Manager to a RealServer, set **`hostname`** to the DNS name or IP address of the RealServer. If the server is running on a port other then 7070, specify the port number.

Example

```
rssm -i -p mypassword 203.70.154.100:7070
```

Interactive Commands

The interactive mode is started with the **`-i`** command, which enables the System Manager to print prompts and accept commands from the command line.

After starting System Manager in interactive mode by using the **`-i`** command-line option, you can enter any of the following interactive commands at the System Manager prompt (**`>`**):

Command	Function
c	Displays the current configuration after it has been retrieved using the t command.
e	Resets peak usage value.
g	Displays the time that the peak usage value was last reset.
h or **?**	Prints a list of commands.
i	Prints the Server's version number and platform.
k	Begins collating hostname information for connected clients by doing reverse DNS lookups on the IP numbers provided by the Server.
l	Provides the current list of connected clients.
n	Modifies a Server configuration variable.
o	Prints # of Players, System Managers, unknowns, and total connections to STDOUT every five minutes, or the number of seconds specified by the **-**l option on the command line. This command can be toggled to start and stop.
p	Prints Server license information.
s	Prints a single line of summarized status information.
u	Continuous display. Updates whenever a client status changes.
x	Exit the program.

System Manager displays the information about clients connected to the Server in the following format:

```
<client>  <name>
```

where **<client>** is the type of client connected (Monitor or Player) and **<name>** is the domain name or IP address of that client.

For example, a client listing might look like:

```
monitor 204.71.154.93
```

```
   Player  204.71.153.24
```

If you prefer to receive System Manager information in a report, use the **-l** option and append the output to a file. To do this, use the following command:

```
rssm -l <seconds> <hostname>[:port] >> monitor.txt
```

where **<seconds>** is the number of seconds between reports, **<hostname>** is the name of the computer you are collecting data from, and **monitor.txt** is the name of the report that the information is appended to.

Example

To monitor a Server in interactive mode, with updates every 20 seconds and fully qualified host names for clients, use the following command:

```
rssm -l 20 -k -i yourServer:7070
```

Starting and Stopping RealServer

Before starting or stopping RealServer, you may want to ascertain whether RealServer is running.

Determining Whether RealServer Is Running

Windows

Windows NT: If RealServer is running, it will be listed on the Applications tab of the Windows NT Task Manager.

Windows 95: If RealServer is running, it will be listed on the Taskbar.

UNIX

Determine whether RealServer is running by typing a **ps** command and a **grep** command that searches for **pnserver**—for example,

```
ps -ef | grep pnserver
```

If the server is running, the **ps** and **grep** command return two lines for the pnserver processes that look similar to this:

```
username 25851 25850 0 16:51:11  0:00 bin/pnserver server.cfg
username 25850     1 0 16:51:11  0:00 bin/pnserver server.cfg
```

The lines above indicate that the pnserver process and the pnserver resolver process (which is a child process of the pnserver process) are running. The PIDs for the two processes should be sequential. In the example above, the pnserver process has a PID of 25850, and the pnserver resolver process has a PID of 25851.

Starting RealServer

When you start the RealServer, it asks you to register online. RealServer will not start unless you register.

When you first install RealServer, it is configured to start automatically each time you start your computer. The instructions below describe how to disable this.

Starting RealServer Automatically

Windows

RealServer is installed as a service under Windows NT, where it can be controlled from the Services Control Panel.

If you want to remove RealServer as a service, first stop RealServer, then run the **delsvc.exe** program in the **bin** directory.

To restore automatic starting, run **crtsvc.exe**, which is also located in the **bin** directory.

When you run RealServer as a Service, errors are written to the Windows NT error logs rather than the Error Log specified in the RealServer configuration file. You can view them just like any other Windows NT errors.

UNIX

If you want to disable automatic starting, remove the RealServer command from the boot-time scripts of your UNIX system.

To restore automatic starting, add the **pnserver** command to start RealServer to the boot-time scripts of your UNIX system. The boot-time scripts generally reside in files or directories beneath the **/etc** subdirectory. Be sure to use complete path names in your script.

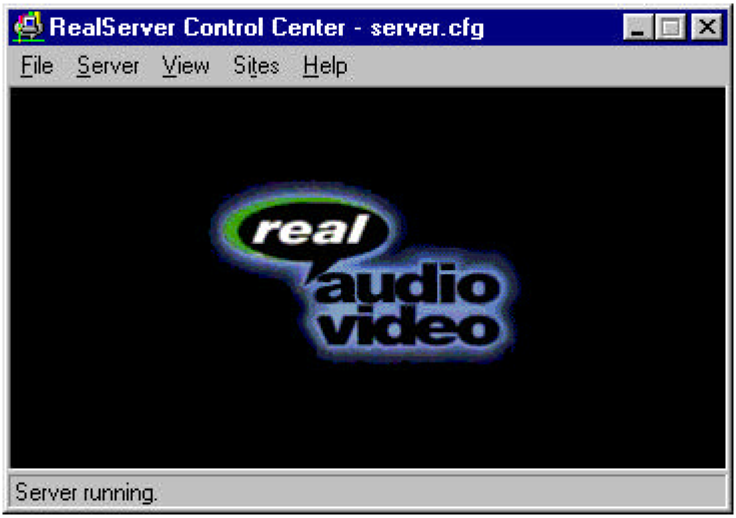
If you do not have permission to change the boot-time scripts on your computer, you may need to have your system administrator do this for you.

Starting RealServer Manually

Windows

To start RealServer manually using the Control Center:

1. On the **Start** menu, point to **Programs**.
2. Point to **RealSystem** and select **RealServer Control Center.** The RealServer Control Center dialog box appears. The default configuration file, **server.cfg**, is automatically loaded. The status bar displays the status of the server.



RealServer Control Center - server.cfg
File Server View Sites Help
real
audio
video
Server running.

To select a different configuration file, click **Open** on the **File** menu. Select a configuration file. Click **Open**.

3. On the **Server** menu, click **Start**.

To start RealServer manually from the command line:

1. At a command line, change to the directory where you installed RealServer.
2. Start RealServer by typing:

```
bin\pnserver server.cfg
```

RealServer does not return any messages to indicate that it has started, and there is no prompt on the screen for as long as it is running.

If RealServer does not start, review the error messages in the RealServer Error Log as described in Chapter 9, “RealServer Log Files.”

UNIX

Because RealServer runs on a high-numbered, unprivileged port, you do not need super-user privileges to start it unless you have configured for PNAviaHTTP by setting HTTPPort 80. However, if you do start it while you are logged on as super-user, then RealServer can configure itself to use additional system resources, such as file descriptors, that it needs to support a large number of users connected simultaneously.

After you start RealServer with super-user privileges and it adjusts its resource limits, RealServer assumes the user and group IDs entered into the configuration file.

To start RealServer manually from the command line:

1. At a command line, change to the directory where you installed RealServer.
2. Type the following:

   ```
   bin/pnserver server.cfg
   ```

   RealServer returns the command prompt and runs in the background. It does not return any messages to indicate that it has started.

If RealServer does not start, review the error messages in the RealServer log files as described in Chapter 9, “RealServer Log Files.”

Stopping RealServer

If you stop RealServer while users are connected, they may receive error messages. To prevent users from receiving error messages, first prevent new users from connecting, then stop the server. If no users are connected, you can skip directly to the instructions listed in “Stopping RealServer.”

Preventing New Users From Connecting

By shutting down RealServer gracefully, you can prevent new connections without disconnecting current users. After your current users have disconnected, stop the RealServer. Be sure to change the **PnaPort** back to its normal value before restarting RealServer.

Java Performance Monitor:

1. In the Java Performance Monitor, double-click **Configuration**.
2. In the **Config Variable Groups** list, double-click **General**.
3. In the **Config Variables** list, double-click **PnaPort**.
4. Change the **PnaPort** configuration setting to an unused value such as 9999.

System Manager:

1. On the **Server** menu, click **Configuration.**
2. Change the **PnaPort** setting to an unused value such as 9999.
3. Click **OK**.

UNIX command line:

1. Change the **PnaPort** configuration setting in **server.cfg** to an unused value such as 9999.
2. Issue the SIGHUP signal.

After the currently connected users stop playing clips, stop the server.

Stopping RealServer

Use either the Administration Web Page or the Control Center to stop RealServer.

To stop RealServer using Control Center, follow the platform-specific directions below.

Windows

If you started RealServer with the **pnserver** utility, press **Ctrl+C** at the command line. If you started RealServer from the Control Center, click **Stop** on the **Server** menu.

UNIX

1. Log on either as super-user or by using the same user ID as RealServer.
2. If you know the process ID, type:

   ```
   kill <processid>
   ```

   If you don’t know the process ID, change to the **pnserver** directory and type:

   ```
   kill 'cat logs/pnserver.pid'
   ```

Network Performance Considerations

A number of factors can interfere with the quality of the media being delivered over the Internet. Media packets can be lost during delivery if they pass through slow routers or if the network is especially busy. Recurrent problems may indicate that you need to modify your connection to your Internet service provider.

To monitor media quality, read the connection statistics in the Access Log to learn more about packets that are early, late, missing, or out-of-order. Also, you should periodically use RealPlayer to listen to the clips on your RealServer.

Open the Statistics window on RealPlayer and monitor the percentage of packet loss that is occurring. If the media quality you experience is poor, it is likely that your users are also experiencing poor media quality.

If you determine that there is a high level of packet loss, consult your Internet provider. You may need a faster Internet connection or there may be other problems with your Internet service.

Troubleshooting RealServer

Testing the Installation

If you are experiencing problems with RealServer, you need to use RealPlayer to test links on your site to isolate the source of the problem. Before you try to connect to your site, launch System Manager to see if your RealServer has an available connection for you to use. If your RealServer has a license that includes Hosting Service, you can use hosting to reserve a stream for your own testing.

The access and Error Logs record errors and information generated by RealServer.

The following questions can help diagnose the problem:

Is RealServer running on the host machine?

Use **ps** (on UNIX), or the Services Control Panel (on Windows NT) to check if RealServer is running. If the Server is not running, start the server.

Is the IP address of the host machine correctly configured in the network routers?

If the Player cannot access the Server over the network, then you cannot expect media to play. Configuring IP address and routers is a complex issue. Contact a networking specialist for help.

Is the machine you are using to test the media connected to the network used by the Server host computer?

You must have a network connection between the RealPlayer and the RealServer for media to play. Contact a networking specialist for help.

Is there a firewall between the Player and the Server?

You need to configure your system's firewalls to permit media to play through them.

Can you connect to the RealServer with the System Manager?

The System Manager application can help you diagnose the problem by validating communications between the Player and RealServer and by letting you view the running state of the connection during attempts to play media.

Is the media file downloading to the Player instead of playing in real time?

Media files cannot be referenced directly by your Web document. Remember that the Web page is being served by a Web server, but the media file is being served by RealServer. The Web page must point the user's Web browser to the media file by way of a metafile, which is a text file you create and save with a .ram extension. The metafile contains the URL of the .ra file located on your RealServer. The Web page contains a link to the metafile.

Is there unreadable text displaying on the screen instead of media?

You have not configured your Web server to recognize RealAudio and/or RealVideo MIME types.

If you still have problems after considering these possibilities, please contact RealNetworks at:

> **www.real.com**

Common Error Messages

The following is a list of the more common error messages you might encounter:

`Could not allocate enough file descriptors to meet capacity. Capacity has been set to <connection number>`

The number of simultaneous connections has exceeded the capacity of your operating system. RealServer has automatically reset the number of media connections allowed to connect.

`Invalid license key or information`

Either you have not specified any licensing information for the **LicenseKey** setting in **server.cfg** or the licensing information you entered was incorrect. Check to make sure the information was entered exactly as you received it.

`This license is for another platform`

The license information you specified for the **LicenseKey** setting in server.cfg is for a different operating system. Check to make sure that you installed RealServer on the proper machine.

`Server cannot be started before <date>`

The RealServer license you purchased does not become valid until the date listed. Because RealServer requires a valid license to operate, your RealServer will not start until the date listed.

`Server cannot be started after <date>`

The RealServer license you purchased is not valid after the date listed. RealServer requires a valid license to operate.

`Your license does not support ISP Hosting.`

You configuration file contains Hosting Service settings, but you license does not include Hosting Service. The Hosting Service configuration settings are ignored.

`You must restart the server for this change to take effect.`

You have made a change in the **server.cfg** file that will not take effect until you restart RealServer.

`Out of Memory`

RealServer is unable to dynamically allocate enough memory to create a new connection or manage existing connections. If you receive an Out of Memory message, you may require additional memory or you may need to add swap space for your RealServer machine to use for dynamic memory allocation.

`Error retrieving <file name>`

A user tried to access a file and the file could not be found. The user may have supplied the wrong URL and the Server rejected the request. However, if you see this more than once for the same file, you should check your metafile to ensure that the URL pointing to the file is accurate.

`SIGPIPE Received, code:13 (UNIX systems only)`

The SIGPIPE signal is sent to RealServer by the operating system when the client abruptly drops the connection. No action is required for this message.

`Error retrieving URL <file name> (Codec error)` `Error retrieving URL <file name> (Insufficient bandwidth)`

The Player requested a file for which it does not have the correct CODEC installed or for which it does not have sufficient bandwidth to play. For example, a RealAudio Player 2.0 requesting RealAudio 3.0 content generates this error message.

Chapter 4 Configuring and Maintaining RealServer

This chapter introduces the server configuration file (**server.cfg**), which contains all the configuration settings for RealServer. When you want to change an aspect of server operation, you must edit the configuration file. A default server configuration file is created in the **Real\Server** directory on Windows NT, and in the **pnserver** directory on UNIX during installation.

The configuration file stores pairs of configuration options and their settings. This is a plain-text file, but settings are case-sensitive such as:

```
LocalHost           Matisse
LiveFilePassword    fauvist
EncoderTimeout      30
```

After making changes to the server configuration file, restart the server so that the changes can take effect.

Editing the Configuration File

There are three tools you can use to edit the configuration file:

- Java Performance Monitor
- System Manager
- Any text editor

With the RealServer Administration page, you can access the Java Performance Monitor automatically.

You can edit the configuration file using the Java Performance Monitor or the System Manager only when the server is running. With a text editor, you can edit the configuration file whether the server is running or not.

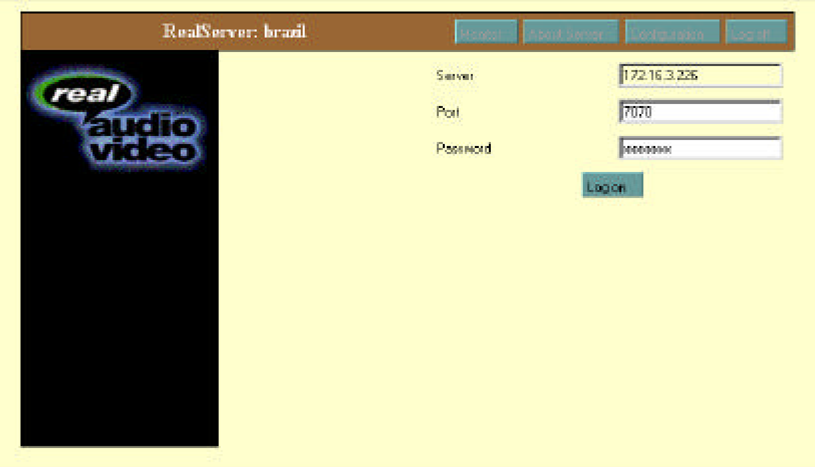
Refer to the appropriate section below for editing procedures for each editing tool.

Editing the Configuration File with the Java Performance Monitor

The Java-based Performance Monitor is a graphical tool for configuring and monitoring a RealServer while the RealServer is running. For a discussion of how to start the Java Performance Monitor and connect it to the server, see Chapter 3, "Using RealServer."

To use the Java Performance Monitor to change RealServer configuration settings:

1. Start the Java Performance Monitor and log on to the server.



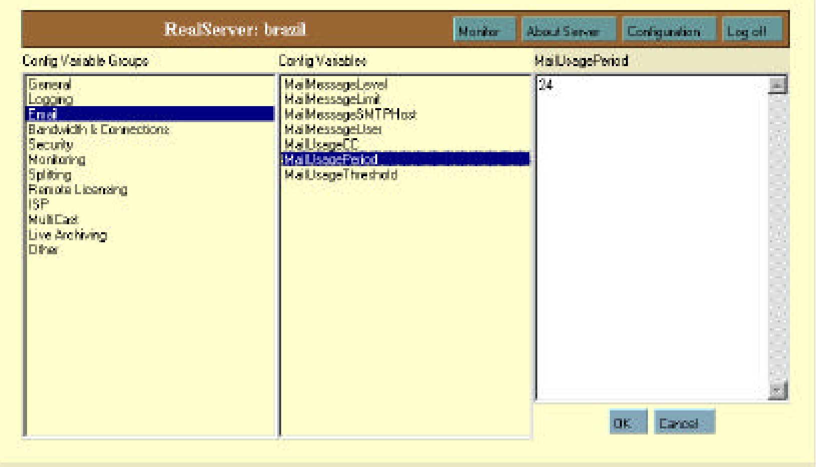
Server 172.16.3.226

Port 7070

Password ********

2. Click **Configuration**.
3. In the **Config Variable Groups** list, double-click the category you want to change. The **Config Variables** shows all configuration options, even those that do not have a value assigned.

4. Double-click the setting you want to change. The value appears on the far right.



RealServer: brazil

Config Variable Groups

- General
- Logging
- Email
- Bandwidth & Connections
- Security
- Monitoring
- Splitting
- Remote Licensing
- ISP
- MultiCast
- Live Archiving
- Other

Config Variables

- MailMessageLevel
- MailMessageLimit
- MailMessageSMTPHost
- MailMessageUser
- MailUsageCC
- MailUsagePeriod
- MailUsageThreshold

MailUsagePeriod

24

5. Edit the setting you want to change and click **OK**.

   Repeat the previous two steps until you are satisfied with your RealServer configuration.

6. When you have finished editing Configuration settings, click **Monitor** to close the Configuration window and return to the Java Performance Monitor.

Restart the RealServer to cause the new configuration settings to take effect.

Editing the Configuration File with the System Manager

System Manager is a graphical tool under Windows and a command-line tool under UNIX (**rssm**).

For a discussion of starting the System Manager, see Chapter 3, “Using RealServer.”

For instructions on using **rssm**, see Appendix A, “Server Commands.”

To use the System Manager to change RealServer configuration settings under Windows:

1. Start System Manager.
2. On the **Server** menu, click **Configuration**.
3. Edit the setting or settings you want to change.
4. When you have finished editing Configuration settings, click **OK**.

Restart the server for the new configuration settings to take effect.

Editing the Configuration File with a Text Editor

You can edit the configuration file with any text editor. When RealServer is not running, editing the configuration file with a text editor is the only way to change settings.

Windows

1. Using a text editor such as Notepad, open the **server.cfg** file located in the **Real\Server** directory.
2. Edit the entry or entries you want to change.
3. Save the file.
4. If RealServer was running when you made changes, restart the server for the changes to take effect.

Note: The configuration file must be saved in a text only format and no line feeds should be included in the file.

UNIX

1. Using a text editor such as **vi**, open the **server.cfg** file located in the **pnserver** directory.
2. Edit the entry or entries you want to change.
3. Save the file as text.
4. To force the new configuration settings to reload, restart the RealServer, or use the following command:

```
kill -HUP 'cat pnserver.pid'
```

Configuration File Settings

The server configuration file (**server.cfg**) controls both the basic and the optional features of RealServer.

There are six groups of settings in the **server.cfg** file:

- General
- Network
- Ad Insertion
- Authentication
- Live broadcasting
- Multicasting

Each group controls a different aspect or feature of your server. Some features require a license in order to be enabled. Detailed descriptions these groups follow.

Note: Because the server file is crucial to the operation of the RealServer, be sure to place it in a location that is inaccessible to any unauthorized users.

General Settings

General settings control the basic operation of RealServer. RealServer is installed with default configuration settings that should work on your computer.

After installing RealServer, you do not have to change any of these settings for RealServer to operate properly. You may find it useful, however, to customize RealServer for your particular requirements and environment.

General settings can be divided into the following groups:

- **Ownership Information—**controls miscellaneous aspects of the server operation.

- **Firewall Settings—**configures the RealServer's IP address and port settings to work with a Web server or firewall.

- **Logging Settings—**specifes what information the server logs and where this information is logged.
- **Notification Settings—**configures e-mail notification for the server, including operation thresholds that generate e-mail messages from the server when exceeded.
- **Bandwidth and Connection Volume—**limits the bandwidth used by the server and the number of simultaneous client connections to the server.
- **IP-Based Access Control (Intranet Solution only)**— allows only computers with certain IP addresses to contact the RealServer.
- **Monitor Settings—**controls the Java Performance Monitor or System Manager access to the server.

Ownership Information

This tables summarizes the general configuration settings that control miscellaneous aspects of RealServer operation. A detailed description of each setting follows the table. To change these settings, double-click General on the Config Variable Groups list.

Setting	Description
BasePath	The path to the root directory of your media files.
CustomerName	Your customer name, as received from RealNetworks. Must be correct for the server to operate.
Group	Default group ID (GID) for server. (UNIX only)
LicenseKey	Encrypted license string from RealNetworks. Must be correct for the server to operate.
LocalHost	Fully-qualified name that overrides the system default name.
MaxThreads	Limits the number of threads used by the server.
MinPlayer Protocol	The minimum PN protocol that a Player must have to connect to the server.
PnaPort	The TCP/IP port on which the server listens for

Setting	Description
	connection requests from clients.
PIDPath	The file that records the server's process ID. (UNIX only)
Timeout	The number of seconds the server waits before disconnecting an inactive Player.
User	The default user ID (UID) for the server. (UNIX only)
RestoreOriginal PrivilegeOnReload	Allows user logged on with name specified by **User** setting to issue the SIGHUP command to load changed configuration file settings, even if the server has already been started by someone with Root permissions.
HTTPPort	Enables players that may be behind firewalls to access content served from your server.
ResolverPort	The TCP port to use for resolving DNS addresses.
IOBufferSize	Determines the amount of memory the RealServer allocates for each sequential read of a RealFlash (.swf) content file.
MobilePlayback OversendRate	The multiple of the streaming rate that is used to determine the effective bit rate at which the server transfers files to Mobile Playback enabled RealPlayers via TCP.

BasePath

Path to root directory of your media files. Most media content delivered by your RealServer resides in, or in a subdirectory of, the directory specified by **BasePath**.

By default**, BasePath** points to the **content** subdirectory of your RealServer installation directory, since this directory contains several sample documents. For further information about organizing media content see “Bandwidth Negotiation” in the *RealAudio and RealVideo Content Creation Guide*.

Default value	content

Range of values	Valid directory names

Syntax

```
BasePath <path>
```

Example

Windows

```
BasePath c:\real\server\content
```

UNIX

```
BasePath /user/local/pnserver/content
```

CustomerName

Name specified in the license you received from RealNetworks (by e-mail) or from your RealNetworks reseller. For your RealServer to function, you must enter this parameter. When prompted for Customer Name and License Key, use copy and paste.

If you purchase a new or upgraded license, type the new CustomerName and LicenseKey values that you receive from RealNetworks or from your RealNetworks reseller.

Default value	(none)
Range of values	Valid alpha-numeric string

Syntax

```
CustomerName <licensename>
```

Example

```
CustomerName Very Big Corporation
```

DefaultErrorFile

RealServer sends a Player an error message when a requested file is not available. If you set the **DefaultErrorFile** setting, RealServer plays the specified media file instead of sending the error message.

The path to your error file should be an absolute path. Your error file should be an audio file recorded in 14.4 format and indicate that there was a format compatibility problem. For example, “We are sorry but the file requested is not available in your Player’s format. Please try another file.”

Note: You can specify a video (.rm) file for **DefaultErrorFile** rather than an audio (.ra) file. However, because RealAudio Players 3.0 (or earlier) will not be able to play a video file, this is not recommended.

Default value	**error.ra**
Range of values	Media file name and path

Syntax

```
DefaultErrorFile <path>
```

Example

Windows

```
DefaultErrorFile pnserver/content/nofile.ra
```

UNIX

```
DefaultErrorFile c:\real\server\content\nofile.ra
```

Group (UNIX only)

Default group name for RealServer for UNIX. The group name must exist on the computer on which RealServer is running; otherwise, RealServer will not start.

If you do not specify a group name when installing RealServer, the group name defaults to the group name of the user who first starts RealServer.

Default value	(none)
Range of values	Valid user and group names

Example

```
Group users
```

LicenseKey

Encrypted license string enabling your RealServer to operate. The default license allows two streams. For your RealServer to operate, you must type the license key exactly as you received it from RealNetworks or your reseller; use copy and paste to ensure there are no errors.

Default value	none
Range of values	N/A

Syntax

```
LicenseKey <encryptedkey>
```

If you downloaded your software, **`encryptedkey`** is in the e-mail message that gave you access to the download URL. If you purchased your software on CD-ROM, **`encryptedkey`** is provided via e-mail from RealNetworks or is affixed to your CD-ROM case or RealServer software package.

If you purchase a new or upgraded license, you must type the new CustomerName and LicenseKey values that you receive from RealNetworks or from your RealNetworks reseller.

Example

```
LicenseKey 43819m554420998372983729857298752983758hf2
           938299192384j6esu3829879298
```

LocalHost

Fully-qualified name that overrides the system default domain name. If you experience problems running multiple processes, you can set the **LocalHost** parameter in your RealServer configuration file.

On some platforms, the system does not return a fully-qualified domain name, which causes difficulty for RealServer in locating other RealServers in a multiprocessing configuration. With the **LocalHost** parameter, you can override the system default domain name and provide RealServer with a fully-qualified domain name.

Default value	(none)
Range of values	Valid domain name

Syntax

```
LocalHost <domain name>
```

Example

```
LocalHost mycomputer.mydomain.com
```

MaxThreads

Maximum number of threads or processes. This entry lets RealServer run multiple processes within a single machine. RealServer can take advantage of multiple CPUs.

Note: This configuration parameter affects your computer’s CPU usage.

Default value	1
Range of values	Integers greater than zero

Syntax

```
MaxThreads <count>
```

Example

```
MaxThreads 5
```

MinPlayerProtocol

The minimum protocol supported by RealServer. Players that do not supply a protocol number equal to or greater than this value as part of their connection information cannot connect to RealServer.

Default value	0
Range of values	**0** All players **4** RealAudio Player 1.0 and later (same a 0) **7** RealAudio Player 2.0 and later **9** RealAudio Player 3.0 only **10** RealPlayer only

Syntax

```
MinPlayerProtocol <number>
```

Example

To allow only RealAudio 2.0 and later players, type:

```
MinPlayerProtocol 7
```

PnaPort

Number of the TCP port RealServer uses for receiving requests from clients. The only reason to use a port other than the default is to allow several Servers to coexist on one system, or to achieve some level of privacy when serving information by using an unusual port number.

Default value	7070
Range of values	Valid port number

Syntax

```
PnaPort <number>
```

Note: To use a port lower than 1024 on a UNIX system, you need to be logged on as super-user.

Example

```
PnaPort 7074
```

PidPath (UNIX Only)

File used by RealServer for UNIX to record its process ID. If you do not specify a PidPath, RealServer records its process ID in **pnserver/logs/pnserver.pid**.

For administration, the process ID file should reside in the same directory as your access and error log files.

Default value	pnserver.pid
Range of values	Valid path and file name

Syntax

```
PidPath <path/filename>
```

Example

```
PidPath /pnserver/logs/pnserver.pid
```

Timeout

Number of seconds RealServer waits before disconnecting an inactive Player. Because every connection consumes valuable resources, connections should not be permitted to sit idle for long periods of time. A connection is idle when the Player has paused playing of media or has reached the end of the media program without disconnecting. The client can automatically reconnect after being timed out by RealServer if the user clicks the **Play** button.

Default value	300
Range of values	120 - 900

Syntax

```
Timeout <Seconds>
```

Example

```
Timeout 240
```

HTTPPort

Enable Players that may be behind firewalls to access content from your server. If not set, the port is not opened and HTTP cloaking is not available.

Default value	80
Range of values	none or 80

Syntax

```
HTTPPort <port>
```

Example

```
HTTPPort 80
```

ResolverPort

The TCP port to use for resolving DNS addresses.

Default value	PnaPort + 1
Range of values	Valid port number

Syntax

```
ResolverPort <port>
```

Example

```
ResolverPort 8081
```

User (UNIX only)

Default user name for RealServer for UNIX. The user name must exist on the computer on which RealServer is running; otherwise, RealServer will not start.

If you do not specify a user name when installing RealServer, the user name defaults to the user name of the user who first starts RealServer.

Default value	Username of first person to log in
Range of values	Valid user and group names

Syntax

```
User <UserName>
```

Example

```
User fredk
```

RestoreOriginalPrivilegeOnReload (UNIX only)

When set to True, allows user logged on with name specified by **User** setting to issue the SIGHUP command to load changed configuration file settings, even if the server has already been started by someone with Root permissions. If **RestoreOriginalPrivilegeOnReload** is set to False, only the person who started the RealServer can issue the SIGHUP command to reload the new settings without restarting the Server.

Default value	False
Range of values	True, False

Syntax

```
RestoreOriginalPrivilegeOnReload <value>
```

Example

```
RestoreOriginalPrivilegeOnReload False
```

IOBufferSize

This setting is specific to RealFlash media. It determines the amount of memory the RealServer allocates for each sequential read of a RealFlash (.swf) content file. Higher values increase the RealServer’s streaming performance, but higher values also reduce the amount of memory available to the system.

RealFlash is an optional feature controlled by the license you purchase from RealNetworks.

Default value	4096
Range of values	4K to 64K

Syntax

```
IOBufferSize <size>
```

Example

```
IOBufferSize 4096
```

MobilePlaybackOversendRate

The multiple of the streaming rate that is used to determine the effective bit rate at which the server transfers files to Mobile Playback enabled RealPlayers via TCP. The file is sent via TCP at the effective bit rate or the network bandwidth, whichever is lower.

Default value	10
Range of values	1-100

Syntax

```
MobilePlaybackOversendRate <number>
```

Example

To serve 6.5 Kbps encoded files at 19.5 Kbps:

```
MobilePlaybackOversendRate 3
```

Firewall Settings

Smart Networking allows RealServer to stream content to a RealPlayer behind a firewall that does not permit TCP/UDP transmissions. This is accomplished through a HTTP-like protocol. If the Web server is on the same computer as the RealServer and has a second IP address available, Smart Networking requires two settings.

Setting	Description
HTTPPort	If no Web server is running on the machine that is running RealServer, the default is to perform HTTP transmission on port 80. The RealServer can also be configured to perform HTTP transmission on other ports.
IPBindingList	For security or convenience in running a Web server and RealServer on the same machine, you can configure the server to transmit on an alternate IP address.

HTTPPort

Alternate ports to which the RealServer can be assigned to stream via HTTP. If no Web server is running on the machine that is running RealServer, the server can be configured to perform HTTP transmission by adding this setting to your **server.cfg** file.

Although you can use any valid TCP/IP port number, RealPlayer will try to connect to port 80. If port 80 is not available, the server will not perform HTTP transmissions.

Note: To use port 80 on UNIX, the server must be started as root (super-user).

Default value	(none)
Range of values	Valid port numbers

Syntax

```
HTTPPort <port number>
```

Example

```
HTTPPort 80
```

IPBindingList

Use when the RealServer and the Web server are installed on one machine that has multiple IP addresses. The **IPBindingList** reserves an IP address (or addresses) for RealServer's exclusive use.

This setting may be used in conjunction with Smart Networking. For example, if you have only two IP Addresses available for the computer on which the RealServer and Web server are installed, refer to one IP address with the **IPBindingList** setting. For additional information, see Chapter 2, "Installing and Configuring RealServer."

Note: Note that Microsoft Internet Information Server (IIS) version 3.0 will bind to all IP addresses, regardless of this setting. IIS version 4.0 does allow binding to multiple IP addresses.

Default value	(none)
Range of values	Valid IP address(es)

Syntax

```
IPBindingList [{<IPAddress1>}, {<IPAddress2>},
{<IPAddress3>},...]
```

where **`<IPAddress>`** is a valid IP address that has been obtained and associated with the host machine.

Example

```
IPBindingList [{174.16.32.60}]
```

To set up virtual IP addresses in Windows

1. On the **Start** menu, point to **Settings**, then select **Control Panel**.
2. Double-click **Network**.
3. On the **Protocol** tab, select **TCP/IP**, and click **Properties**.
4. If you have two ethernet cards, select the other card and change the last digits of the IP address.

   If you have one ethernet card, click **Advanced**, then click **Add**. Type the new IP address. Click **Add**.
5. Restart your computer.

To set up virtual IP addresses in UNIX

The procedure for setting up virtual IP addresses on UNIX depends upon the operating system. Refer to your operating system manual for more information. If your system does not automatically route traffic to the new address, be sure to manually route to the virtual IP address.

Logging

You can specify what information RealServer tracks about client connections and errors. You can also specify where RealServer logs this information.

The following table summarizes the access and error log settings. A detailed description of each setting follows the table.

Setting	Description
ErrorLogPath	The path and file name of the error log file.
LogPath	The path and file name of the access log file, which logs information about client access to RealServer.
LoggingStyle	Type and amount of information to capture.
StatsMask	Specifies additional access log statistics to request from Players.

ErrorLogPath

Path and filename of the file storing errors that occur during the operation of RealServer. If the ErrorLogPath setting is not specified in **server.cfg**, RealServer records errors in the Error Log file located in the same directory as pnserver.

Default value	logs/pnerror.log
Range of values	Valid path and file name

Syntax

```
ErrorLogPath <path>
```

Example

Windows

```
ErrorLogPath c:\real\server\logs\pnerror.log
```

UNIX

```
ErrorLogPath /pnserver/pnerror
```

LogPath

Path and file name for the access log file. Uses a relative path from the directory from which RealServer was started. RealServer logs information regarding every access to your Server into the file specified by the LogPath.

During installation, LogPath is set to **pnaccess.log** or **pnaccess** in the logs subdirectory of your RealServer directory. If the LogPath setting is not specified in **server.cfg**, RealServer records access information in the **pnaccess.log** file located in the same directory as pnserver.

Default value	logs/pnaccess.log
Range of values	relative path

Syntax

```
LogPath <path>
```

Example

Windows

```
LogPath c:\real\server\logs\pnaccess.log
```

UNIX

```
LogPath /logs/pnaccess
```

LoggingStyle

Specifies what format to use for the Access Log. Used with **StatsMask**. The **StatsMask** parameter specifies which additional information is included when **LoggingStyle** is set to 1 or 2. Style 2 adds a unique player ID to style 1.

Default value	0
Range of values	0 **-** Original RealAudio Format 1 **-** RealAudio 3.0 2 **-** RealServer 4.0 3 **-** RealServer 5.0

Syntax

```
LoggingStyle <value>
```

Example

```
LoggingStyle 1
```

StatsMask

Specifies which additional Access Log statistics to request from Players. These statistics are included in the Access Log only when the **LoggingStyle** configuration parameter is set to 1. Detailed information on the statistics types is available in Chapter 9, "RealServer Log Files."

Default value	0	
Range of values	**0**	No additional statistics
	1	Statistics type 1 only
	2	Statistics type 2 only
	3	Both statistics types 1 and 2
	4	Statistics type 3 only
	5	Both statistics types 1 and 3
	6	Both statistics types 2 and 3
	7	All statistics (types 1, 2, and 3)

Syntax

```
StatsMask <value>
```

Example

```
StatsMask 3
```

Note: Statistics type 2 are returned only by RealAudio Player 3.0. Statistics type 3 are returned by RealPlayer 5.0.

Notification Settings

You can configure RealServer to send e-mail messages that alert you to important RealServer events. You can specify the following aspects of e-mail operation:

- One or two addresses to which RealServer sends e-mail messages

- The SMTP mail server that RealServer uses to send e-mail messages
- The types of messages (informational, warning, or error) that RealServer sends
- The number of times RealServer sends an e-mail message about a particular event (such as a usage threshold being crossed)
- A RealServer usage threshold that, when exceeded, causes RealServer to notify you by e-mail

Configuring Basic E-mail Operation

The following configuration settings control RealServer notification:

Setting	Description
MailMessageLevel	Minimum severity level for e-mail messages sent by RealServer
MailMessageLimit	Maximum number of e-mail messages to sends about a particular event.
MailMessageSMTPHost	The mail server that RealServer uses to send e-mail messages.
MailMessageUser	E-mail address of the primary contact for RealServer e-mail messages.
MailUsageCC	E-mail address of the secondary contact for RealServer e-mail messages

MailMessageLevel

Specifies the severity of messages that are e-mailed to the system administrator. Specifying a level prevents RealServer from sending messages with a lower severity. The default value, which is no value, prevents RealServer from sending e-mail messages.

Default value	(none)
Range of values	INFO, WARNING, ERROR

Syntax

```
MailMessageLevel <level>
```

Example

```
MailMessageLevel WARNING
```

Sends e-mail about WARNING and ERROR messages, but not INFORMATION messages.

MailMessageLimit

The number of times that RealServer sends a specific e-mail message. Limits the number of times RealServer notifies the system administrator of the same problem. You can disable RealServer e-mail messaging by setting **MailMessageLimit** to 0.

Default value	5
Range of values	Integers greater than or equal to 0

Syntax

```
MailMessageLimit <number>
```

Example

```
MailMessageLimit 3
```

MailMessageSMTPHost

The e-mail server that RealServer uses to send e-mail messages.

Default value	(none)
Range of values	Valid Domain Name System (DNS) name or IP address

Syntax

```
MailMessageSMTPHost <address>
```

Example

```
MailMessageSMTPHost mail.mycorp.com
```

MailMessageUser

The e-mail address of the primary contact, or system administrator, to which the e-mail messages will be sent. You can use a group email ID to broadcast the message to several addresses.

Default value	(none)
Range of values	Valid e-mail address

Syntax

```
MailMessageUser <address>
```

Example

```
MailMessageUser sysadmin@mycorp.com
```

MailUsageCC

The secondary e-mail address to receive e-mail from RealServer.

Default value	sales@real.com
Range of values	Valid e-mail address

Syntax

```
MailUsageCC <address>
```

Example

```
MailUsageCC support@corp.com
```

Configuring Threshold E-mail

RealServer can send notification messages when a percentage of RealServer's bandwidth or connection limits is exceeded. These two thresholds are set with **MaxBandwidth** setting (see page 85) and **ClientConnections** setting (see page 84).

Setting	Description
MailUsagePeriod	Specifies the time period over which **MailUsageThreshold** is calculated.

MailUsageThreshold	Specifies the server usage level that causes RealServer to send a usage e-mail message.

The notification e-mail lists the number of licensed streams, the threshold exceeded, and the time for which the threshold was exceeded. A sample email message that is sent when bandwidth threshold is exceeded:

```
In last 89 hours, server usage exceeded 92% for
a total of 5342 seconds
Licensed Streams: 812
>--- BANDWIDTH THRESHOLD exceeded ---<
Maximum Bandwidth: 413
Bandwidth Used: 398
Maximum Audio Connections: 750
Number of Connections Used: 732
```

MailUsagePeriod

Specifies the time period over which **MailUsageThreshold** is calculated. At the end of each **MailUsagePeriod**, the server resets the statistics used to calculate **MailUsageThreshold**. RealServer resets the **MailUsageThreshold** statistics every week.

Default value	24
Range of values	Positive Integers

Syntax

```
MailUsagePeriod <hours>
```

Example

```
MailUsagePeriod 168
```

MailUsageThreshold

Percentage of maximum bandwidth (if specified) or total license streams that will generate a usage threshold e-mail.

If **MaxBandwidth** is set to a value, **MailUsageThreshold** is a percentage of the **MaxBandwidth** value.

If no maximum bandwidth is specified, **MailUsageThreshold** is a percentage of the **ClientConnections** value.

If **ClientConnections** is not set, **MailUsageThreshold** is a percentage of RealServer's licensed streams.

Default value	80
Range of values	1 to 100

Syntax

```
MailUsageThreshold <percent>
```

Example

```
MailUsageThreshold 85
```

Bandwidth and Connection Volume

RealServer lets you control how much of your network resources are dedicated to media by letting you specify how much bandwidth RealServer can use. You can also control how many clients can simultaneously connect to RealServer.

A detailed description of each setting follows the table.

Setting	Description
ClientConnections	The maximum number of simultaneous client connections to RealServer.
MaxBandwidth	The maximum bandwidth that RealServer can use on particular network connection.

ClientConnections

Maximum number of simultaneous client connections. The number of connections that RealServer can support is determined by the license you purchase unless you are using a remote license server. You can set a lower limit than that specified by your license.

The maximum number of connections cannot exceed the maximum number that the bandwidth of your network connection supports. If **ClientConnections** is not specified, RealServer uses the number of streams specified by your license. If a **LicenseKey** is not specified or if your license key expires, RealServer allows a minimum of 2 streams.

A value of 0 means to use the maximum number of streams allowed by the license key. To specify a greater number of connections than the license on this computer allows, obtain additional streams from a remote license server.

Default value	Licensed number of streams
Range of values	0, 1 - 32767

Syntax

```
ClientConnections <integer>
```

Example

```
ClientConnections 750
```

MaxBandwidth

Maximum bandwidth (in Kbps) that RealServer can use out of the total bandwidth capacity of a particular network connection.

Default value	0
Range of values	Positive Integers

The value 0 means bandwidth is limited indirectly by the number of simultaneous streams that RealServer can support. The maximum number of simultaneous streams is determined by the lesser of **ClientConnections** and the number of licensed streams.

Syntax

```
MaxBandwidth <number>
```

Example

```
MaxBandwidth 750
```

Restricts RealServer to using half of a T1 connection’s capacity.

IP-Based Access Control

Aside from using a firewall, there are two ways that you can control access to media files served by RealServer:

- Put URLs for media files on restricted Web pages.
- Use the **ConnectControlList** setting to specify which network domains can access RealServer.

If you purchased an intranet license for RealServer, you must specify both these settings to enable splitters on your intranet to access your RealServer.

Setting	Description
ConnectControlList	Addresses from which clients are allowed to access RealServer.
SplitterControlList	Splitter domain names that are allowed to access RealServer.

ConnectControlList

Specifies the IP addresses allowed to access RealServer. If you purchased an intranet license for RealServer, you must specify a **ConnectControlList** to enable users on your intranet to access your RealServer.

If you purchased an Internet license for RealServer, you can optionally use this setting to restrict access to your Server.

Default value	(none)
Range of values	**`<address>`** is the domain address of the computer allowed to access RealServer. All four octets of the address must be specified. **`<net mask>`** is a mask that specifies which bits in the domain address are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards.

Syntax

```
    ConnectControlList [{<address>, <net mask>},...]
```

Example

```
ConnectControlList [{121.23.101.0, 255.255.255.0}]
```

This entry accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the netmask in this example were 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted.

The net mask 255.255.255.255 accepts only the single IP address specified.

Note: Servers with intranet licenses cannot specify a net mask of 0.0.0.0.

To allow any player to connect, do not include a **ConnectControlList** setting in your configuration file. To prevent any player from connecting, specify:

```
ConnectControlList [{0.0.0.0, 255.255.255.255}]
```

SplitterControlList

Lists the splitter domain names that are allowed to access RealServer. To use this setting, you must purchase splitting as part of your RealServer.

Default value	(none)
Range of values	**`<address>`** the domain address of the splitter computer allowed to access RealServer. **`<net mask>`** specifies the bits in the domain address that are treated as wildcards.

Syntax

```
SplitterControlList [{<address>, <net mask>}, ...]
```

The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards.

Example

```
SplitterControlList [{121.23.101.0, 255.255.255.0}]
```

This example accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128 all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted.

The net mask 255.255.255.255 accepts only the single IP address specified.

Note: Servers with intranet licenses cannot specify a net mask of 0.0.0.0.

To allow any player to connect, do not include a **SplitterControlList** setting in your configuration file. To prevent any player from connecting, specify:

```
SplitterControlList [{0.0.0.0, 255.255.255.255}]
```

Monitor Settings

You can specify the maximum number of Java Performance Monitor or System Manager sessions that can connect to RealServer at the same time. You can also specify a password that Java Performance Monitor must use to connect to RealServer.

Setting	Description
MonitorConnections	Maximum number of Java Performance Monitor sessions that can connect to RealServer.
MonitorPassword	Password that Java Performance Monitor must use to connect to RealServer.

MonitorConnections

Maximum number of Performance Monitor sessions that can connect to RealServer. The maximum number of Java Performance Monitor connections does not reduce the allowed number of media connections. The Java Performance Monitor connects to RealServer over a TCP/IP connection. The connections should be restricted to the number of system administrators you anticipate will monitor this server.

Default value	4
Range of values	Whole number greater than or equal to zero

Syntax

```
MonitorConnections <count>
```

Example

```
MonitorConnections 6
```

MonitorPassword

Password that allows the Performance Monitor to connect to RealServer.

Note: If you accepted the default password (letmein) during Setup, you should change this setting immediately so that the RealServer is secure.

Default value	(none)
Range of values	Alphanumeric string without spaces

Syntax

```
MonitorPassword <password>
```

Example

```
MonitorPassword SrvTest1
```

Networked Settings

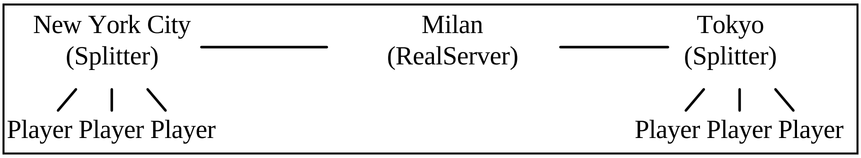
RealServer provides support for UDP-based splitting. Splitting allows servers to transmit live data sent from other RealServers. The servers receiving data from another server are referred to as splitters. Splitting can increase flexibility and efficiency in delivering content to users.

Splitting Streams between RealServers

With RealServer, the server providing live content to be split (sent to other servers) can establish a connection port for splitters to request the live streams. Splitters contact the server on this port and establish a UDP data stream.

Configuring Splitting

In addition to offering recorded media files or live broadcasts sent directly from RealAudio and RealVideo Encoders, RealServer can now offer live media sent from another RealServer. This gives you greater efficiency and flexibility in delivering live broadcasts to users.



For example, a concert from Milan can be broadcast over the Internet to RealServers in New York City and Tokyo. Users in those cities connect to the RealServer closest to them, thereby getting better media quality and performance.

The ability to split media streams is an optional feature controlled by the license you purchase from RealNetworks. If you would like to add this capability to your network, contact your RealNetworks reseller or RealNetworks.

To split a media stream, you need:

- One RealServer supplying the media stream.
- Another RealServer licensed for splitting.

All streams from the live server are broadcast to the splitter. If two servers are sending to the same server and they have a stream with the same name, the first one will be the only one split.

The following configuration file settings are required for splitting:

- **SplitterSourceList**
- **SplitterAnnouncePort**
- **SplitterControlList**

Setting the **SplitterBufferDelay** configuration parameter on the splitter is important for preventing dropouts in the media stream. The recommended value is 20 seconds; a minimum of at least 10 seconds should usually be used.

Required Splitter Setting

Setting	Description
SplitterSourceList	The Live Servers that this splitter should contact and split all live streams from.

Required Source Server Settings

Setting	Description
SplitterAnnouncePort	Defines the server port that a splitter will contact to request the splitting of its live streams.
SplitterControlList	Defines the valid servers that can connect and obtain split streams from this server. Any servers that contact on the SplitterAnnouncePort and are not in this list are ignored.

The following settings should not be changed unless you need to tune or optimize the splitting system.

Splitter Tuning Settings

Setting	Description
SplitterBufferDelay [number of seconds]	Amount of data the Splitter will buffer before releasing the stream to players.
SplitterTimeout [number of seconds]	The number of seconds for the splitter to wait after a stream has been stopped or interrupted. Alter only to shut down streams more quickly or streams are stopping when they shouldn't.
SplitterSourceProbe Interval	How often the splitter requests details about live streams from a source server. This may need to be adjusted if Timeout values are changed.

Source Server Tuning Settings

Setting	Description
SplitterResendBuffer	Depth of the buffer used for UDP resends in the Splitter protocol. Increase if you are getting consistent loss on all players from a split stream.
SplitterSourceTimeout	Time before source Server stops sending data to a splitter when the splitter is not responding.
SplitterMaxResendPPS	Maximum packets per second that a splitter source will output in response to a negative acknowledgement from a splitter.

Controlling Splitter Access to a Server

You can specify the splitters that are allowed to access a RealServer. If you do not limit the splitters, any splitter can access your server.

The **SplitterControlList** configuration setting lists the addresses of splitters that are allowed to access your Server.

If you specify a **SplitterControlList** configuration setting, only splitters from the specified addresses can access your Server. If you do not specify a **SplitterControlList** value, any splitter is accepted.

SplitterSourceList

Specifies the Live Servers that this splitter should contact and split all live streams from.

To use this setting, you must purchase splitting as part of your RealServer.

Default value	none
Range of values	Any number of pairs of any valid IP address or valid domain name and any valid port

Syntax

```
SplitterSourceList [{host, port}]
```

Example

```
SplitterSourceList [{172.16.2.54,5780}
{norton.prognet.com, 5781}]
```

SplitterAnnouncePort

Defines the server port that a splitter will contact to request the splitting of it’s live streams.

To use this setting, you must purchase splitting as part of your RealServer.

Default value	0
Range of values	Valid UDP port

Syntax

```
SplitterAnnouncePort <port#>
```

Example

```
SplitterAnnouncePort 8090
```

SplitterControlList

List of splitter domain names that are allowed to access RealServer. To use this setting, you must purchase splitting as part of your RealServer license.

Default value	(none)

Range of values	Valid IP addresses or domain addresses of the splitter computers allowed to access RealServer. <netmask> specifies wildcards

Syntax

```
SplitterControlList [{<address>, <net mask>},...]
```

where:

The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards.

Example

```
SplitterControlList [{121.23.101.0, 255.255.255.0}]
```

This example accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted.

The net mask 255.255.255.255 accepts only the single IP address specified.

To allow any player to connect, do not include a **SplitterControlList** setting in your configuration file. To prevent any player from connecting, specify:

```
SplitterControlList [{0.0.0.0, 255.255.255.255}]
```

SplitterBufferDelay

Amount of media, in seconds, to store in the TCP buffer for splitting. Buffering helps reduce packets loses (dropouts) over a splitter connection.

The recommended value is 20 seconds; a minimum of at least 10 seconds should usually be used. This parameter is set on the RealServer acting as a splitter for another Server.

To use this setting, you must purchase splitting as part of your RealServer license.

Default value	0
Range of values	positive integers

Syntax

```
SplitterBufferDelay <seconds>
```

Example

```
SplitterBufferDelay 20
```

SplitterTimeout

Defines the number of seconds the splitter will wait after a stream has been stopped or interrupted before shutting down. Alter only to shut down streams quicker or if streams are stopping when they shouldn’t.

To use this setting, you must have purchased splitting as part of your RealServer license.

Default value	20
Range of values	0-32767

Syntax

```
SplitterTimeout <seconds>
```

Example

```
SplitterTimeout 40
```

SplitterSourceProbeInterval

Determines how often the splitter requests details about live streams from a server. This may require altering if the Timeout values are changed.

To use this setting, you must purchase splitting as part of your RealServer.

Default value	10
Range of values	0-32767 seconds

Syntax

```
SplitterSourceProbeInterval <seconds>
```

Example

```
SplitterSourceProbeInterval 20
```

SplitterResendBuffer

Defines the depth of the buffer used for UDP resends in the Splitter protocol. Increase if you are getting consistent loss on all players from a split stream.

To use this setting, you must have purchased splitting as part of your RealServer license.

Default value	20
Range of values	0-32767 seconds

Syntax

```
SplitterResendBuffer <seconds>
```

Example

```
SplitterResendBuffer 30
```

SplitterSourceTimeout

Defines length of time, in seconds, for the source Server to stop sending data to a splitter when the splitter is not responding.

To use this setting, you must have purchased splitting as part of your RealServer license.

Default value	40
Range of values	0-32767

Syntax

```
SplitterSourceTimeout <seconds>
```

Example

```
SplitterSourceTimeout 60
```

SplitterMaxResendPPS

Maximum packets per second (PPS) that a splitter source will output in response to a negative acknowledgement from a splitter.

Default value	500

Range of values	250 - 5000

Syntax

```
SplitterMaxResendPPS <pps>
```

Example

```
SplitterMaxResendPPS 500
```

Configuring for Multiple User Accounts

You can divide the stream capacity of your server among multiple accounts. For example, an Internet Service Provider (ISP) can buy a RealServer and allocate media streams to individuals, companies, or organizations who use their service to host Web sites on the Internet.

Allocation is an optional feature controlled by the license you purchase from RealNetworks.

You can create individual accounts and specify the number of streams allocated to each, or you can use a naming convention to allocate the same number of streams to a large number of accounts.

UserList entries cannot be added to or deleted from the Java Performance Monitor or System manager; the UserList entry only supports changes to existing entries. For example, you can change the maximum or minimum number of connections a particular account is authorized.

Creating Individual Accounts

To divide your media stream capacity between specified individual accounts, you need to specify the users, the location of the users’ files, and the minimum and maximum number of streams each user is guaranteed.

UserList

Default value	none
Range of values	0-32767

Syntax

```
UserList [ {<user>, <path>, <min>, <max>}, ...]
```

where:

`<user>`	The name of the user. The key that the URL passes to RealServer to allow selection of a particular account entry. Name does not have to be a user directory and can be a string up to 1024 characters. In the URL, name is preceded by a tilde (~). For example: `pnm://audio.real.com/~fred/test.ra`

Selects the account entry defined for the user fred and then plays the media file **`test.ra`** from the **`fred`** private RealAudio path directory.

`<path>` Relative path for the directory of user files. Creates a separate path for media files in each account, allowing the owner of the account to alter the files in his or her directory without having access to any other users' files.

`<min>` The minimum number of streams allocated to the account. These streams are no longer available to any general RealServer requests. If **`min`** is 0, no streams are reserved for that account.

`<max>` Maximum number of streams allocated to the account. This number can be from 0 to the total number of streams available on RealServer.

Example

```
UserList [{ElectroMotors, /usr/electro/ra, 2, 5},
{CityWeld, /usr/cityweld/ra, 1, 3}]
```

This example allocates eight streams between two businesses posting Web sites on your service.

If more than the available streams are allocated to individual account entries, RealServer logs an error and provides access to the minimum streams for those entries in **UserList** before the limit is exceeded. All account entries after the limit is exceeded are not allocated streams.

Warning: If you are using Authentication to control access to secured content, note that when using account hosting, the absolute paths specified in the **UserList** configuration setting must *not* reside in the directory specified by the **BasePath** setting; if any directory specified in **UserList** does reside in the **BasePath** directory, media files in that **UserList** directory will not be secure.

Creating Accounts Using a Naming Convention

If you need to create a large number of accounts, and allocate the same number of streams to each, you can use a naming convention instead of listing each account individually. This function is typically used by Internet Service Providers who make RealServer available to a large number of customers.

The ability to create accounts using a naming convention is an optional feature controlled by the license you purchase from RealNetworks. If you would like to add this capability to your Server, contact your RealNetworks reseller or RealNetworks.

You can define a naming convention for most accounts, and still create individual accounts with different numbers of streams.

You can use one or both of the following naming conventions to allocate large numbers of accounts.

Naming Convention One

All accounts using this naming convention have a URL with the following format:

```
pnm://server.com/~account/directory/file.ra
```

All URL requests that begin with the same value for **`account`** are counted against that account's stream allocation.

The files for this account must be located in the **/directory/** directory relative to the **`path`** specified in the following UserList entry.

The following special UserList entry specifies the number of streams allocated to each account that uses this naming convention:

```
{ ~*, <path>, <min>, <max> }
```

In either of these cases, on UNIX you can use the special tilde character (~) as the **`<path>`** entry and the server will use the account to look up the password entry for this user and use their home directory to locate the content.

Naming Convention Two

All accounts using this naming convention have a URL with the following format:

```
pnm://server.com/dir1/dir2/dir3/dir4/file.ra
```

All URL requests that begin with the same value for the specified number of directory levels are counted against that account's stream allocation. If the directory level is set to 3, then **`/dir1/dir2/dir3/`** becomes the unique account identifier.

The files for this account must be located in the /**dir1/dir2/dir3** directory relative to the **`path`** specified in the following UserList entry.

The following special UserList entry specifies the number of streams allocated to each account that uses this naming convention:

```
{ *n, <path>, <min>, <max> }
```

where **`n`** is the number of directory levels that make up the unique account.

Note: If you use the *n naming convention, you must use it for all files on that RealServer. URLs relative to the BasePath directory do not work.

Setting	Description
UserDir	Path to be appended to the path defined for the account entries defined in the UserList setting.
UserList	List of accounts that are allocated media streams for private use.

UserDir

Path to be appended to the path defined for the account entries defined in the **UserList** setting. To use this setting, you must have hosting as part of your RealServer license.

Default value	(none)
Range of values	Absolute path name

Syntax

```
UserDir <string>
```

If no **UserDir** is specified then RealServer looks for media files in the path specified in the **UserList** entry.

Example

```
UserDir content
```

In this example, RealServer looks for media files in the subdirectory **content** of the directory specified in the **UserList** entry.

UserList

List of accounts that are allocated media streams for private use. To use this setting, you must have hosting as part of your RealServer license. This setting is used to create RealServer Hosting.

Default value	(none)
Range of values	see example

Syntax

```
UserList [ {<Accnt>, <privateRApath>,
<minStreams>, <maxStreams>}, ...]
```

There can be as many entries in this list as required. For more information on hosting, see “Creating Individual Accounts” on page 97.

Two special values (**~*** and ***n**) for **`<Accnt>`** enable you to define accounts using naming conventions. You can have a large number of accounts without having to list them individually. For more information, see “Creating Accounts Using a Naming Convention” on page 98.

Example

```
UserList
[{~*, /usr/persacct/, 1, 2},
{ElectroMotors, /usr/electro/ra, 2, 5},
{CityWeld, /usr/cityweld/ra, 0, 3}]
```

Ad Insertion Settings

Ad Insertion is an optional feature controlled by the license you purchase from RealNetworks.

The following settings pertain to Ad Insertion:

Setting	Description
AdEnabled	Initializes the RealServer for advertising play. If Ad Insertion is enabled, a plug-in must be defined (see AdPlugin).
AdPlugin	Specifies which plug-in is used to drive advertising insertion.
AdDefaultCfg	Optional. Global default advertising insert method for the RealServer.
AdCfgList	Specifies the location(s) of content media files for which ad insertion is enabled and type of ad insertion for those files.
AdLogPath	The relative path and name of the log file in which RealServer chronicles advertising presentation activity (**pnad.log** or **pnad**).

When any of these settings are changed, you must restart RealServer in order for the change to take effect. Data specified in these parameters is loaded into the server's memory.

AdEnabled

Turns Ad Insertion on (True) or off (False). If **AdEnabled** is set to True, a plug-in must be defined (see **AdPlugin** setting).

Default value	True
Range of values	True, False

Syntax

```
AdEnabled <Boolean>
```

Examples

```
AdEnabled True
```

AdPlugin

AdPlugIn specifies the .dll to drive Advertising Insertion in the RealServer. If **AdEnabled** is set to True, an ad plug-in must be defined.

Default value	Windows: \inetpub\rvroot\adli.3250.dll UNIX: /rvroot/adlist.so.5.0
Range of values	any valid filename for operating system **Ad List Plug-In:** Ad selection based on a text file that you define (see ads.txt in Chapter 6, “Ad Insertion”). **Ad Server Plug-In:** Ad selection based on third-party ad selection software.

To use third party ad targeting software for selecting ads, check with the software company for the correct *filename*.

Syntax

```
AdPlugin <filename>
```

Examples

```
AdPlugin d:\inetpub\rvroot\adli3250.dll
AdPlugin /rvroot/adlist.so.5.0
```

AdDefaultCfg

Optional. If used, specifies the global default ad insertion type for the RealServer. If ad insertion is enabled this is the ad insertion type for all content media files unless otherwise specified. You can customize ad insertion for content files by directory using the **AdCfgList** command.

Default value	7
Range of values	**0** No ads played **1** Play lead ad only

	2 Play end ad only **3** Play both lead and end ads **4** Play in-line ads only (pre-defined content markers must be defined in a .rad file) **5** Play lead and in-line ads **6** Play end and in-line ads **7** Play lead, end and in-line ads

Syntax

```
AdDefaultCfg <value>
```

Example

```
AdDefaultCfg 2
```

AdCfgList

Specifies the location(s) of content media files for which ad insertion is enabled and the type of ad insertion for the files in that path.

This command overrides the global setting specified by the **AdDefaultCfg** command.

Default value	none
Range of values	pathname relative to **BasePath**

Syntax

```
AdCfgList [{<path1>, <adinsert type>}, {<path2>,
<adinsert type>}, ...]
```

where:

`<path>` is the path to the directory containing the media clips and is relative to the **BasePath** setting

`<adinsert type>` is type of ad insertion permitted for files in the path**:**

Default value	7

Range of values	**0** No ads played **1** Play lead ad only **2** Play end ad only **3** Play both lead and end ads **4** Play in-line ads only (pre-defined content markers must be defined in a .rad file) **5** Play lead and in-line ads **6** Play end and in-line ads **7** Play lead, end and in-line ads

Examples

```
AdCfgList [{noads/video, 0}, {ads/video/20k, 4}]
AdCfgList [{film/previews, 3}, {news, 0}]
```

AdLogPath

Specifies the relative path and name of the log file in which RealServer chronicles advertising presentation activity.

Default value	logs/pnad.log
Range of values	valid path and file name

Syntax

```
AdLogPath <path/filename.log>
```

Example

```
AdLogPath /logs/pnad.log
```

For details about log file maintenance, see Chapter 9, “RealServer Log Files.”

Authentication Settings

Authentication is an optional feature controlled by the license you purchase from RealNetworks.

The following settings pertain to Authentication settings:

Setting	Description
AuthMode	Can be 0 (Player-based Authentication), 1 (User-based Authentication). Default is 0 (Player-based Authentication).
AuthPath	The path to the secure content directory.
AuthRegPrefix	User-defined keyword that is sent to CGI. Default is “register”.
AuthDBPlugin	Name of included text file, database, or user-defined data storage. Default is **rn-ppv-basic**.
AuthDBName	Name of mSQL or ODBC database, or path to supplied text file data source.
AuthDBUserID	Name RealServer uses to connect to the (optional) commercial database.
AuthDBPassword	Password RealServer uses to connect to the (optional) commercial database.
AuthAllowDuplicateIDs	Establishes whether multiple users all using the same user ID and password can retrieve secured content.
Realm	Keyword used to encrypt users’ passwords in the authentication database when configured for User-based authentication. Default is your customer name. **Warning**: Changing this variable means that every password in the authentication database must be reencrypted with the new variable and re-entered to work with the server.

AuthMode

Identifies type of authentication (Player-based or User-based) used by RealServer. **`0`** is Player-based and **`1`** is User-based.

Default value	1
Range of values	0,1

Syntax

```
AuthMode <number>
```

Example

```
AuthMode 1
```

AuthPath

The path to the secure content directory, relative to **BasePath**. The server must be able to identify content to which the server administrator wants to authorize access. The path and directory specified here contain that content.

Default value	/nonexistent
Range of values	Valid path and directory

Syntax

```
AuthPath <path>
```

Example

```
AuthPath secure\video
```

AuthRegPrefix

User-defined keyword that is sent to CGI. This variable is used for Player-based authentication only. In Player-based authentication, the Player registers by sending a special .rm clip to the server. This .rm clip's dynamically generated **`pnm://`** address also carries in it the username to be associated with the RealPlayer. The position of the username in the **`pnm://`** address is directly after a register prefix, which can be any word – but must be set here and in the **register.html** template.

Default value	register
Range of values	any word of multiple characters

Syntax

```
AuthRegPrefix <word>
```

Example

```
AuthRegPrefix register
```

AuthDBPlugin

The path to the plug-in text file or DLL for the database specified by **AuthDBName**.

Default value	plugins
Range of values	(short name of database file)

Syntax

```
AuthDBPlugin <name>
```

Example

```
AuthDBPlugin rn-ppv-basic
```

where:

Short name of database file	Actual name and platform
rn-ppv-basic	**ppvbasic.so.5.0** (text file, UNIX)

rn-ppv-basic	**ppvb3250.dll** (text file, Windows)
rn-ppv-msql	**ppvmsql.so.5.0** (mSQL database, UNIX)
rn-ppv-odbc	**ppvo3250.dll** (ODBC database, Windows)

AuthDBName

Name of mSQL or ODBC database, or path to supplied text file data source. If no value is specified, Authentication is considered disabled.

Default value	rn-ppv-basic
Range of values	valid path or file name

Syntax

```
AuthDBName <data source name or location>
```

Example

If using **rn-ppv-basic**,

```
AuthDBName c:\real\server\plugins
```

If using any other plugin,

```
AuthDBName auth.db
```

AuthDBUserID

Name RealServer uses to connect to the (optional) commercial database. Many databases require a username and password to read or write data. The RealServer will present itself to the authentication database using the chosen name and password, so must be able to operate under the access restrictions imposed on that username.

Default value	(blank)
Range of values	any alphanumeric string

Syntax

```
AuthDBUserID <name>
```

Example

```
AuthDBUserID MyServerUserName
```

AuthDBPassword

Password RealServer uses to connect to the (optional) commercial database.

Default value	(blank)
Range of values	any alphanumeric string

Syntax

```
AuthDBPassword <password>
```

Example

```
AuthDBPassword MyPassword
```

AuthAllowDuplicateIDs

Establishes whether multiple users all using the same user ID and password can retrieve secured content.

When **AuthAllowDuplicateIDs** is set to “1,” under Player-based authentication, customers who log on using the same RealPlayer are allowed to access secure content. Under User-based authentication, customers logged on with the same user ID and password and allowed to access secure content.

When **AuthAllowDuplicateIDs** is set to “0, ” the second person to log on under an existing connection is allowed to connect and the first person is given a warning that a duplicate RealPlayer ID is in use.

Default value	0 (FALSE), 1 (TRUE)
Range of values	0, 1

Syntax

```
AuthAllowDuplicateIDs <number>
```

Example

```
AuthAllowDuplicateIDs 1
```

Realm

User-defined keyword that is used to encrypt users’ passwords in the authentication database.

Default value	<Customer Name>
Range of values	any alphanumeric string

In Used-based authentication, the authentication database or text file stores user names and passwords, as well as access information and permissions. As an added security measure, passwords are encrypted. The **Realm** variable is used to increase the level of encryption.

Note: Changing this variable means that every password in the authentication database must be reencrypted with the new variable and re-entered to work with the server. For information on the password tool, see Chapter 5, “Authentication.”

Syntax

```
Realm <word>
```

Example

```
Realm realfred
```

Live Broadcasting Settings

Bandwidth negotiation during live events is a feature from RealAudio 3.0 that is no longer supported in RealServer. If you are using RealAudio 3.0, connect one encoder for each encoding algorithm you want to support. Specify the same file name as the output from each encoder. RealServer recognizes the format of each stream and directs it to RealPlayers requesting that format.

Setting	Description
EncoderPassword	Password used by RealAudio Encoder, RealVideo Encoder, and the **rvslta** utility program to connect to RealServer.
EncoderControlList	List of users and passwords. Allows multiple people to encode and use the same file name for different files.
EncoderTimeout	The time in seconds that the Server will wait before disconnecting a RealAudio Encoder or RealVideo Encoder that is not sending data.
LiveFilePassword	Password used for archiving live broadcasts.
LiveFileSize	Size of file, in megabytes, used for creating archive files of live broadcasts.
LiveFileTarget	File or directory to use to create the archive files of live broadcasts.
LiveFileTime	Maximum length, in time, of a archive file of a live broadcast.
URL	URL that points to the live media stream to be recorded by **rafile**.
BandwidthEncoding	The default bandwidth for archive files of live broadcasts.
LiveFileBandwidthNegotiation	Specifies that the **rafile** program use bandwidth**-**negotiation style of naming for archive files of live broadcasts.
InputFile	The path of a file to convert to a live file.

Setting	Description
OutputFile	Name of the simulated live stream sent using the **rvslta** utility.
ServerHost	Name of a RealServer to receive the live file.
ServerPassword	Password that **rvslta** must use to connect to RealServer.
ServerPort	Number of port on the RealServer to receive the live file from **rvslta**.

EncoderPassword

Password used by RealEncoder, RealPublisher, and the **rvslta** utility program to connect to RealServer. Note that the default setting of no password allows any RealAudio Encoder or RealVideo Encoder to connect to the server.

Default value	(none)
Range of values	Alpha-numeric string without spaces

Syntax

```
EncoderPassword <password>
```

Example

```
EncoderPassword rmRecord1
```

EncoderControlList

Allows multiple people to encode live content and use the same file name for different files. The file is published to a virtual directory beginning with the user’s name.

Default value	(none)
Range of values	Alpha-numeric string without spaces

Syntax

```
EncoderControlList [ {user1, user1password},
{user2, user2password}, ...]
```

Example

```
EncoderControlList [ {smith, encd} ]
```

In this example, any files encoded by the user “smith” will appear in a virtual directory named “smith.” So if the user encodes a file and calls it “live.rm,” the path to file will be “pnm://my.server.com/smith/live.rm”.

EncoderTimeout

The time in seconds that the Server will wait before disconnecting a RealEncoder or RealPublisher that is not sending data. If the connection to the Encoder is lost, the Server must disconnect before the Encoder can reconnect. Setting EncoderTimeout to less than 10 seconds is not recommended.

Default value	30
Range of values	1 - 32767 seconds

Syntax

```
EncoderTimeout <seconds>
```

Example

```
EncoderTimeout 20
```

LiveFilePassword

Password used for archiving live broadcasts. Used by RealServer for automatic archiving and by the **rmfile** or **rafile** utility program.

Default value	(none)
Range of values	alphanumeric string without spaces

Syntax

```
LiveFilePassword <password>
```

Example

```
LiveFilePassword rmBroadcast1
```

LiveFileSize

Size of file, in megabytes, used for creating archive files of live broadcasts. Used by **pnserver**, **rmfile**, and **rafile**. By default, **rmfile** or **rafile** uses this setting unless overridden with the **-s** option.

Default value	0
Range of values	Integers greater than or equal to zero

Syntax

```
LiveFileSize <value>
```

Example

```
LiveFileSize 5
```

LiveFileTarget

File or directory to use to create the archive files of live broadcasts. Used by **pnserver**, **rmfile**, **rafile**. By default, **rmfile** or **rafile** uses this value unless overridden by a file or directory name on the command line.

If name is a directory name, **rmfile** or **rafile** uses the filename of the live broadcast to name files. If it is a filename, it creates files in the working directory used to start **rmfile** or **rafile** or RealServer. In either case, it appends numbers to the archive files, starting at 0.

Default value	(none)
Range of values	Valid file name

Syntax

```
LiveFileTarget <name>
```

Example

```
LiveFileTarget pnfm.rm
```

Makes **rmfile** create archive files named **pnfm1.rm**, **pnfm2.rm**, and so on.

Windows

```
LiveFileTarget c:\real\server\content
```

UNIX

```
LiveFileTarget /usr/evand/rmfiles
```

Makes **rmfile** create archive files in the directory **/usr/evand/rmfiles** and names files using the filename list in the URL setting.

LiveFileTime

Maximum length, in time, of an archive file of a live broadcast. Specify time as a number and letter, such as 1m for one minute, 1h for one hour, and 1d for one day.

Default value	0
Range of values	Integers greater than or equal to zero and letters d, h, and m.

By default, **rmfile** or **rafile** uses this setting unless overridden with the **-t** option.

Syntax

```
LiveFileTime <value>
```

Example

```
LiveFileTime 1h
```

URL

URL that points to the live media stream to be recorded by **rafile**. Used by **rmfile**, **rafile**. **Rmfile** and **rafile** use this setting unless overridden by a URL on the command line.

Default value	(none)
Range of values	valid URL

Syntax

```
URL <url>
```

Example

```
URL pnm://server:7070/live1.rm
```

BandwidthEncoding

Specifies the default bandwidth for archive files of live broadcasts. Required for **rafile**; not required for **rmfile**. By default, **rafile** uses this setting unless overridden with the **-e** option.

Default value	None
Range of values	14_4.18, dnet.20, dnet.25, 28_8.36, dnet.50, dnet.100

Syntax

```
BandwidthEncoding <value>
```

Example

```
BandwidthEncoding dnet.20
```

Note: **Rmfile** is the main utility and supports .rm files; .ra files were used in the RealAudio 3.0 Server.

LiveFileBandwidthNegotiation

Specifies that the **rafile** program use bandwidth-negotiation style of naming for archive files of live broadcasts. Used by **pnserver**, **rafile**; not required for **rmfile**. By default, **rafile** uses this setting unless overridden with the **-b** option.

Default value	False
Range of values	True, False

Syntax

```
LiveFileBandwidthNegotiation <value>
```

Example

```
LiveFileBandwidthNegotiation TRUE
```

Note: **Rmfile** is the main utility and supports .rm files; .ra files were used in the RealAudio 3.0 Server.

InputFile

The path of a file to convert to a live file. Used by **rvslta**. By default, rvslta uses this value unless overridden by a filename on the command line.

Default value	(none)
Range of values	Valid file name

Syntax

```
InputFile <filename>
```

Example

Windows

```
InputFile c:\real\server\content\show1.rm
```

UNIX

```
InputFile /usr/content/show1.rm
```

OutputFile

Name of the simulated live stream sent using the **rvslta** utility. **Rvslta** uses this setting unless overridden by a filename on the command line.

Default value	(none)
Range of values	Valid media file name

Syntax

```
OutputFile <filename>
```

Example

```
OutputFile broadcast.rm
```

ServerHost

Name of a RealServer to receive the live file. The **rvslta** utility uses this setting unless overridden by a host name on the command line.

Default value	(none)
Range of values	Valid DNS name

Syntax

```
ServerHost <host>
```

Example

```
ServerHost server1.real.com
```

ServerPassword

Password that **rvslta** must use to connect to RealServer. By default, **rvslta** uses this value unless overridden by a password on the command line.

Default value	(none)
Range of values	Alpha-numeric string without spaces

Syntax

```
ServerPassword <password>
```

Example

```
ServerPassword StreamKey1
```

ServerPort

Number of port on the RealServer to receive the live file from **rvslta**. Must be the port number of the RealServer specified by the **ServerHost** setting. **Rvslta** uses this setting unless overridden by a port number on the command line. For information on using **rvslta**, see Chapter 7, “Delivering Live Content.”

Default value	(none)
Range of values	Valid port number

Syntax

```
ServerPort <port>
```

Example

```
ServerPort 8081
```

Multicasting Settings

Setting	Description
MulticastAddressRange	The range of multicast addresses (host group addresses) to which RealServer can send a live multicast stream.
MulticastControlList	The range of client or network addresses (typically on an intranet) which RealServer allows to receive multicast delivery.
MulticastDeliveryOnly	Restricts the client addresses specified in **MulticastControlList** to multicast reception only—client addresses so restricted cannot connect to RealServer in unicast mode.
MulticastPort	The port number for multicast broadcasts.
MulticastTTL	The Time To Live (TTL) for multicast packets.

MulticastAddressRange

The range of multicast addresses (host group addresses) to which RealServer can send a live multicast stream. The network administrator will know which multicast addresses are available on the local intranet.

When starting a multicast delivery of a live broadcast, RealServer scans the address range specified in **MulticastAddressRange** for the first available address. RealServer sends the live broadcast to the first available multicast address that it finds within the range.

Be sure to include enough addresses in **MulticastAddressRange** to accommodate your broadcasting needs. If you are running only one RealServer process (see **MaxThreads** setting on page 68), you need one multicast address for each live multicast broadcast. For example, if you are broadcasting three live programs by multicast, you need at least three addresses in the **MulticastAddressRange.** If you are running more than one RealServer process, you should make available one multicast address for each broadcast times the number of forks used by RealServer. For example, if you are broadcasting three live programs via multicast, and RealServer is using three forks, you need at least nine multicast addresses.

Multicast addresses can range from 224.0.0.0 to 239.244.255.255. The addresses between 224.0.0.0 and 224.0.0.255 are reserved for routing protocols and other protocols. Other addresses and ranges are reserved for other applications. See RFC 1700, “Assigned Numbers,” for a complete list of restricted addresses. In general, if the multicast address does not start with the decimal numbers 244, it should be usable. **MulticastAddressRange** is required for multicast support.

Default value	(none)
Range of values	Valid IP addresses in the range 224.0.0.0 - 239.255.255.255

Syntax

```
MulticastAddressRange <address>-<address>
```

where:

`<address>` is an IP address configured for multicast delivery.

Example

```
MulticastAddressRange 230.125.141.0-230.125.141.255
```

MulticastControlList

The range of client or network addresses (typically on an intranet) which RealServer allows to receive multicast delivery. RealServer gives a client whose address is within the range access to multicast delivery only when the client requests such delivery.

This **MulticastControList** is required for multicast support.

Default value	(none)
Range of values	Valid IP addresses

Syntax

```
     MulticastControlList [{<address>, <net mask>}, ...]
```

where:

`<address>` is the domain address of the client computer or network for which RealServer uses multicast delivery if requested by the player.

`<net mask>` is a mask that specifies the bits in the domain address that are treated as wildcards. The bits in the IP address that correspond with the zeros in the net mask are treated as wildcards. For example, an address of 121.23.101.0 with a net mask of 255.255.255.0 accepts all IP addresses from 121.23.101.0 to 121.23.101.255. If the net mask is 255.255.255.128, all IP addresses from 121.23.101.0 to 121.23.101.127 are accepted.

If you want to grant multicast access to all addresses within a domain, you can specify a net mask of 0.0.0.0 if you have a RealServer licensed for the Internet. If you have an intranet-only RealServer, you cannot specify a net mask of 0.0.0.0.

To prevent all Players from accessing multicast delivery, do not include a **MulticastControlList** value in your configuration file.

The net mask 255.255.255.255 accepts only the single IP address specified in the address.

Example

```
     MulticastControlList [{204.71.154.0, 255.255.255.0}]
```

MulticastDeliveryOnly

When set to True, restricts the client addresses specified in **MulticastControlList** to multicast reception only—client addresses so restricted cannot connect to RealServer in unicast mode. This option can be used to help control bandwidth on a network (typically on an intranet).

MulticastDeliveryOnly does not affect client addresses that are not in the **MulticastControlList**—such addresses can connect to RealServer in unicast mode.

This **MulticastDeliveryOnly** setting is optional for multicast support.

Default value	False
Range of values	True, False

Syntax

```
MulticastDeliveryOnly <value>
```

where:

`<value>` is True to restrict client addresses in the **MulticastControlList** to multicast reception only, or False to allow any type of reception.

Example

```
MulticastDeliveryOnly True
```

MulticastPort

The port number for multicast broadcasts. The **MulticastPort** setting is optional for multicast support. Note that **MulticastPort** and **PNAPort** have different purposes. The multicast port is the port to which RealServer sends live multicast broadcasts. The PNA port is the port that a Player uses to establish a TCP/IP control connection with RealServer.

Default value	7070
Range of values	Any valid port number

Syntax

```
MulticastPort <value>
```

where:

`<value>` is the port number for multicast broadcasts.

Example

```
MulticastPort 7075
```

MulticastTTL

The Time To Live (TTL) for multicast packets. This value is used by routers in your network to determine whether a multicast packet is allowed to pass through the router. The **MulticastTTL** setting is optional for multicast support.

Default value	16
Range of values	0 - 255

Syntax

```
MulticastTTL <value>
```

where:

`<value>` is the TTL value included in multicast packet headers.

The following are the typical TTL values and their meanings:

TTL Value	Keep Packets Within
0 or 1	Local host
16	Site
63	Region
129-255	World

For most multicast uses, you should keep the multicast packets within your intranet by setting **MulticastTTL** to 16 or less.

See your network administrator for information on how your network is configured.

Example

```
MulticastTTL 16
```

Configuring Web Servers to Work with RealServer

RealServer works with any Web server that supports configurable MIME types. Setting the correct MIME type makes the user’s Web browser play the contents of a media file with a RealPlayer rather than download the contents of the file.

Your Web server needs to define the following MIME types:

audio/x-pn-realaudio (files with a .ra, .rm or .ram file extension)
audio/x-pn-realaudio-plugin (files with a .rpm file extension)

The procedure for associating media files with these MIME types varies from one Web server to another. If the Web server is on the RealServer machine, this is done by installers on Internet Explorer and Netscape on Windows NT, and on Apache, Netscape and NCSA on UNIX. The following procedures tell how to add MIME types to some common brands of Web servers. If you have questions, or if your Web server is not listed here, please consult your Web server documentation or the online documentation at the RealNetworks Web site:

www.real.com

CERN HTTPD (v.3.0) Server

1. Add the following lines to the **httpd.conf** file under the server’s root directory:
2.
```
AddType .ram audio/x-pn-realaudio              binary
AddType .rpm audio/x-pn-realaudio-plugin       binary
```
3.
```
AddType .ra audio/x-pn-realaudio               binary
AddType .rm audio/x-pn-realaudio               binary
```
4. Reinitialize the Web server.

EMWAC HTTPS (Windows NT Only)

1. In Control Panel, start the HTTP server applet.

2. Click **New Mapping**.
3. In the Extension edit box, type the filename extension:

   ```
   RAM
   ```

4. In the Mime Type edit box, type the full MIME type:

   ```
   audio/x-pn-realaudio
   ```

5. Click **OK**.
6. Repeat Steps 3 and 4, using the file extension:

   RPM and the MIME type:

   ```
   audio/x-pn-realaudio-plugin
   ```

7. Reinitialize the Web server.

Mac HTTP and HTTPD4Mac Servers

1. Enter the following information into your configuration file in the format appropriate for your server:

   ```
   Action: TEXT
   File Suffix: .ram
   File Type: *
   MIME Type: audio/x-pn-realaudio
   Creator: *
   ```

2. Repeat with the File Suffix:

   ```
   .rpm
   ```

   and the MIME Type:

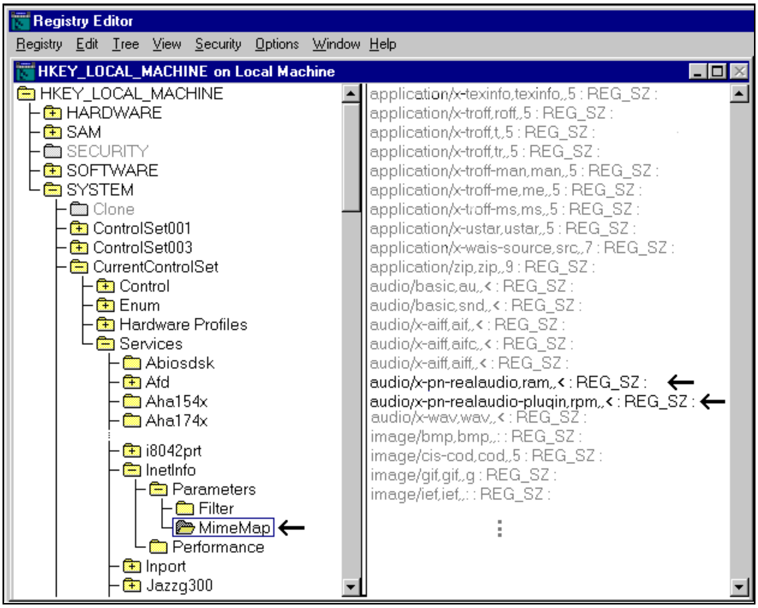
   ```
   audio/x-pn-realaudio-plugin
   ```

Microsoft Internet Information Server (Windows NT Only)

MIME type configuration is done in the Windows NT registry. To edit the registry:

1. Log on as Administrator.
2. Start **Regedt32**.
3. Click the entry:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\
   Services\InetInfo\Parameters\MimeMap
   ```



```
application/x-texinfo,texinfo,,5 : REG_SZ :
application/x-troff,roff,,5 : REG_SZ :
application/x-troff,t,,5 : REG_SZ :
application/x-troff,tr,,5 : REG_SZ :
application/x-troff-man,man,,5 : REG_SZ :
application/x-troff-me,me,,5 : REG_SZ :
application/x-troff-ms,ms,,5 : REG_SZ :
application/x-ustar,ustar,,5 : REG_SZ :
application/x-wais-source,src,,7 : REG_SZ :
application/zip,zip,,9 : REG_SZ :
audio/basic,au,,< : REG_SZ :
audio/basic,snd,,< : REG_SZ :
audio/x-aiff,aif,,< : REG_SZ :
audio/x-aiff,aifc,,< : REG_SZ :
audio/x-aiff,aiff,,< : REG_SZ :
audio/x-pn-realaudio,ram,,< : REG_SZ :
audio/x-pn-realaudio-plugin,rpm,,< : REG_SZ :
audio/x-wav,wav,,< : REG_SZ :
image/bmp,bmp,,: : REG_SZ :
image/cis-cod,cod,,5 : REG_SZ :
image/gif,gif,,g : REG_SZ :
image/ief,ief,,: : REG_SZ :
```

4. On the Edit menu select **Add Value**.
5. In the **Add Value** box, type:

   `audio/x-pn-realaudio-plugin,rpm,,<`
6. In the **Data Type** box select:

   `REG_SZ`
7. Click **OK**.
8. Leave the **String** box blank and click **OK**.
9. Repeat Steps 4 through 7. For Step 5, enter:

   `audio/x-pn-realaudio,ram,,<`

NCSA HTTPD (v. 1.3 and 1.4) Server

1. In the file **srm.conf** in the **SERVER_ROOT/conf** subdirectory, add the following lines:
2. AddType audio/x-pn-realaudio .ram
   AddType audio/x-pn-realaudio-plugin .rpm
3. AddType audio/x-pn-realaudio .ra
   AddType audio/x-pn-realaudio .rm
4. Reinitialize the Web server.

Netscape Netsite Server

1. Add the following to the **MIME.types** file:
2. type=audio/x-pn-realaudio exts=ram
   type=audio/x-pn-realaudio-plugin exts=rpm
3. Add the following line to the Server’s main configuration file (called **magnus.conf** in the examples given in the Netsite documentation):
4. Init fn=load-types mime-types=mime.types
5. Reinitialize the Web server.

O’Reilly Website NT Server

Use the admin tool on the mapping page to change the content type by entering the following commands:

```
.ram audio/x-pn-realaudio
.rpm audio/x-pn-realaudio-plugin
.ra audio/x-pn-realaudio
.rm audio/x-pn-realaudio
```

Webstar and Webstar PS

1. Start the Admin program for the Webstar server.
2. On the Configure menu, click **Suffix Mapping**.
3. Enter the MIME type information into its associated fields exactly as shown in the following example (these fields are case sensitive):

   ```
   Action: TEXT
   File Suffix: .ram
   File Type: *
   MIME Type: audio/x-pn-realaudio
   Creator: *
   ```

4. Click the **Add** button to update the MIME types directory.
5. Repeat Steps 3 and 4, using the File Suffix:

   ```
   .rpm
   ```

   and the MIME Type:

   ```
   audio/x-pn-realaudio-plugin
   ```

Spinner 1.0b12 - 1.0b15 / Roxen 1.0

1. Point your browser to the following URL:

   ```
   http://<server_name>:18830/Configurations/Gnats/
   Contenttypes/
   Extensions?40
   ```

   where:

   `<server_name>` is the name of computer running your Web server
   `<18830>` is the default administration server port; change this port number to your administration server port if necessary
2. Type the MIME types in the dialog box.

Apache 1.1.1

Apache comes preconfigured, but the MIME type for RealAudio and RealVideo files needs to be changed from audio/x-realaudio to audio/x-pn-realaudio. MIME types are normally stored in **/usr/local/etc/httpd/conf**.

Chapter 5 Authentication

Authentication lets you determine who can view specific content and for how long, thus providing a secured environment for your media clips.

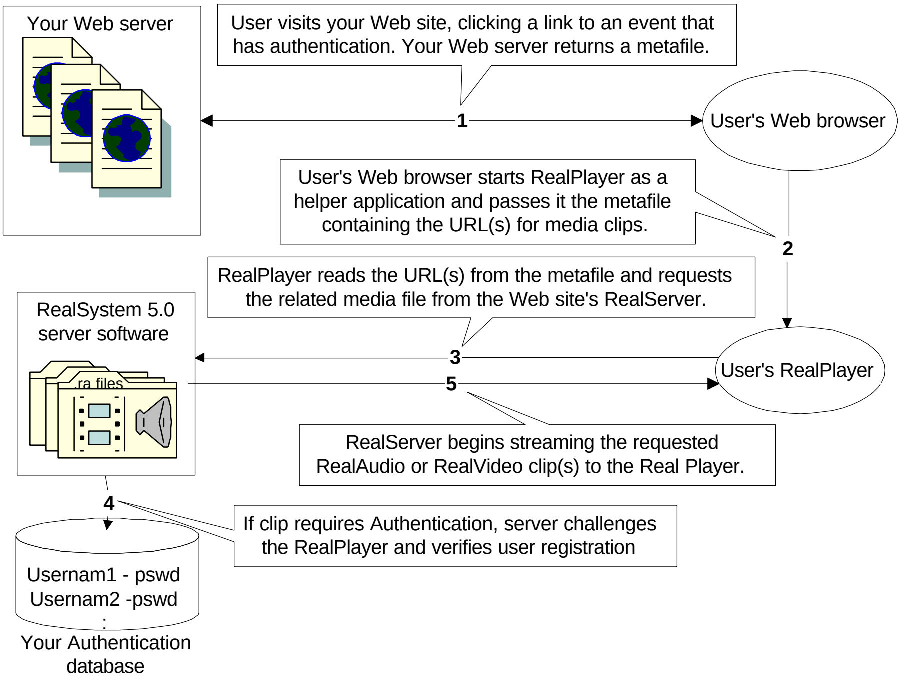
Potential applications include rigorous tracking of user demographics, pay-per-view events, music distribution, distance learning, premium content subscription, programmed music videos, karaoke, and restricted executive briefings.

There are two types of Authentication, Player-based and User-based, which differ in their degree of security and end-user interaction. Player-based authentication is based on a unique RealPlayer identification, contained within the RealPlayer, whereas User-based authentication involves exchange of an encrypted username and password. Player-based authentication involves less viewer interaction, but each RealPlayer must be registered individually by the viewer or central administrator. User-based authentication access privileges can be set by the central administrator prior to Player registration.

There are also three primary types of customer access to clips: unlimited event-based access, duration-based access, and calendar-based access. With event-based access, visitors can be given access to unlimited access to one or many media files. In duration-based access, visitors might receive five hours total viewing time of one or many media files or events. In calendar-based access, customers could have unlimited viewing of one or many media files or events until a certain calendar date.

In addition, Authentication can be further customized to best meet individual sites' needs. You can customize Authentication to create different types of payment and registration procedures.

The diagram below outlines the interaction between a visitor to a Web site and the RealServer Authentication process.


Your Web server
User visits your Web site, clicking a link to an event that has authentication. Your Web server returns a metafile.
1
User's Web browser
User's Web browser starts RealPlayer as a helper application and passes it the metafile containing the URL(s) for media clips.
2
RealPlayer reads the URL(s) from the metafile and requests the related media file from the Web site's RealServer.
RealSystem 5.0 server software
.ra files
3
5
User's RealPlayer
RealServer begins streaming the requested RealAudio or RealVideo clip(s) to the Real Player.
4
If clip requires Authentication, server challenges the RealPlayer and verifies user registration
Usernam1 - pswd
Usernam2 -pswd
:
Your Authentication database

Authentication is an optional feature of RealServer. Depending on which options you purchased, this feature may not be part of your installation. If you did not purchase this feature and would like to do so, please contact RealNetworks or your reseller.

Player-based and User-based Authentication

Authentication provides a way for the Web site administrator to control who has access to specific clips. That access can take the form of merely asking a visitor’s name once to requiring that the user enter a password and additional information every time he or she seeks access.

RealServer uses two types of authentication: Player-based authentication and User-based authentication.

Player-based authentication is most appropriate for situations where minimal user interaction is desired. The Authentication portion of RealServer is configured to allow or deny access to individual RealPlayers (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. Player authentication is best suited to applications like fan clubs, premium groups, microcommerce, intranet, and demographic tracking.

User-based authentication is most appropriate for situations where higher security is desired. RealServer allows or denies access to individual users, and authorization requires user interaction—a username and password challenge dialog box appears whenever the visitor attempts to enter a zone containing secure content, and the resulting exchange is encrypted. User-based authentication is tied to the person, rather than to the RealPlayer. With User-based authentication, it does not matter which computer a visitor uses to connect to the Web site. User-based authentication is best suited to applications like pay-per-view, executive briefing, and distance learning.

Player-based and user-based authentication can be run simultaneously on the same Web pages and applied to different clips, but this requires two RealServers, one for each type of authentication.

The following table describes how the two types of authentication appear to a Web site visitor:

	Player-based Authentication	**User-based Authentication**
Registration	1. Web page prompts user to create a user ID or password. 2. Visitor clicks **Submit**. 3. Server activates RealPlayer and returns a confirmation Web page. 4. Visitor confirms the submitted information.	1. Web page prompts user to create a user ID and password. 2. Visitor clicks **Submit** and is confirmed in a single step.
Action When Attempting To View Secure Content	None.	When user enters secure clip area, RealPlayer prompts user to enter user ID and password.
When Privileges Are Established by Administrator	System administrator sets visitor’s privileges concurrent with or after RealPlayer installation.	System administrator sets visitor’s privileges at any time. System administrator can independently distribute User IDs and passwords.
Clip Request	Visitor clicks link or directly types clip address into RealPlayer.	Same as Player-based Authentication.

Access Control

Access control features determine how long a user can view a particular clip or allots a certain length of time for viewing clips.

There are three types of authentication access:

- Event-based access
- Duration-based access
- Calendar-based access

A single RealServer can simultaneously deliver multiple types of access.

Event-based Access

In event-based access, the visitor registers in advance for unlimited access to one or more specific media clips. The steps involved are:

1. When a visitor clicks a link on a Web site, the site's Web server sends a metafile to the visitor's Web browser. Depending on which type of Authentication is in place, registration is automatic or requires a password.

2. The visitor's browser starts RealPlayer and passes it the metafile.

3. RealPlayer reads the **`pnm://`** address in the metafile and contacts RealServer for the appropriate clip.

   The visitor can bypass steps 1 and 2 by typing the **`pnm://`** address directly in RealPlayer. The **`pnm://`** address is used specifically by RealServers.

4. The RealServer challenges the RealPlayer. In Player-based authentication, this is transparent to the visitor. In User-based authentication, the visitor is prompted with a name and password dialog box.

5. Upon verifying the visitor's registration, RealServer plays the media clip or live broadcast. The visitor gets endless access to the clip.

Duration-based Access

In duration-based access, a visitor is granted the right to access a media clip or event for a specific length of time.

The registration process follows that of event-based registration, but permission is granted for a specific length of time and/or event (for example, five hours total viewing time applied to any or all of some number of specified videos).

If the time granted expires while the visitor plays a clip, RealSystem halts transmission of the clip, and the following error message appears:

> Server alert: Your account has expired, contact your content provider for more information.

Calendar-based Access

The process for expiration-based access follows that of event-based 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 RealPlayer is stopped, and an error message appears.

System Component Interaction

This section describes what happens when a visitor clicks a link on a sample Web site, which is based on the templates included with RealServer. These sample files configure the site for Player-based authentication, event-based access, and self-registration through the Web site. Files supplied with the RealServer installation are identified as such; all other files refer to files you create.

1. Visitor opens a page that contains links to content on your Web site. The page also contains a link related to registration. For example, this link might be labeled “Click here to register now.” If the page also contains links to secure content, the visitor will receive an error message if he or she attempts to access secure content before registering.

2. Visitor clicks the “Click here to register now” link, which requests the bandwidth-negotiated media file stored at the URL listed in **auth.ram**. If the visitor has a version of RealPlayer prior to 4.0, RealServer detects this and sends the visitor an audio warning and redirects the Web browser to the RealNetworks upgrade page. If the visitor has RealPlayer version 4.0 or later, RealServer redirects the Web browser to the register page.

3. Visitor registers by completing the form on the supplied **register.html** page hosted on the Web server, which includes a field for a unique user ID. This ID can be anything unique and memorable; email is a commonly used ID. In User-based authentication, this ID and password can also be assigned by a central administrator before the visitor connects to the site; in this case, steps 4 and 5 can be skipped.

4. Visitor clicks “Submit,” which calls **ppvodemo.cgi** (in Windows) or **ppvmdemo.cgi** (in UNIX).

5. The **ppvodemo.cgi** or **ppvmdemo.cgi** file creates a new record in the data records.

   In Player-based authentication, **ppvodemo.cgi** or **ppvmdemo.cgi** also triggers **register.rm**, a non-audio file containing an rmmerged event, which in turn calls the supplied **confirm.html** page. The **register.rm** file is essential for Player-based authentication because it carries the embedded PlayerID to the RealServer. In User-based authentication, **ppvodemo.cgi** or **ppvmdemo.cgi** calls the supplied **confirm.html** directly.

6. The visitor re-types his or her user ID into the Confirmation page and clicks “**Confirm**” or “**Never Mind**,” calling **ppvodemo.cgi** or **ppvmdemo.cgi** again.

   - If the visitor clicks “**Never mind**,” **ppvodemo.cgi** or **ppvmdemo.cgi** removes the complete visitor record from the data storage, then posts an HTML page displaying the message “You have not been registered.”

   - If the visitor clicks “**Confirm,**” **ppvodemo.cgi** or **ppvmdemo.cgi** grants access based on user ID and displays the starting page, which contains links to secured clips.

   Subsequent visitor access to the secured clips is transparent in Player-based authentication. Visitors access the site with no visible system interaction. Player IDs are automatically checked against the data storage whenever visitors request a clip. By contrast, in User-based authentication, visitors must reenter their user ID and password whenever they enter a different secure zone.

Implementing Authentication

The Authentication Setup program automatically places the RealServer files in correct locations and adds configuration settings to the **server.cfg** file. Information below is provided for your information should you want to customize this feature.

There are seven steps required to manually implement Authentication:

1. Place your files in the correct locations.
2. Set up data storage.
3. Configure the Authentication settings.
4. Configure the CGI files.
5. Configure the HTML files.
6. Configure the .ram files.
7. Configure the .rm files.

These steps are described in the following sections.

Placing Files in the Correct Locations

Setup places RealServer files in correct locations; you must place your files in specific places. Here are key aspects to consider:

- The data storage must be on the same computer as the RealServer.
- Your Web server and C compiler (if you are customizing the CGI) must be running on the same operating system as the RealServer. If they aren’t on the same operating system, obtain the data storage plug-in and CGI files appropriate to the second operating system and recompile the CGI for that platform.
- All plug-ins must be stored in the **plugins** directory.

- The **secure** directory and the **auth.rm** directory must be located under the directory with the **BasePath** setting in the configuration file. The BasePath directory is the default location for media clips. The Secure directory is the default location for all clips and directories to which you want to restrict access.
- The **auth.rm** directory contains clips used to detect older versions of RealPlayer. If an older version of RealPlayer is detected, files in this directory display a message instructing the visitor to obtain a newer version RealPlayer. While **auth.rm** and the associated clips are not strictly necessary, it is strongly advised that they be used so that visitors with older RealPlayers receive upgrade instructions instead of an error message.

Setting Up Data Storage

To authenticate end users, the RealServer stores user IDs and passwords or playerIDs, and their associated access permission information. When a RealPlayer tries to access a clip, the RealServer looks up this information to see whether the RealPlayer or visitor is authorized to view the clip. The information can be stored in either a series of text files or in a database. See Appendix B, “Authentication Text File and Database Structure, ” for information on the structure of these storage files or database. Templates for common databases are installed during Setup.

Storing Information in Text Files

The text file method is the default server installation, as it allows greater insight into the access permission structure, but the text file method lacks the scaleability and flexibility of a full database application. It is recommended that the text file only be used for demographic tracking applications or for troubleshooting the system before linking the database to the RealServer.

The text files do not exist physically when the RealServer is first installed. They are created when you run RealServer the first time and the **AuthDBPlugin** configuration file setting is set to **rn-ppv-basic**. 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.

Note: 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.

See also "Configuring Authentication Settings" on page 140 of this chapter.

Storing Information in a Database

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.

RealServer knows to use the database (rather than the text file structure) when the **AuthDBPlugin** is set to **rn-ppv-msql** or **rn-ppv-odbc**, and **AuthDBName** is set to the name of the database.

To set up an ODBC-compliant database other than those for which template files have been included, first create the database source, then create the database .ddl file, and finally use the procedure below to set up your computer for ODBC compliance.

To set up your Windows computer for ODBC compliance:

Note: If you are running RealSystem on a UNIX platform, you may skip this step.

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 have now told 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:

   ```
   ./msql2d &
   ```

2. Create the database by typing the following:

   ```
   ./msql create <databasename>
   ```

   Note that whatever you type for **<databasename>** will need to match the **AuthDBName** as described in the next section.

3. Create the tables using the database text file by typing the following:

   ```
   .msql -h <localhost> <databasename>
   < <textfilename>
   ```

   Be sure to include the less-than sign (<).

Configuring Authentication Settings

Add the Authentication settings to **server.cfg**. For instructions on editing the file and which settings to add, see Chapter 2, “Installing RealServer.” The settings to add are:

- **AuthMode**
- **AuthPath**
- **AuthRegPrefix**
- **AuthDBName**
- **AuthDBPlugin**
- **AuthDBUserID**
- **AuthDBPassword**
- **AuthAllowDuplicateIDs**
- **Realm**

AuthDBPlugin and **AuthDBName** work in conjunction to tell RealServer whether Authentication is enabled, and if so, what data storage file(s) to use.

There are four valid combinations of the **AuthDBName** and **AuthDBPlugin** settings:

AuthDBPlugin value	AuthDBName value	Result
blank	blank	Authentication is disabled.
`rn-ppv-basic`	path to directory – for example, **`c:\real\server\plugins`**	Authentication is enabled, and the supplied text files are used as authentication data storage.
`rn-ppv-msql`	name of database – for example, **`auth.db`**	Authentication is enabled, and a supplied or custom database is used as authentication data storage.
`rn-ppv-odbc`	name of database—for example, “mydatabase” (as set up in ODBC control panel)	Authentication is enabled, and a supplied or custom database is used as authentication data storage.

Configuring CGI files

Default CGI files are included with the RealServer installation, as well as uncompiled templates for creating customized CGI files.

If the supplied CGI files meet your needs, place them in your Web server script folder.

If you want to customize the supplied CGI files, use the steps below.

To customize your interface:

1. Open **ppvodemo.cpp** (in Windows) or **ppvmdemo.cpp** (in UNIX) in your editor.

   Replace or rename parameters passed in by the .html form. See Appendix B, “Authentication API Calls,” for a list of available options.

2. Recompile the CGI file.

3. Install the CGI file in the Web server’s CGI directory. Make sure appropriate folder exists and file permissions are set.

Configuring HTML files

These files determine what the viewer sees when registering on your site.

To customize register.html:

1. Open the supplied **register.html** in your HTML or text editor.
2. Replace “your.webserver.here” with the name of your Web server.
3. Replace “path-to-cgi-bin” with the path to your Web server Script directory.
4. Replace “auth.cgi” with the name of the CGI script you created in “Configuring CGI files,” above.
5. Replace “path-to-the-plugin” with the full path to the plug-in you will be using.
6. Replace “auth.db” with the value specified by the **AuthDBName** configuration setting in **server. cfg**.
7. Replace the value for “Realm” with the value of the **Realm** setting in the configuration file.
8. Replace “0” with the value specified by the **AuthMode** configuration setting in **server.cfg**.
9. Make the appropriate changes for type of Authentication you’ll be using:

    - Player-based authentication:

        a. Replace “your.realserver.here” with the name of your RealServer.

        b. Replace “register” with the value specified by the **AuthRegPrefix** configuration file setting in **server.cfg**.

    - User-based authentication:

        a. Replace “pnm://your.realserver.here/register” with “http://your.webserver.here/confirm.html,” where “your.webserver.here” is the name of your Web server.

        b. Remove subsequent line.

10. The other lines in **register.html** may be safely modified by following the instructions at the top of the page, but these aren't required.
11. Move the newly modified **register.html** to your Web server root directory.

To customize confirm.html

1. Open the supplied **confirm.html**.
2. Replace “your.webserver.here” with the name of your Web server.
3. Replace “path-to-cgi-bin” with any path information required to get to the Web server Script directory.
4. Replace “auth.cgi” with the name of the CGI script you created in “Configuring CGI files,” above.
5. Replace “path-to-the-plugin” with the full path to the plug-in you will be using.
6. Replace “auth.db” with the value of the **AuthDBName** configuration setting in the **server.cfg** file.
7. Make the appropriate change for the type of Access you’ll be providing:

    - Event-based access:
        a. Replace “secure” with the path from the BasePath to the secure content directory (such as secure)
        b. Remove the lines referring to “debitttime” and “expirestime.”

    - Duration-based access:
        a. Replace “secure” with the path (from BasePath) to the secure content file (such as “secure/secure.rm”).
        b. On the line referring to “debittime,” replace “0” with “1”.
        c. Remove the line referring to “expirestime.”
        d. Replace “05/24/70” with the proper expiration date.

    - Calendar-based authentication:
        a. Replace “secure” with the path (from BasePath) to the secure content file (usually secure/secure.rm).
        b. On the line referring to “permissiontype,” replace “0” with “2”.
        c. On the line referring to “debittime,” replace “300” with the proper time value (in seconds).
        d. Remove the line referring to “expirestime”.
8. Replace “urltype” with “1” for a directory, “0” for an individual clip. It is recommended that directory authentication be used, as bandwidth negotiation will not function with clip-level authentication.
9. Replace “your.webserver.here” with the name of your Web server.

10. The other lines in **confirm.html** may be safely modified by following the instructions at the top of the page, but these aren't required.
11. Move the newly modified **register.html** to your Web server root directory.

To customize authlink.html

The **authlink.html** files works with its default settings. Place it in the same directory where you placed **register.html** and **confirm.html**.

Configuring .ram files

The .ram metafiles are the link between your Web pages and the media files stored on the RealSystem. The sample files contain placeholders for the paths to the **auth.rm** and **secure.rm** directories.

1. Open the supplied **auth.ram** in your text editor.
2. Replace "your.realserver.here" in the following line with the name of your RealServer:

   ```
   pnm://your.realserver.here/auth.rm
   ```

3. Open the supplied **secure.ram** in your text editor.
4. Replace "your.realserver.here" in the following line with the name of your RealServer:

   ```
   pnm://your.realserver.here/secure/secure.rm
   ```

5. Place the .ram files in the Web server directory that contains the **authlink.html** file you customized in "Configuring HTML files," above.

Configuring .rm files

If you are using Player-based authentication, you must configure **register.rm**.

In addition, in both Player-based and User-based authentication, the files in the auth.rm directory are used to detect earlier versions of RealPlayer that are installed on visitors' computers. These .rm files contain embedded events which redirect visitors’ Web browsers to the RealServer download page at **www.real.com**.

RealPlayer versions 3.0 and earlier do not work with Authentication and may display an error message. RealPlayer version 4.0 works with Player-based Authentication only. RealPlayer version 5.0 supports both Player-based Authentication and User-based Authentication.

Although the files related to **auth.rm** are not necessary for running a secured Web site, they are highly recommended. If these files are not installed and configured, visitors with older versions of RealPlayer will receive an error message instead of being sent to the RealNetworks Web page to upgrade their players.

If you choose not to use these files, modify the link on **authlink.html** that points to **auth.ram** so that it points directly to **register.html** instead of to **auth.ram**.

To create and configure the auth.rm files:

1. Replace the URL in the supplied **auth.txt** file with the URL of the **register.html** page.
2. Run **rmmerge** (command line: `rmmerge -f rmevents.dll auth.txt auth.rm`). For more information, see the *RealAudio and RealVideo Content Creation Guide*.
3. Make 6 copies of the **auth.rm** file.
4. Rename the copies and give them the following names:

   pnrv.144

   pnrv.18

   pnrv.1930

   pnrv.24

   pnrv.36

   pnrv.70
5. Place the newly created files in the **\Content\auth.rm** directory on the RealServer.

If you’re using Player-based Authentication, create the **register.rm** file, which will extract each RealPlayer ID.

To create and configure the register.rm file for Player-based Authentication

1. Replace the URL in the supplied **register.txt** with the URL of the **confirm.html** page.

2. Run **rmmerge** (command line: **`rmmerge -f rmevents.dll register.txt register.rm`**).
3. Move the newly created **register.rm** file to the content directory of the Web server.

Using the Password Tool

In User-based authentication, the 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 Password tool, then copied into the appropriate field in the authentication database or text file. For information on the appropriate field in the database or text file, see Appendix B, “Authentication Text File and Database Structure.”

The command for using the password tool is

```
ppvpass <realm> <username> <new password>
```

where **`<realm>`** is the same word used as the **Realm** configuration setting in the **server.cfg** file, **`<username>`** is the username exactly as it is entered or will be entered in the authentication database or text file, and **`<new password>`** is the unencrypted password.

The resulting encrypted password will be displayed on the screen.

RealServer encrypts passwords with the MD5 encryption algorithm. It uses the form **`MD5("User:ServerRealmFromConfig:Password")`**. On BSD systems and some other UNIX systems, you can generate these passwords with the command:

```
echo -n "User:Realm:Password" | md5
```

This encryption is described in the Digest Access Authentication RFC (RFC 2069).

Troubleshooting Authentication

If an aspect of Authentication is not working, use the steps below with the supplied files to ensure that Authentication is working correctly in its default configuration.

1. Make sure your database is running and that appropriate access permissions and paths have been set such that the RealServer can access the database.

2. Start RealServer. If it is already running, stop and restart it. The server must always be started after the database is started, and after any changes are made to the database structure.

3. Point your Web browser to the **secure.ram** link on the **authlink.html** page. The RealPlayer starts and the message “Server alert: access to secure content denied” appears. If the browser is able to immediately receive media content or receives a database error, your RealServer is unable to access your database. Check your permissions and restart the server.

4. Now point your browser to the “**register**” link. Your RealPlayer plays a 0-second clip (one of the **pnrv** files located in **auth.rm**) and your browser displays the **register.html** page. If there is an error playing the clip, check the path in the auth.ram file and make sure your RealServer is running and able to serve other clips from the content directory. If the clip plays, but doesn’t show the **register.html** page, make sure your merged the correct path into your clip from the register.txt file.

5. Fill in a username on the register page and click Submit. Under Player-based Authentication, your RealPlayer plays a 0-second clip (the **register.rm** file) and your browser displays the **confirm.html** page. Under User-based Authentication, your browser displays the **confirm.html**. If a CGI error occurs, check the path to the CGI in your html files and check the placement of the CGI on your Web server. If a RealPlayer “Error 14” occurs, make sure your RealServer is able to serve other clips, make sure the register.rm file is in the path identified in your .html, and make sure the RealServer was started after the database. If the clip plays but the confirm.html page does not appear, make sure you correctly merged the correct path from register.txt into register.rm.

6. Fill in a username in the confirm page and click Submit. You are returned to the **authlink.html** page, and the secure content is now accessible. If you aren't returned to the **authlink.html** page, check the address given in **confirm.html**. If you are returned but are still given a "Server alert" message, check the database to see whether you were properly registered. The **confirm.html** will grant access to a secure directory for Player-based authentication, and to a secure directory and clip for User-based authentication.

7. If you are able to register and unregister (filling in the same username in the **confirm.html** page and pressing "Never Mind") and are allowed and denied access to the clip in the **secure.rm** directory, you have successfully installed Authentication and can further customize your interface.

Chapter 6 Ad Insertion

RealServer allows you to easily insert ads into on-demand content and live content from a rotated list. This capability can also be extended to allow the easy configuration of a “playlist” of content clips that play sequentially without interruption.

The presentation of content and ads is like television or radio in that the transitions between content and ads are seamless.

Previous versions of RealPlayer paused to re-buffer whenever a new content clip was introduced into the broadcast stream. Users visiting your Web site experienced this re-buffering as a delay at the beginning of each clip.

With the release of RealServer 5.0, you may now:

- Insert advertisements or announcements at the beginning of any live media stream.
- Insert advertisements or announcements in the front, middle or at the completion of any on-demand media clip.
- Configure rotating ads with sequential or weighted distribution.
- Configure a playlist of content clips that play together sequentially without interruption.
- Apply media maps and clickthrough URLs to video ads to better direct targeted users.
- Integrate with third party ad servers, such as NetGravity, for advanced advertisement rotation and placement management and reporting.
- Capture information about the ad clips served, type of Ad Insertion, duration of ad play and more.

Ad Insertion is an optional feature of RealServer. Depending on which options you purchased, this feature may not be part of your installation. If you did not purchase this feature and would like to do so, please contact RealNetworks or your reseller.

Features

Ad Insertion features can be used in many ways.

Ad Insertion for On-Demand Content

RealServer can insert advertisements before, during, or at the end of an on-demand content clip. Ads played before content clips are called lead ads; ads played after content clips are called end ads; ads inserted into a clip mid-play are called in-line ads. You can specify when you want in-line ads to appear during content clip play. For example, you can insert ads at 10 seconds, 30 seconds and 90 seconds into the content clip. All ads can be used in any fashion: lead, end, or in-line.

Ad Insertion for Live Event Broadcasts

The RealServer can insert ads from a rotated list at the beginning of a live event broadcast. The ads will be played prior to the live content broadcast with a seamless transition between the ads and live content.

Ad List

The RealServer can cycle through a pre-configured list of ad clips. For example, the first time a user plays a content clip configured for lead ads, RealServer inserts the first ad on the ad list at the beginning of the content clip. The next time a user plays the same content clip, the second ad is played before the content clip. This is repeated until all the ads on the list have been played at the beginning of the content clip, when RealServer returns to the first ad on the list and repeats the process.

Content Rotation

A list of content clips can be defined for playback. When the user clicks a RealAudio or RealVideo link, a string of pre-configured content clips will be played sequentially with no re-buffering or pausing between clips.

Image Maps

Ad clips can contain Image Maps, just as content clips can contain Image Maps. For more information, see the *RealAudio and RealVideo Content Creation Guide*.

Third Party Database Application Integration

For more specific ad targeting beyond simple ad insertion and more control of content rotation, RealServer can interface with third party database applications such as ad servers. RealServer is equipped with an open Application Programming Interface (API) for integration with other such ad servers. Detailed information on the Ad Insertion API, as well as SDKs and toolkits, is available at **www.real.com**.

When a user accesses content that has Ad Insertion, the third party software uses information passed from the RealServer to determine which ad is to be inserted by the RealServer or which content clips to string together.

The RealServer passes information about the user to the ad server, which analyzes this information, selects the most appropriate ad clip, and returns the filename and URL for that ad clip to the RealSystem. The data passed from the RealServer to the third party application includes:

- RealPlayer PlayerID
- RealPlayer IP
- URL of the content that the RealPlayer is viewing or has selected to view
- Time and date information from the RealPlayer
- Stream Compatibility Information:

- Audio/Video Codec used by the featured content being played
- Bit rate(s) of content stream
- Video frame rate of content
- Video frame dimension

Ad Insertion Files

Ad Insertion uses the following types of files:

File	Description
pnad.log or **pnad**	Advertising Presentation Log. Log file that chronicles all insertion and rotation activity performed by RealServer, including information related to ad clips and RealPlayer. For more details about this file, see Chapter 9, “RealServer Log Files.”
ads.txt	Lists ad files to play for a given content clip directory. RealServer rotates through this list. This file is also used to list content clips for Content Rotation.
***Filename*.rm**	The RealAudio or RealVideo content clip into which an ad is to be inserted.
***Filename*.rad**	Ad cue file. Lists the in-line Ad Insertion schedule for the companion content clip (.rm file). *Filename* must match the file name of the companion content clip. This file contains one or more timestamps and is used by RealServer to determine at what point in the content clip to insert an ad. For Content Rotation, this file is used to list the Content clip schedule.

Note: The **.rad** file and **ads.txt** file must reside in the same directory as the content clip they reference, and the directory must be marked for Ad Insertion in the configuration settings file. The **AdCfgList** setting identifies which directories contain ads and what type of insertion (lead, end, or in-line) is to be used.

Ad Insertion Data Types

The RealServer can insert ads for RealAudio and for RealVideo data types.

Location of Advertising Media

Ad files that are to be inserted into the on-demand or live broadcast stream must reside on or be accessible by the RealServer that is broadcasting the featured content.

Bandwidth Negotiation

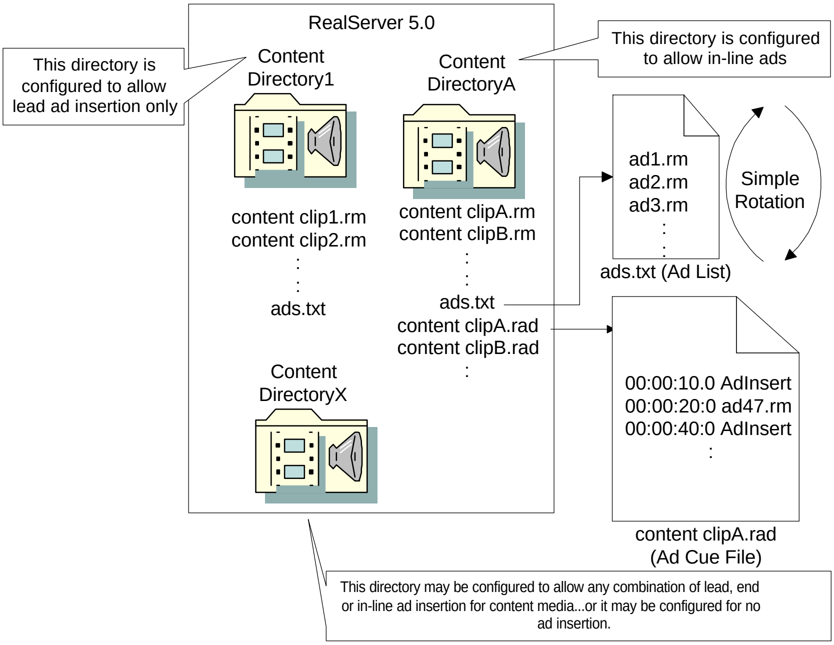
The Ad Insertion feature supports bandwidth negotiation. For information on bandwidth negotiation, see the *RealAudio and RealVideo Content Creation Guide*.

Ad Insertion Process Flow

There are two methods for inserting ads into a media stream. You can use an ad list to incorporate specific ads from a rotating list on RealServer, or you can send user information to third party ad software to select an ad for insertion. The process flows for each of these methods are discussed in the following sections.

Process Flow Using a RealServer Ad List

The following diagram portrays the scenario for Ad Insertion with on-demand (pre-recorded) content, using an ad list on the RealServer.



When a visitor clicks a RealAudio or RealVideo link on your Web site:

1. RealServer checks the Ad Insertion configuration settings in **server.cfg** for the directory in which that media file resides.

   - If the link points to a content clip in a directory that is configured o play a lead ad (Content Directory1 in this example), the first ad in the ad list file (**ads.txt**) of that directory is played and then the content clip is played.

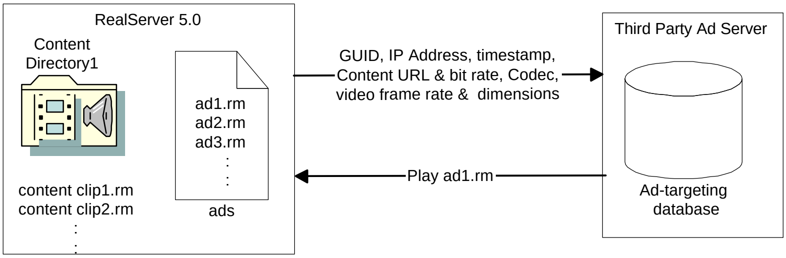
   - If the link points to a content clip in a directory that is configured for in-line ads (Content DirectoryA in this example),the server opens the .rad file for that content clip, and reads the timestamp(s) and ad(s) to insert. If the server encounters an **AdInsert** variable in the .rad file, it plays the next ad listed in **ads.txt**.

2. The content clip and ad clip stream plays with no break, no time delay, and no rebuffering between clips. All transitions between advertising clips and the content clip are seamless.
3. For RealVideo ads and content, users can click a video map on the screen to have their Web browsers display another URL. The media continues to stream to the user's RealPlayer.

A directory containing live content does not actually contain a live content file at all—a file name for the live content is specified by the RealEncoder or RealPublisher, and it points to the directory where **ads.txt** or the .rad file and the .rm ad files are located, but the file name is a virtual name and is not actually placed in the directory. The live content file does not actually exist, unless it is also archived as it is broadcast. However, the RealServer uses the **ads.txt** or .rad file located in that directory and plays the lead ads just as if a live content file were there.

Process Flow Using Third Party Software

The following diagram portrays the scenario for Ad Insertion using decision-making software on a third party ad server.



When the user clicks a RealAudio or RealVideo link on your Web site:

1. The RealServer sends detailed information about the user to the third party software.
2. The third party targeting software selects an appropriate ad based on the data and sends the name of that ad file back to the RealServer.
3. The RealPlayer begins to play either the ad or content clip based on the instruction received from the third party ad software.

In the example, **ad1.rm** will be displayed. As the content clip plays, RealServer checks with the third party database application to obtain the timestamp and filename of the next ad.

The content clip and ad clip play with no break, no time delay, and no rebuffering between clips. Transitions between advertising clips and the content clip are seamless.

4. For RealVideo ads and content, users can click a video map on the screen to have their Web browsers display another URL. The media continues to stream to the user’s RealPlayer.

RealPlayer Interaction with Ad Insertion

RealPlayer Seek Bar Performance

RealPlayer displays the time for the sum of all the content and ad lengths. They appear to the user as a single clip, making it difficult for the user to skip past advertisements.

Title, Author & Copyright

Whether a content clip or an ad clip is playing, RealPlayer shows the appropriate title, author, and copyright information for that clip.

Clickthrough URLs

Like video banner ads, you can apply media map coordinates (a clickthrough URL) to any streaming ad. If the user clicks anywhere in the RealPlayer display during an ad, the user’s Browser displays the underlying Web page without interruption to the RealPlayer.

For more information about clickthrough URLs, refer to the *RealAudio and RealVideo Content Creation Guide*.

Implementing Ad Insertion

The RealServer must be configured for advertising play before any ads can be played. When the RealServer is configured for Ad Insertion, it inserts audio and video advertising clips into current audio and video broadcast streams.

There are four steps to configuring RealServer for Ad Insertion:

1. Edit the RealServer configuration settings file.
2. Create ad lists for content directories.
3. If using in-line ads, create ad cue files for content directories.
4. Author content for ad insertion.

Detailed discussion and procedures for each of these steps are found in the following sections.

Editing the RealServer Configuration Settings File

Edit the following parameters in the RealServer **server.cfg** file to configure your RealServer for Ad Insertion. After modifying any of these parameters, restart the server. These settings inform RealServer which directories contain content clips and advertisements, and when the ads are to be played—lead, end, in-line, or a combination.

- **AdEnabled**
- **AdPlugin**
- **AdDefaultCfg** (optional)
- **AdCfgList**
- **AdLogPath**

For detailed procedures for setting or changing these parameters, see Chapter 4, “Configuring and Maintaining RealServer.”

Creating Ad Lists for Content Directories

In each content directory that you configured for Ad Insertion, create a text file called **ads.txt** which lists ad location and order. All content clips in that directory are subject to the ads defined in this file.

Content directories can have at most one **ads.txt** file.

The ad list is not used when the RealServer is integrated with a third party database application.

If the RealServer is running on a UNIX computer, and you make changes to an **ads.txt** file while users are connected to the RealServer, you can force the changes to appear to new users if you issue the SIGHUP command.

To create an ad list:

1. Using a text editor, create a text file named **ads.txt**.
2. In the file, type the paths to the RealAudio or RealVideo content file into which you want to insert ads. Type the path and file name of each ad file, relative to the **BasePath** setting in the server configuration file.

   To weight an ad, type its path and name on multiple lines. In the sample below, the **wa.rm** ad is weighted to run twice as often as the **mcat.rm** or **sport.rm** ads.
3. Save the file.
4. If you are using bandwidth negotiation, place the ads.txt file in the directory bearing the name of your ad content directory. Note that the entries listed in the **ads.txt** file are directory names that follow bandwidth negotiation format.

Sample ads.txt File

In the example below, the **game.rm** ad is inserted first, followed by the **wa.rm** ad. **Sport.rm** is played next, followed by **wa.rm** again.

```
ads\56kbs\game.rm
ads\56kbs\wa.rm
ads\56kbs\sport.rm
ads\56kbs\wa.rm
```

Creating Ad Cue Files for Content Directories

If you use the RealServer ad list for in-line ads, create a companion ad cue file for each content clip. The ad cue file contains timestamp(s) which instruct the RealServer when to insert an ad into the content clip.

The content file and the ad cue file must be in the same directory. That directory must be configured to have Ad Insertion enabled, with the ad insert type set for in-line. For details, see **AdCfgList** options 3, 5, 6 or 7 in Chapter 4, “Configuring and Maintaining RealServer.”

Ad Cue File (.rad file) Format

An ad cue file is a plain text file that has the same name as the content clip file (.rm), but with a .rad extension. Each record in the ad cue file specifies a timestamp and a pointer to an ad file. The pointer can be a relative path (relative to the directory specified in the **BasePath** setting) and file name for a specific ad, or an AdInsert event that prompts the RealServer to select the next ad from the **ads.txt** file.

Syntax

```
<timestamp > [<AdInsert> or <filename>]
```

where:

`<timestamp>` Point at which the ad will be inserted in the content clip (elapsed time).

Syntax: **`<dd>:<hh>:<mm>:<ss>.<t>`**

where ‘t’ is tenths of a second

`<AdInsert>` Instructs the RealServer to read the next ad listed in the **ads.txt** file of the current directory.

`<filename>` The path and file name, relative to the **BasePath** setting, of a specific ad clip to be inserted.

Example

```
00:00:03:00.0 logos\promo.rm
```

An ad cue file can list specific ads and it can also use the **AdInsert** variable. **AdInsert** points to the **ads.txt** file located in this directory or to the third party database application.

Ads are inserted just before a keyframe in the content clip. The timestamp you specify will be rounded up from 0 to 5 seconds, depending on where a particular keyframe exists in the content clip.

Sample Ad Cue (.rad) File

This ad cue file contains a pointer to a specific ad file, and uses the **AdInsert** pointer to direct the server to the **ads.txt** file located in this directory. If a third party database application is in use, **AdInsert** points to it.

```
00:00:00:30.0 logos\promo.rm
00:00:01:30.0 AdInsert
00:00:03:00.0 AdInsert
```

In this example, **promo.rm** is played 30 seconds into the content clip. At one minute and 30 seconds, the **AdInsert** variable directs RealServer to play the first ad listed in **ads.txt** file or the next file specified in the third party database application. Three minutes into content play, RealServer plays the second ad listed in **ads.txt** or third party software.

Authoring Content for Ad Insertion

Audio and video ads must be compatible with featured content files to achieve in-stream continuity. The following file properties must be identical for content files and the ad files that are to be inserted. It is recommended that the same RealEncoder version or RealPublisher version be used to create both ads and content, to ensure consistency in these settings:

- Audio Bit Rate (Kbps)
- Video Bit Rate (Kbps)
- Total Bit Rate (Kbps)
- Audio Codec
- Video Codec
- Video Frame Height

- Video Frame Width

If the ad file properties do not exactly match the content clip, the RealServer will not insert the ad file and will continue to play the content clip until the next ad insertion point as specified by **ads.txt** or .ram file. This error will be logged in the **pnad.log** or **pnad** file.

For more details about authoring content and ad clips, refer to the *RealAudio and RealVideo Content Creation Guide*.

Implementing Content Rotation

To create a playlist of content clips that play sequentially, an Ad Cue File (.rad file) is created with all timestamps set to 00:00:00:00.0.

Sample Ad Cue (.rad) File for Content Rotation

This ad cue file contains pointers to specific content clips:

```
00:00:00:00.0 clip\show1.rm
00:00:00:00.0 clip\show2.rm
00:00:00:00.0 clip\show3.rm
```

The visitor clicks on the content clip link with the associated .rad file and sees show1.rm first, followed by show2.rm and show3.rm and finally the content clip itself.

If the **ads.txt** file is used, the same example above can be configured with the following .rad file:

```
00:00:00:00.0 AdInsert
00:00:00:00.0 AdInsert
00:00:00:00.0 AdInsert
```

where the **ads.txt** file contains the following entries:

```
clip\show1.rm
clip\show2.rm
clip\show3.rm
```

Monitoring Ad Insertion Results

As each ad clip is inserted into a content file or content clip is used in a playlist, a log record is generated on in the advertising presentation log file (**pnad.log** or **pnad**) on your RealServer. This record includes, among other things, the playerID and IP address of the user, the content URL that was chosen for play, the ad file name, whether the ad was successfully inserted.

If an ad clip is incompatible with the featured content clip, the RealServer will not play the ad and a file incompatibility error is appended to the RealServer advertising presentation log.

If the RealServer fails to get a response from a third party plug-in, or if an ad list (**ads.txt** file) is not found, the featured content clip plays without interruption, and an error is written to the RealServer advertising presentation log.

In addition, the Access Log can track Player activity during clip play. Actions such as seek, pause, and image map click-through can be recorded when the **StatsMask** value is 4, 5, 6, or 7.

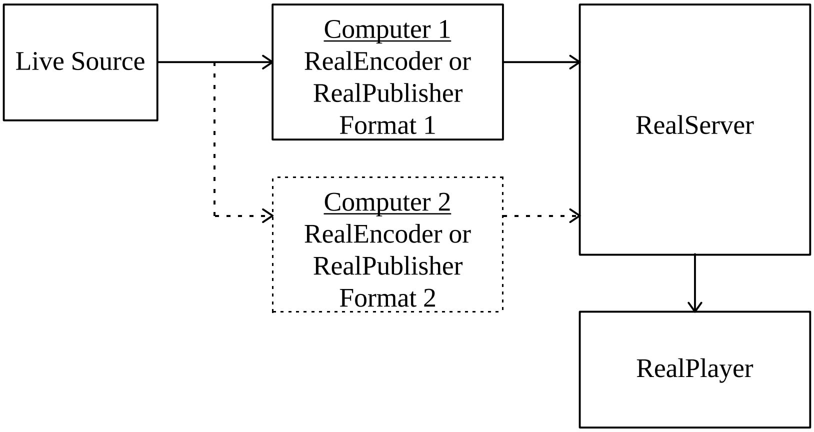
For more details about the Advertising Presentation Log and the Access Log, refer to Chapter 9, “RealServer Log Files.”

Chapter 7 Delivering Live Content

With RealServer and RealEncoder or RealPublisher, you can send live events such as performances, speeches and public events directly to users' computers. Live events can also be saved to disk for later transmission or for archive purposes.

RealServer can deliver live content in the same way it delivers pre-recorded or static content, or if your all components of your network are configured for multicasting, you can multicast the live content.

To provide content in multiple compression algorithms, you need to run a RealEncoder or RealPublisher and a separate computer for each codec, all with the same input signal. All concurrently running RealEncoders or RealPublishers should use identical settings except for compression type and stream name. Each computer sends its output to a RealServer.



Note: Bandwidth negotiation during live events is not supported by RealEncoder 5.0, RealPublisher 5.0 or RealServer 5.0.

Advertising Your Event

If you want help advertising your site or live event, you can advertise in Timecast: Your RealMedia Guide **(www.timecast.com**), the RealNetworks online resource for RealAudio and RealVideo sites and live events.

Inform the Timecast staff of your site or event by completing an online form. After you submit the online form, the data is verified by a staff member of Timecast and posted on the Timecast site. Timecast strives to be a quality listing service and therefore maintains editorial control over all information submitted. If you have any questions or need to make changes to any submissions please email **Timecast@Timecast.com**.

To advertise your site on Timecast:

Use the Web form located at **http://cgi2.timecast.com/cgi-bin/addsite.cgi** to submit your newly completed site for inclusion in either Timecast’s Audio, Video, or Live Stations Guide**.** On the Web form, you enter basic information about the site such as the site name, URL, and description.

To advertise your event on Timecast:

Use the Web form located at **http://cgi2.timecast.com/cgi-bin/addlive.cgi** to submit all your live events. On the Web form, you enter basic information about the event, such as name, complete URL, date and time of the event, and a short description. You can also indicate whether the event is recurring (occurs weekly) or non-recurring (one-time only).

Delivering Live Content

Make sure that your system requirements are met, that the configuration settings reflect your choices, and that you know how to use RealEncoder or RealPublisher. For information on using these programs, see the *RealAudio and RealVideo Content Creation Guide.* You will need:

- A live source
- RealEncoder or RealPublisher
- RealServer

The computer running RealEncoder or RealPublisher and the computer running RealServer can be on different platforms.

To deliver live content:

1. Attach your audio or video source to the audio or video capture card.
2. Configure the **server.cfg** file. For information on how to configure this file, see Chapter 4, “Configuring and Maintaining RealServer.”

   Be sure the following configuration settings are included in the RealServer configuration file:

   - **PnaPort -** the port number to which RealEncoder or RealPublisher connects.
   - **EncoderPassword -** the password the RealEncoder uses to connect. Passwords are necessary to keep unauthorized users from connecting to the stream of your live broadcast.
   - **EncoderTimeout -** configuration parameter specifies how long RealServer stays connected to a RealEncoder/RealPublisher that is not sending data.
   - **EncoderControlList -** Allows multiple people to encode live content and use the same file name for different files. The file is published to a virtual directory beginning with the user’s name..server.com/smith/live.rm”.
   - **LiveFileTarget** and **LiveFilePassword -** (Optional) Specify that the server saves the live content as a file.

3. Configure RealEncoder or RealPublisher for live broadcasting and click **Start** to begin encoding.

Archiving Live Broadcasts

You can choose to save (or “archive”) a live broadcast for playback later. The RealServer can be configured to automatically archive live broadcasts or you can use the **rafile** or **rmfile** utility program to archive broadcasts from any RealServer over a network. You can choose to create just one file, a new file based on elapsed time such as every 30 minutes, or a new file based on size such as every 5 MB. **Rafile** archives audio-only broadcasts; **rmfile** archives video-only or audio-and-video broadcasts.

If the RealServer or **rafile** or **rmfile** archives a live broadcast with the same destination path and file name as an existing file, RealServer renames the existing 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 the file as “concert.rm.86400”. 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.

Rmfile can be driven either by command line options or by a configuration file. The command line options will always overrule the configuration file.

To configure RealServer to archive automatically:

If you specify the **LiveFileTarget** and **LiveFilePassword** settings in the server’s configuration file (see Chapter 4, “Configuring and Maintaining RealServer”), RealServer automatically archives any live media stream that arrives at the Server. These media streams are created as RealMedia (.rm) files. Be sure you have enough available disk space to store the files generated from a live broadcast. The archive files are stored in the directory specified by the **LiveFileTarget** setting, or in the working directory that was used to start RealServer if a target directory is not specified.

To configure RealServer to save manually:

The **rafile** and **rmfile** programs can run on a different computer than RealServer or RealEncoder/RealPublisher because they accept a network address for the media source. Archive files written by **rafile** or **rmfile** are stored in the directory specified, or in the working directory used to start the **rafile** or **rmfile** program if no directory is specified.

For a list of options for the **rmfile** program, at a command line, type:

```
rmfile /?
```

Detailed information on the **rafile** and **rmfile** programs and their settings is available in the *RealAudio and RealVideo Content Creation Guide.*

Example 1:

A television station broadcasts over the Internet and wants to archive the entire broadcast day in multiple files each 60 minutes long to the **/usr/Archive** directory. The broadcast is named Live.rm and is available in RealVideo 28.8 format only. The relevant configuration settings are:

```
LiveFilePassword   rmBroadcast1
LiveFileTarget     /usr/Archive
LiveFileTime       1h
```

While the RealEncoder or RealPublisher is still connected, RealServer automatically archives the live broadcast to a series of files named Live0.rm, Live1.rm, Live2.rm, and so on, in the /usr/Archive directory. Each file contains one hour of audio data encoded in the RealVideo 28.8 format. If it is restarted, and it encounters files with the same names, it will append a number based on a timestamp to the pre-existing file.

Example 2:

A concert promoter broadcasts a live concert over the Internet and wants to archive the entire concert on a separate computer. The archive computer runs the **rmfile** utility program from a command line.

The configuration file setting on the RealServer computer is:

```
LiveFilePassword rmBroadcastZ
```

The command on the archive computer is:

```
rmfile -b -p rmBroadcastZ pnm://my.server.com:7070/live.rm
c:\real\server\content\archive
```

The **rmfile** utility program connects to the server using the URL pnm://my.server.com/live.ra. The resulting files can be copied to a RealServer for later rebroadcast of the concert.

Simulating a Live Broadcast

At times, you might want to play a recorded media file as if it were being broadcast live. Perhaps you want to test your system before a live event or delay broadcast of a concert. The **rvslta** (Simulated Live Transfer Agent) utility lets you play a recorded media file as if it were live. Users connecting to the site will get the event “in progress.” It can be used as a test, to delay broadcast of a live event or to multicast pre-recorded content.

To use **rvslta**, you need to specify the password listed under the **EncoderPassword** configuration setting (in the RealServer configuration file) and the names of the input and output files.

Syntax

```
rvslta -i <inputfile.rm> -o <outputfile.rm> -s
<server> [-p <port>] [-w <password>]
[-L <iterations>]
```

where:

`<inputfile>` is the path and file name to the input file.
`<outputfile>` is the path and file name to the output file.
`<server>` is the server name.
`<port>` is the server port (port defaults to 7070).
`<password>` specifies the password **rvslta** uses to connect to the server. It matches the password specified by the **EncoderPassword** in the configuration settings file. For information on setting this password, see Chapter 4, “Configuring and Maintaining RealServer.”
`<iterations>` is an integer that specifies how many times to loop the content.

Example

```
rvslta -i c:\archive.rm -o livenow.rm -s
www.my.server.com -p 7070 -w rmRecord1
```

Enhancing Live Broadcasts with Video Image Maps and Synchronized Web Pages

You can deliver a Synchronized Multimedia presentation or apply an image map to a video stream with a live broadcast. The events are delivered relative to the time a client begins playing the live broadcast, instead of relative to an absolute position in the broadcast.

The URL events must be defined before the broadcast, but you can change the content of the Web page the URL references during the broadcast.

To create a Synchronized Multimedia presentation for a live broadcast:

1. Create the input file that defines each event and the time that the event is sent. Remember that the times you specify are relative to when the client begins playing the live broadcast.

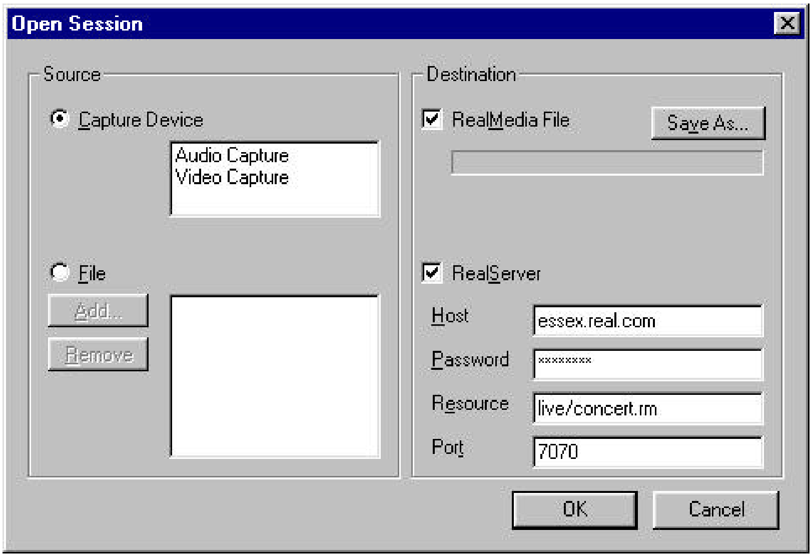
2. Run the **rmmerge** tool to create an .rm file.

3. Repeat steps 1 and 2 for any image maps.

4. Name the .rm file with the same filename you specify in the **Resource** box in RealPublisher for the broadcast.

5. Move the .rm file to the path specified in the **Resource** box in RealPublisher.

For example, if you set up RealPublisher as follows, the file must be named **concert.rm** and it must be located in the **/live** directory relative to the server's base path.



● Capture Device

Audio Capture
Video Capture

○ File

☑ RealMedia File

☑ RealServer

Host: essex.real.com

Password: ********

Resource: live/concert.rm

Port: 7070

Multicasting Live Content

Multicasting is a way of sending a live broadcast as a single data stream to only those clients who request the data. Multicasting contrasts with two other modes of transmission:

- Unicasting, which sends a separate, point-to-point data stream to each client that requests it.
- Broadcasting, which sends a single data stream to all clients on a subnet, regardless of whether a client has requested the data. (Note that broadcasting in this sense is not the same as broadcasting a live event from RealServer. Broadcasting a live event means transmitting the event live—it does not indicate the mode of transmission.)

Multicasting consumes much less bandwidth than either unicasting or broadcasting. Multicasting sends only a single data stream (rather than many copies of the data stream) to only those clients who request the data (rather than all clients on a subnet).

Multicast delivery over the Internet is made possible to a limited degree by the Internet Multicast Backbone (Mbone), which is a virtual network consisting of those portions of the Internet that are multicast-enabled. For the most part, however, multicasting is used over intranets.

Multicast Transmission

A server transmitting a live media event using multicast sends UDP multicast datagrams to a single IP address. This multicast address represents a particular transmit session rather than a specific client’s address. A multicast address is also called a host group address since, in effect, it also represents a group of clients (hosts) that have requested to receive the broadcast. Multicast addresses can range from 224.0.0.0 to 239.255.255.255.

Multicast datagrams are sent using the same IP operations as UDP unicast datagrams. In essence, a multicast broadcast is a unicast broadcast sent to a multicast address.

Each multicast packet contains a time-to-live (TTL) field in its IP header. The TTL field limits the scope of the broadcast by limiting the number of times that the packet can be forwarded by a router. The TTL setting is often used to limit the scope of a broadcast to a particular intranet.

RealServer Transmission to RealPlayer Reception

Multicast reception is much more complicated than multicast transmission. The following description focuses on how RealPlayer and RealServer interact to establish a multicast connection:

1. When a user requests to receive RealAudio or RealVideo media, RealPlayer opens a TCP/IP connection to RealServer. RealPlayer indicates that it is multicast-enabled.

2. If the media is a live broadcast stream, RealServer sends back a message over the TCP/IP connection that notifies RealPlayer of the multicast address and the multicast port. If RealServer is not already sending multicast packets to the multicast address, it scans the address range specified in **MulticastAddressRange** setting (see Chapter 4, “Configuring and Maintaining RealServer”) for the first available address. RealServer sends the live broadcast to the first available multicast address that it finds within the range.

3. RealPlayer creates two UDP connections. On one connection, it sends an Internet Group Management Protocol (IGMP) query that indicates to its nearest router that it wants to join the multicast host group at the multicast address and port specified by RealServer.

4. One or more routers, which must be multicast-enabled, then route the multicast packets to the subnet on which RealPlayer is located.

5. The computer on which RealPlayer is running starts filtering the network for packets sent to the multicast address and passes these packets to RealPlayer.

RealPlayer Reception

By default, RealPlayer is configured for various fall-back connection options if a multicast connection cannot be established with RealServer (see “Configuring RealPlayer for Multicast Reception” on page 175):

1. If, after sending out an IGMP query, RealPlayer does not receive any multicast packets after a specified number of seconds, it breaks the TCP/IP connection with RealServer.
2. RealPlayer then re-establishes a new TCP/IP connection with RealServer. This time, however, RealPlayer indicates that it will not receive multicast broadcasts and expects to receive UDP unicast packets.
3. If RealPlayer does not receive a UDP unicast packet after a specified number of seconds, it again breaks the TCP/IP connection with RealServer and re-establishes a new TCP/IP connection with RealServer. This time, RealPlayer indicates that it wants to receive the media over the TCP/IP connection, which is a more secure (in terms of packet loss) but slower connection.

Multicast Requirements

To support multicasting, both the sending and receiving nodes, as well as the routers between the sending and receiving nodes, must be multicast-enabled. Details of these requirements are beyond the scope of this chapter. This chapter describes only what is required to enable RealServer and RealPlayer for multicast broadcasting and multicast reception, respectively.

Multicast delivery of RealAudio or RealVideo files requires:

- Multicast-enabled client computers running RealAudio Player 3.0 or RealPlayer. The TCP/IP protocol stack on the client computer must support multicast reception. The network adapter and driver on the client computer must be able to filter for data link layer addresses mapped from network-layer multicast addresses.

- A correctly configured RealServer running on a computer that is correctly configured for multicast support. The TCP/IP protocol stack on the computer must support multicast transmission.

- Any routers on the client network must be multicast enabled.

Combining Splitting and Multicasting

To reach large audiences across the Internet, use splitters to send data across the Internet, and then use multicast delivery within each target intranet.

How to Multicast

Information on the configuration settings mentioned below can be found in Chapter 3, "Configuring and Maintaining RealServer."

To Enable Multicast Delivery:

1. Verify with your network administrator that the routers in your network are multicast-enabled and that the computer running RealServer is correctly configured for multicast support.

2. Specify the range of multicast destination addresses (host group addresses) available to RealServer, using the **MulticastAddressRange** configuration setting. The network administrator should know which multicast addresses are available on the intranet. Be sure to include enough addresses in the range to accommodate your multicast broadcasting needs.

3. Specify the addresses of client computers or networks that are allowed to connect to multicast broadcasts from RealServer. Use the **MulticastControlList** configuration setting to do this. Clients whose addresses are not included in the **MulticastControlList** can connect to RealServer only in broadcast mode.

4. Specify how far multicast packets can travel in your network. Use the **MulticastTTL** configuration setting to do this. The default value of 16 keeps multicast packets within a typical internal network at a site.

To Specify Optional Multicast Settings:

5. If you want to limit clients and networks specified in **MulticastControlList** to multicast connections only, set the **MulticastDeliveryOnly** configuration setting to `True`. This is an optional setting that can help reduce bandwidth usage on an intranet. A client whose address is not in the **MulticastControlList** can still make unicast connections to RealServer.

6. If you want to specify a TCP/IP port for multicasting that is different than the default port (7070), set the **MulticastPort** configuration setting. You should not need to change this setting unless you are having problems.

Configuring RealPlayer for Multicast Reception

For a client to receive a multicast broadcast, it must request and be granted membership in the host group to which the multicast broadcast is being sent. By default, RealPlayer is enabled to request multicast transmission of live broadcasts.

To check whether a RealPlayer is enabled for multicasts:

1. On the **View** menu, click **Preferences**.
2. Click **Transport**.
3. Click **Use specified transports**.
4. Click **Specify transports**.
5. Verify that **Attempt to use Multicast for live content** is selected.
6. Click **OK**.

Chapter 8 Configuring Your Web Site

After encoding your RealMedia files, you are ready to attach the files to Web pages. The following sections explain the construction and use of RealMedia content on your Web site. When you have your RealServer set up, use this information to showcase audio and video content from your site.

Making the Most of Your Content

To get the most out of your RealMedia content, educate your Web site's visitors about RealAudio and RealVideo. Let people know that they can listen and view your clips instantly, without download delays. Identify each clip with a Real bubble icon to distinguish it as real-time audio and video.

The graphics are available from the RealNetworks Web site at

www.real.com

Make it easy for your visitors to get RealPlayer by linking the graphics to the RealNetworks home page at:

www.real.com/

Metafiles

RealMedia material is reached via links, just as HTML documents use hyperlinks to connect Web pages. However, the RealMedia links you put into your HTML pages are not direct references to the RealAudio or RealVideo files. Instead they are references to text files which contain information needed to establish a connection between your RealServer and the user's RealPlayer and to initiate playback. These files are called metafiles.

Metafiles contain the URL of the file (or files) you want associated with the hyperlink. The visitor's browser passes the URLs to RealPlayer, which retrieves **.ra** or **.rm** files from your RealServer.

Metafiles contain the addresses of RealAudio (.ra), RealVideo (.rm) and RealFlash (.swf) files. These addresses are in the form of URLs. The URLs used to establish a direct connection between a RealPlayer and your RealServer begin with "pnm://" (RealNetworks Metafile). Once in place, this direct connection is used to stream to the Player and to carry commands (such as seek or pause) back to your RealServer.

Note: With certain browsers, when a Web page points to a local .ram file that in turn points to a local clip, the Player displays the following error message: "The requested URL is not valid." Some versions of some browsers require that ram files include an absolute path when pointing to local files.

Creating Metafiles

Your Web pages will link to metafiles, which in turn give the DNS names or IP addresses of your RealAudio, RealVideo, and RealFlash files.

If you are using RealFlash Animation, you'll include that information in the metafiles.

To create a metafile:

1. Use a text editor (such as Notepad) to create a file containing the RealAudio or RealVideo URL(s). The contents of your file should be in the following form:

```
pnm://hostname/path/filename
```

where **hostname** is the name of your Web server, **path** is the path to the file (relative to the BasePath of the RealServer), and **filename** is the name of the file ending with .rm, .ra, or +.swf.

For example, to provide access to a RealVideo file called **hello.rm**, the text of your metafile would be:

```
pnm://www.server1.com/hello.rm
```

where **www.server1.com** is the DNS name of the machine running your RealServer, on which you store your RealAudio or RealVideo files.

If you want more than one file to play in sequence when the user clicks your link, create a metafile containing several URLs (on separate lines with no intervening blank lines). For example, if your metafile contains:

```
pnm://www.server1.com/hello.rm
pnm://www.server1.com/welcome.rm
pnm://www.server1.com/coolstuff.rm
```

the Player automatically plays your three files in sequence. A listener can use the Clip menu on the Player to move forward and backward between clips.

If you are using RealFlash Animation, combine the audio file name with the RealFlash file name using the plus (+) sign:

```
pnm://www.server1.com/hello.rm+hello.swf
pnm://www.server1.com/welcome.rm
pnm://www.server1.com/coolstuff.rm
```

2. Save your metafile in the “All files (*.*)” format, using a **.ram** file name extension.

   For example, you could save the three lines shown above in a file named welcome.ram on your Web server.

3. In your HTML document, reference the metafile in a hyperlink, followed by a reference to the RealAudio or RealVideo icon (so that the icon is displayed to the left of the file):

   ```
   <A HREF="http://www.real.com/welcome.ram"> <IMG
   SRC="http://www.real.com/pics/rvfile.gif"
   align=left border=0> Welcome!</A>
   ```

   where **welcome.ram** is the metafile.

File Name Extensions

Each metafile that you create must be given a file name extension. This extension tells your Web server what the metafile is, to ensure that the enclosed URL is handled properly.

There are two metafile types: **.ram** and **.rpm**. These different file name extensions are passed on by your Web server and tell the user's Web browser which application to launch to play the referenced file:

.ram file – Web browser launches RealPlayer
.rpm file – Web browser launches RealPlayer Plug-in (see below)

Note: You must configure your Web server to understand that the extension .ram refers to the MIME type **x-pn-realaudio** and that the extension .rpm refers to the MIME type **x-pn-realaudio-plugin**.

Customizing Calls to Video and Audio Content

Optional arguments may be added to metafiles to finesse what is seen and heard by users when they click your RealMedia link. You may alter the point in a clip at which play starts or ends, or the Title, Author, and Copyright information is displayed by RealPlayer.

Add the options to your metafile following the URL to which they apply. Options must be preceded by a ? (question mark) and separated from each other by an & (ampersand). The syntax is as follows:

```
pnm://www.real.com/test.rm?[opt1]&[opt2]
```

where opt is a name/value pair separated by an equal sign (=). For example,

```
title="mymovie"
```

Changing Start and Stop Times

To create a link that starts playing a clip from a point other than the beginning of the file, use the **start** command. Specify the time into the clip at which play should begin. For example:

```
pnm://www.real.com/test.rm?start="30"
```

would result in playback starting thirty seconds into the file.

The format for the start time is as follows:

```
start="dd:hh:mm:ss.ss"
```

Tenths of seconds are separated from seconds by a decimal point; the other units of time are separated by colons. The time is interpreted from right to left, and it is not necessary to specify days, hours, or minutes if these are not relevant.

Similar to the **start** option is the **end** option. For example, the metafile text:

```
pnm://www.real.com/test.rm?end="5:30"
```

is used to provide for playback of test.rm that terminates five minutes and thirty seconds from the start of the clip.

Note: The end time is always measured from the actual start of the data in the file, even in the case where playback begins elsewhere. For example, the line:

```
pnm://www.real.com/test.rm?start="30"&end="5:30"
```

is used to start play of test.rm from the thirty-second mark and to stop play five minutes later.

Changing Title, Author, or Copyright Information

The following options exist to change the descriptive information from within the metafile:

```
title="new title"
author="new author"
copyright="new copyright"
```

Strings can be changed independently or in combination.

Changing information in this manner does not change what is stored in your file—only what is displayed when it is accessed through this particular metafile. This functionality is especially useful if, for example, you have one large file that contains your band's entire CD, and you want to credit the author of each song as it plays. You could create a multi-clip .ram file as follows:

```
pnm://www.server/band.rm?end="5:30"&title="Song1"
pnm://www.server/band.rm?start="5:31"&end="7:45"
&title="song2"&author="Joe Smith"
pnm://www.server/band.rm?start="7:46"&end="15:01"
&title="song3"&author="Jane Smith"&copyright="My
Music, 1996"
```

HTTP Streaming

HTTP streaming enables content providers to stream RealMedia clips from a Web server. While this method is not as robust as using metafiles, it provides a reasonable method for providing short RealMedia content to a limited number of users.

Before you can stream RealAudio and RealVideo clips through HTTP, you must define the following MIME types for 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)

video/x-pn-RealVideo (files with a .ra, .rm or .ram file extension)

video/x-pn-RealVideo-plugin (files with a .rpm file extension)

Some Web servers are pre-configured with these MIME types.

Note: If you are running a Web page off an ISP server, ask the ISP administrator to configure the server for the RealPlayer MIME types.

To stream RealAudio and RealVideo content using HTTP:

1. Copy your encoded RealAudio and RealVideo files (files with the .ra or .rm extension) to your Web server.
2. Use a text editor (such as Notepad) to create a metafile containing the RealAudio or RealVideo URL. For example, the contents of your file should be in the following form:

   ```
   http://hostname/path/filename
   ```

where **`hostname`** is the name of your Web server. For example: **`www.real.com`**

3. Save your metafile as “All Files (*.*) using a .ram file name extension.
4. In your HTML document, reference the metafile in a hyperlink. For example:

```
<A HREF="file.ram">
<A HREF="http://hostname/file.ram">
```

5. You can use relative or complete paths. If you use complete paths, you must include both the hostname and the complete path. For example:

```
<A HREF="http://www.real.com/home/welcome.ram">
```

6. When a user clicks on the link, the audio or video file(s) begin to download. RealPlayer begins playing after a few seconds; it does not need to wait for the entire file to be downloaded.

Custom Controls for RealAudio and RealVideo

RealAudio and RealVideo enables seamless integration of RealPlayer Controls into your Web page layout. You can place individual interactive components, such as a play button or image window, anywhere on your page, just as you would place an image using the <IMG> tag in HTML.

There are two products which, in conjunction with the most popular Web browsers, enable “in page” audio controls.

- RealPlayer Plug-in provides Player-like features to browsers that support the Netscape Navigator Plug-in architecture. This plug-in also works in Internet Explorer 3.0.
- RealPlayer Control for ActiveX works with Internet Explorer 3.0 and Visual Basic applications to provide RealAudio and RealVideo playback capabilities.

Using RealPlayer Plug-in

The Plug-in runs as an adjunct to Web browsers that support Netscape's Plug-in architecture. The RealPlayer Plug-in is included in the RealPlayer installation.

The **<EMBED>** tag specifies Plug-in attributes in HTML pages in much the same way that the **<IMG>** tag specifies image attributes. The basic **<EMBED>** tag for RealPlayer contains only the attributes **SRC**, **WIDTH**, and **HEIGHT**, as shown below:

```
<EMBED SRC=metafile.rpm WIDTH=width_value
HEIGHT=height_value>
```

For example:

```
<EMBED SRC="sample1.rpm" WIDTH=300 HEIGHT=134>
```

creates an in-page RealPlayer that is 300 pixels wide and 134 pixels high.

For the Plug-in, metafiles are stored with a .rpm file name extension.

The name tag is supported as an option with JavaScript to refer to a specific plug-in.

Note: Do not place the **<EMBED>** tag within a table.

Feature	Description
SRC Attribute	The SRC attribute specifies a metafile to be accessed. RealPlayer Plug-in is associated with a .rpm file name extension. This extension tells the user's Web browser to load RealPlayer Plug-in rather than the stand-alone RealPlayer. For the user's Web browser to correctly identify .rpm files, you or your system administrator must first configure the .rpm MIME type in your Web server. Users do not need to configure their Web browsers to recognize the .rpm MIME type. The plug-in architecture automatically sends .rpm files to RealPlayer Plug-in. Files with a .rpm extension are identical to .ram files, except for the extension.
WIDTH and	The WIDTH and HEIGHT attributes specify the size of

Feature	**Description**
HEIGHT Attributes	the embedded RealAudio or RealVideo component. Unlike images, Plug-ins do not size automatically. The WIDTH and HEIGHT can be specified in pixels (the default) or as a percentage of the Web browser window (for example: WIDTH=100%). **Note** If the WIDTH and HEIGHT attributes are not included, the Plug-in may appear as a tiny (and useless) icon with some browsers. If you want your Plug-in component to maintain an absolute size, specify HEIGHT and WIDTH in pixels. If you want the Plug-in graphic to scale with the Web browser window, specify size as a percentage. For example, if you want to fit the entire width of the Web browser window, use WIDTH=100%.
CONTROLS attribute	The CONTROLS attribute of the <EMBED> tag allows you to place individual control elements within your page. You can use multiple <EMBED> statements to construct a custom interface, made up of individual controls. You can also place multiple controls within a single <EMBED> statement. CONTROLS supports the following values: All, ControlPanel, InfoVolumePanel, InfoPanel, StatusPanel, StatusBar, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, StatusField, ImageWindow.
CONSOLE attribute	Sets a console name used to link multiple control instances. All controls with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same RealAudio or RealVideo clip. Call this function once for each instance of the Play or Stop button you want to link. The console name. “_master” links to all instances. “_unique” links to no other instances.
AUTOSTART attribute	Sets whether or not the control automatically starts playing once the source data is available. Valid values are TRUE or FALSE.

Feature	Description
NOLABELS attribute	Suppresses the Title, Author, and Copyright label text in the controls window. The text strings in the fields are still displayed.
RESET attribute	Resets RealPlayer Control for ActiveX playlist. Valid values are TRUE or FALSE.
AUTOGOTOURL attribute	Specifies how a URL is handled. Valid values are TRUE or FALSE. TRUE indicates that RealPlayer Control for ActiveX automatically forwards the URL event to the browser. FALSE indicates that the OnGotoURL VBScript event is used instead.

Creating HTML Pages for Browsers Unable to Use the Plug-in

Some Web browsers do not support plug-ins. You can create HTML pages that are enhanced for plug-ins but which also work for other browsers. Simply use the <**NOEMBED**> tag to include HTML statements for use by Web browsers that do not support Plug-ins.

The <**NOEMBED**> command should appear after an <**EMBED**> command and take the following syntax:

```
<NOEMBED> HTML to be ignored </NOEMBED>
```

For example, the command:

```
<EMBED SRC="sample1.rpm" WIDTH=300 HEIGHT=134>
<NOEMBED> <A HREF="sample1.ram"> Please play the
clip using the stand-alone RealPlayer.
</A></NOEMBED>
```

would show a page with the Plug-in if your page were accessed by a browser supporting Plug-ins, and would otherwise display the message “Please play the clip using the stand-alone RealPlayer.” (and allow playback with the standard RealPlayer).

Using RealPlayer Control for ActiveX

You can embed RealPlayer Control for ActiveX in HTML pages using the Object tag.

The following is an example of RealPlayer Control <OBJECT> in an HTML page.

```
<OBJECT
  ID=RAOCX
  CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
  HEIGHT=140
  WIDTH=312>
  <PARAM NAME="SRC" VALUE="pnm://audio.real.com/file.rm">
  <PARAM NAME="CONTROLS" VALUE="all">
</OBJECT>
```

Note: Directory names cannot have spaces.

Embedded Object Parameters for ActiveX

Feature	**Description**
OBJECT	Tag is used to embed RealPlayer Control for ActiveX. There should be one <**OBJECT**> tag per RealPlayer Control on the page.
CLASSID	Specifies the control's CLSID. This value is always “clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA” for RealPlayer Control for ActiveX.
HEIGHT	Specifies the control's height on the HTML page. A value of 0 makes the control invisible.
WIDTH	Specifies the control's width on the HTML page. A value of 0 makes the control invisible.
PARAM	Specially embedded tag for supplying parameters to the ActiveX object.

The following properties are available to the PARAM setting:

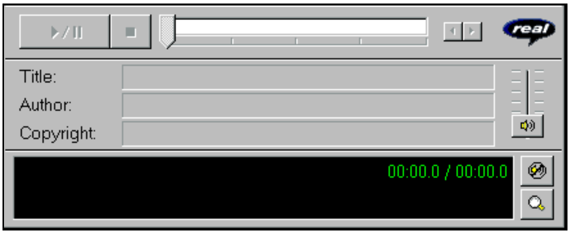
Property	**Description**
SRC	Sets the source of RealAudio or RealVideo clip. The SRC location can be pnm, file or http protocol. This parameter is required.
CONTROLS	Sets the visible components of the control. Valid CONTROLS include All, ControlPanel, InfoVolumePanel, InfoPanel, StatusPanel, StatusBar, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, StatusField, and ImageWindow.
CONSOLE	Sets a console name used to link multiple control instances. All controls with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same RealAudio or RealVideo clip. Call this function once for each instance of the Play or Stop button you want to link. The console name. "_master" links to all instances. "_unique" links to no other instances.
AUTOSTART	Sets whether or not the control automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
NOLABELS	Suppresses the Title, Author, and Copyright label text in the controls window. The text strings in the fields are still displayed.
RESET	Resets RealPlayer Control for ActiveX playlist. Valid values are TRUE or FALSE.
AUTOGOTOURL	Specifies how a URL is handled. Valid values are TRUE or FALSE. TRUE indicates that RealPlayer Control for ActiveX automatically forwards the URL event to the browser. FALSE indicates that the OnGotoURL VBScript event is used instead.

Specifying How the Control Should Look

The CONTROLS attribute allows you to place individual control elements within your page. The CONTROLS attributes for the Netscape Navigator Plug-in and RealPlayer ActiveX Control are the same. The following explains the output of each attribute:

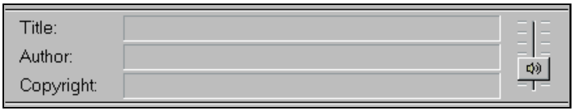
All - Displays a full Player view including the Control Panel, Information-and-Volume Panel and Status Bar.

Minimum Width: 21% **Maximum Width**: 100%
Minimum Height: 23% **Maximum Height**: 80%



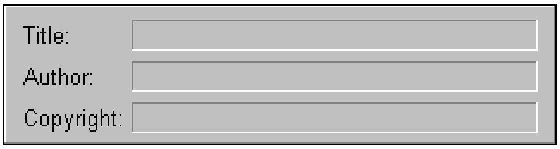
InfoVolumePanel - Displays the Title, Author, and Copyright information panel and the volume slider.

Minimum Width: 21% **Maximum Width**: 100%
Minimum Height: 12% **Maximum Height**: 50%



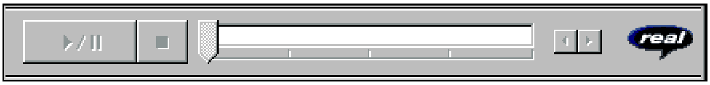
InfoPanel - Displays the Title, Author, and Copyright information.

Minimum Width: 20% **Maximum Width**: 100%
Minimum Height: 10% **Maximum Height**: 50%

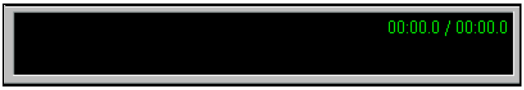


ControlPanel - Displays the play/pause button, the stop button and the position slider.

Minimum Width: 21%
Maximum Width: 100%
Minimum Height: 8%
Maximum Height: 25%



StatusPanel - Displays the Status Panel showing informational messages, current time position, and clip length. If you do not embed a Status Panel in your page, error messages are displayed in the Web browser's status bar.



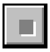
00:00.0 / 00:00.0

PlayButton - Displays the play/pause button.

Minimum Width: 5%
Maximum Width: 100%
Minimum Height: 2%
Maximum Height: 25%

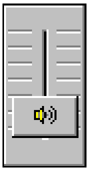
StopButton - Displays the stop button.

Minimum Width: 5%
Maximum Width: 100%
Minimum Height: 2%
Maximum Height: 25%



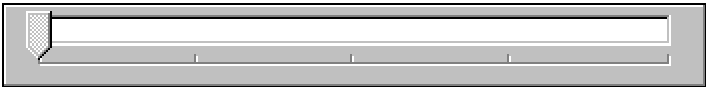
VolumeSlider - Displays the volume slider.

Minimum Width4% **Maximum Width**100%
Minimum Height10% **Maximum Height**100%



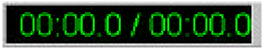
PositionSlider - Displays the position slider.

Minimum Width 8% **Maximum Width** 100%
Minimum Height 5% **Maximum Height** 25%



PositionField - Displays the field of the Status Bar showing position and clip length.

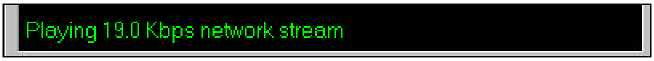
Minimum Width: 10% **Maximum Width**: 100%
Minimum Height: 4% **Maximum Height**: 25%



00:00.0 / 00:00.0

StatusField - Displays the message text area of the Status Bar.

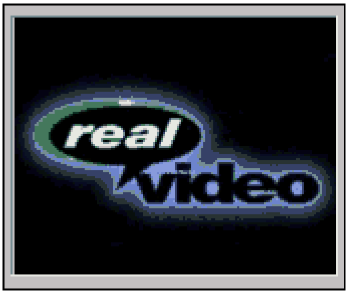
Minimum Width: 13% **Maximum Width**: 100%
Minimum Height: 4% **Maximum Height**: 15%



Playing 19.0 Kbps network stream

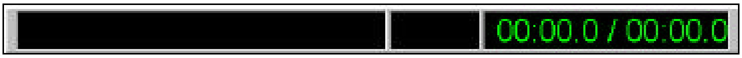
ImageWindow - Displays the video image. (Only available for .rm files)

Minimum Width: 100% **Maximum Width**: 100%
Minimum Height: 100% **Maximum Height**: 100%



StatusBar - Displays the status field, position field, channels (stereo/mono).

Minimum Width: 21% **Maximum Width**: 100%
Minimum Height: 5% **Maximum Height**: 14%



ActiveX Methods and Properties

Methods

Methods are functions that control the performance of the Control. Unless otherwise noted, these methods have no return values and no parameters.

Method	Description
DoPlayPause	Plays or pauses the current clip. Equivalent to clicking the Play/Pause button.
DoStop	Stops the clip. Equivalent to clicking the Stop button.
DoNextItem	Skips to the next clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealAudio or RealVideo files.
DoPrevItem	Skips to the previous clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealAudio or RealVideo files.
CanPlayPause	Tests if Play/Pause function is available. Returns TRUE or FALSE
CanStop	Tests if Stop function is available. Returns TRUE or FALSE.
HasNextItem	Tests if the next clip function is available. The next clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the last clip in the .ram file. Returns true or false.
HasPrevItem	Tests if the previous clip function is available. The previous clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the first clip in the .ram file. Returns true or false.
AboutBox	Opens the Copyright information for the control.

Method	Description
EditPreferences	Opens the Preferences dialog box. Enables the end user to set his or her preferences. These preferences are global to all RealAudio or RealVideo clients on the machine.
HideShowStatistics	Shows or hides the Connection Statistics dialog box.
IsStatisticsVisible	Tests if the Connection Statistics dialog box is displayed. Returns true or false.
DoGotoURL(url, target)	Causes the control to attempt a navigation to the specified URL in the specified frame target. The container must support URL browsing. Parameters: string URL, string target.

Object Properties

The object properties are set within Visual Basic and specify properties about the control.

Property	Description
Source	Specifies the URL of the clip to play. The Source location can be pnm:, file: or http: protocol.
Controls	Returns/Sets the visible components of the control. Valid CONTROLS include ALL, ControlPanel, InfoVolumePanel, InfoPanel, StatusBar, StatusPanel, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, ImageWindow and StatusField.
Console	Sets a console name used to link multiple control instances. All controls with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same clip. Call this function once for each instance of the Play or Stop button you want to link. The console name “_master” links to all instances while unique links to no other instances.

Property	Description
Autostart	Sets whether or not the control automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
NoLabels	Suppresses the Title, Author, and Copyright label text in the controls window. The text strings in the fields are still displayed.
AutoGotoURL	Specifies how a URL will be handled. Valid values are TRUE or FALSE. TRUE indicates that the RealPlayer ActiveX Control will automatically forward the URL event to the browser. FALSE indicates that the OnGotoURL VBScript event will be sent instead.

Java and JavaScript Methods

Methods

Methods are functions that control the performance of the Java enabled RealPlayer Plug-in.

Method	Description
SetSource(String *Source*)	Specifies the URL of the RealAudio clip to play. The Source location can be pnm:, file: or http: protocol.
SetControlsString (String *ControlsString*)	Returns/Sets the visible components of the control. Valid CONTROLS include ALL, ControlPanel, InfoVolumePanel, InfoPanel, StatusBar, PlayButton, StopButton, VolumeSlider, PositionSlider, PositionField, StatusField, StatusPanel, and ImageWindow.
SetConsoleName (String *ConsoleName* Boolean *True)*	Sets a console name used to link multiple RealVideo Plug-in instances. All Plug-ins with the same console name work together. For example, if you have multiple Play and Stop buttons on the same page, the console name would enable them to control the same

Method	**Description**
	clip. Call this function once for each instance of the Play or Stop button you want to link. The console name "_master" links to all instances while unique links to no other instances.
SetAutoStart(Boolean *bAutoStart*)	Sets whether or not the Plug-in automatically starts playing once the source data is available. Valid values are TRUE or FALSE.
SetNoLabels(Boolean *bNoLabels*)	Suppresses the Title, Author, and Copyright label text in the Plug-in window. The text strings in the fields are still displayed.
DoPlayPause()	Plays or pauses the current clip. Equivalent to clicking the Play/Pause button.
DoStop()	Stops the RealAudio clip. Equivalent to clicking the Stop button.
DoNextItem()	Skips to the next clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealAudio files.
DoPrevItem()	Skips to the previous clip in a .ram file that contains multiple clips. A .ram file is a metafile that points to one or more RealVideo files.
CanPlayPause()	Tests if Play/Pause function is available. Returns TRUE or FALSE
CanStop()	Tests if Stop function is available. Returns TRUE or FALSE.
HasNextItem()	Tests if the next clip function is available. The next clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the last clip in the .ram file. Returns true or false.
HasPrevItem()	Tests if the previous clip function is available. The previous clip function is available when the connected source is a .ram file that contains multiple clips, and the current clip is not the first clip in the .ram file.

Method	Description
	Returns true or false.
AboutBox()	Opens the Copyright information for the control.
EditPreferences()	Opens the Preferences dialog box. Enables the end user to set his or her preferences. These preferences are global to all clients on the machine.
HideShowStatistics()	Shows or hides the Connection Statistics dialog box.
IsStatisticsVisible()	Tests if the Connection Statistics dialog box is displayed. Returns true or false.

CallBack Methods

CallBack methods are functions that provide notification from the RealAudio or RealVideo engine. CallBack methods can only be used with Java applets by implementing the RAObserver interface.

For more information about these methods, visit the DevZone page at **www.real.com.**

Method	Description
onClipOpened(String *shortClipname*, String URL)	Sent when a clip has been opened.
onClipClosed()	Sent to indicate that no clip is currently open.
onShowStatus(String *status*)	Sent to indicate that the status text is changing.
setAutoGoToURL (boolean *bAutoGoToURL*)	Specifies how a URL will be handled. Valid values are true or false. True indicates that the RealPlayer Plug-in will automatically forward the URL event to the browser. False indicates that the OnGoToURL event will be handled by the Java applet instead.
onGoToURL(String *url*, String *target*)	Sent when a URL event has been encountered for the currently playing clip. This event will only occur

Method	Description
	if the AutoGotoURL property is false.

Advanced Control Attributes

The more exciting features of RealPlayer Plug-in and RealPlayer Control for ActiveX can be enabled by specifying custom attributes within your HTML documents.

Removing Text Labels from Controls

If your Web page is in a language other than English, or if you want to use the Description fields to display information other than Title, Author, and Copyright, you may remove the Title, Author, and Copyright Labels from the information area. Controls that display Title, Author, and Copyright information for a clip, support a NOLABELS=TRUE attribute.

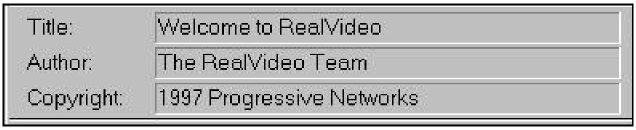
For example, allowing the default behavior

```
<EMBED SRC="use_lbl.rpm" WIDTH=350 HEIGHT=80
CONTROLS=InfoPanel>
```

or specifying NOLABELS=FALSE

```
<EMBED SRC="use_lbl.rpm" WIDTH=350 HEIGHT=80
CONTROLS=InfoPanel NOLABELS=FALSE>
```

results in the following display.

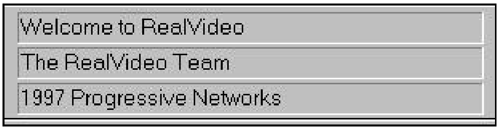


Title: Welcome to RealVideo
Author: The RealVideo Team
Copyright: 1997 Progressive Networks

On the other hand, using NOLABELS=TRUE, for example:

```
<EMBED SRC="no_lbl.rpm" WIDTH=350 HEIGHT=80
CONTROLS=InfoPanel NOLABELS=TRUE>
```

produces the following InfoPanel:



Welcome to RealVideo

The RealVideo Team

1997 Progressive Networks

Starting Clips Automatically

Adding an AUTOSTART=TRUE attribute tells the user's browser to automatically begin playing your clip when the page is visited. You can use this feature to begin narration, to play a welcome message or start a video.

Since only one clip can play at a time, if you specify AUTOSTART for more than one control instance, only the last control to load will play. The order in which your files are delivered is dependent on the Web server and on the browser's cache size. This is not necessarily the order in which you put them within your HTML. Therefore, you should specify AUTOSTART for only one control instance per page.

Playing Clips with a Hidden Control or Plug-in

If you want to play RealAudio or RealVideo clips without having a visible Plug-in control, you hide the control. By embedding a Plug-in in your page that has its size set to width=2 height=0, no image appears on your Web page. You can control the Plug-in with JavaScript.

For example:

```
<script Language=JavaScript>
function playSource()
{if (navigator.appName == "Netscape")
{document.javaPlug1.DoPlayPause();}
else
{RAOCX.DoPlayPause();}
}
</script>
<A HREF="#" onClick="playSource()"><IMG
SRC="button.gif"></A>
```

```
<OBJECT ID=RAOCX CLASSID="clsid:CFCDAA03-8BE4-
11cf-B84B-0020AFBBCCFA" WIDTH=2 HEIGHT=0>
<PARAM NAME="SRC"
VALUE="pnm://audio.real.com/welcome.rm">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
<embed src="start.rpm" Width=2 Height=0
Controls=PlayButton name=javaPlug1>
</OBJECT>
```

The above example works in both Netscape and Internet Explorer. Only one control will appear on the Web page.

Making Controls Work Together

You can embed any number of RealPlayer controls within a Web page. Normally, each tag instance refers to different content. Sometimes, however, you may want to link two or more controls together. For example, you can create a play button and an image window that work as a pair.

To include multiple components that work together, specify a CONSOLE attribute for each control and assign this the same value in each instance. For example:

```
<EMBED SRC="sample1.rpm" WIDTH=30 HEIGHT=33
CONTROLS="PlayButton" CONSOLE="Clip1">
<EMBED SRC="empty1.rpm" WIDTH=176 HEIGHT=144
CONTROLS="ImageWindow" CONSOLE="Clip1">
```

Note: Each **`<EMBED>`** tag must have a unique **`SRC`** attribute (if the same source is specified for two tags, one is ignored). Create a dummy .rpm file (named, for example: empty1.rpm) for a second control accessing a clip. Put a hard carriage return (ASCII code 13) into the dummy file.

Specifying a **`CONSOLE`** value of “_master” links a particular control to all other RealPlayer Controls on the page. Use this value, for example, to add a Status Bar to display information for all clips, for example:

```
<EMBED SRC="sample1.rpm" WIDTH=300 HEIGHT=33
CONTROLS="StatusBar" CONSOLE="_master">
```

Because many platforms, including Windows, only support one volume setting, all volume sliders act on the same underlying value and affect all clips regardless of CONSOLE name. For this reason, you may want to include only one volume slider per page, with no reference to a .rm file in its associated .rpm file.

If you only include an ImageWindow control on the Web page, the video stream may still be controlled by using the context sensitive pop-up menu. For Windows users, right-click the ImageWindow; Macintosh users, click and hold the mouse button until the pop-up menu appears.

Frame Document

A frame document describes the sub-HTML documents or frames that make up a window. The basic structure of a frame document is similar to that of a normal HTML document except that the FRAMESET tag replaces the BODY tag. Each frame is defined by the FRAME tag. In order to effectively use RealVideo, each FRAME tag needs the SRC and NAME attributes. The SRC points to the URL to be displayed in the frame. The NAME attribute assigns a name to the frame so that it can be targeted by links in other documents. The example below shows a simple frame document that would create two frames:

```
<HTML>
<FRAMESET ROWS="105,*">
    <FRAME SRC="banner.html" NAME="banner">
    <FRAME SRC="lyrics.html" NAME="Lyric">
</FRAMESET>
</HTML>
```

Synchronizing Frames and Video and Audio

Frames and RealVideo and RealAudio content are synchronized in the same way that a regular Web page and RealVideo and RealAudio content are synchronized: a .rae file. The difference lies in the addition of the targeted frame name to the text file that is used to create the .rae file. The syntax for each entry should follow the format:

```
u starttime endtime &&framename&&EventURL
```

where:

u stands for URL event (each line starts with the letter **u)**
starttime is the time into the clip when the new event is shown

`endtime` is the time into the clip when that event ends, **`&&`** is a delimiter
`framename` is the frame name as specified in your frame document
`EventURL` (generally beginning with "http:" or "file:") is the URL for that event (usually an HTML document).

The time format is:

```
[[[days:]hours:]minutes:]seconds[.tenths]
```

The lines of the input file must be in ascending order of start time. The end time should be at least one tenth of a second before the start time for the next event.

The example below shows an .rae input file for a Web site that displays lyrics as a song plays. Each verse is displayed within a frame called Lyric.

```
u 00:00:10.0 00:00:35.0
&&Lyric&&http://www.songs.com/ver1.html
u 00:00:35.0 00:00:50.0
&&Lyric&&http://www.songs.com/ver2.html
```

Note: You cannot use frames within a presentation using RealPlayer as a helper application. You can only use frames within a multimedia presentation using the Plug-in or ActiveX Control.

Synchronized Multimedia

In addition to basic audio and video content, RealAudio and RealVideo System allows you to create real-time on-demand multimedia presentations using the RMMerge tool (Windows or UNIX). These presentations can be as simple as a narrated slide show of your home page or as intricate as a multi-frame training program that the viewer controls.

RealAudio and RealVideo System includes the ability to synchronize World Wide Web pages with audio. Thus the audio can be used as a “time line” to display new pages or frames in the Web browser or to update its content. This enables the creation of Internet slide shows, presentations, guided tours and site walk-throughs. A user can have full random access (fast forward and rewind), and the Web browser content is synchronized with the audio.

RealAudio and RealVideo System stores the information for the synchronized events in a file with a .rm file extension. RealServer streams audio and event information to RealPlayer. The event information is streamed to RealPlayer, and in turn RealPlayer sends Web page information to the Web browser telling it to update the page's content.

Another way to create a synchronized multi-media presentation is using RealPlayer Plug-in. However, since sending the Web browser to a new URL unloads RealPlayer Plug-in when the HTML page is unloaded, it is best to create separate frames for RealPlayer Controls and for the changing images.

Creating a RealVideo Events File

Creating a synchronized multimedia event using .rm files uses the **rmmerge** tool. This is a three-step process:

1. Use a text editor to create an input file specifying the display time for each URL, title, author or copyright event.
2. Use the **rmmerge** tool to generate a binary file from the text input file by typing the following command:

   ```
   rmmerge -f rmevents.dll event.txt output.rm
   ```

   where:

   `event` is the text file created in step 1

   `output` is the .rm file that contains these events
3. Merge your new events .rm file with an audio or video file using the following command:

   ```
   rmmerge <event file> <audio or video file> <output
   file>
   ```

   where:

   `<event file>` is the .rm file created in the previous step
   `<audio or video file>` is the file merged with the event file
   `<output file>` is the file resulting from the merge

   For example:

```
rmmerge event.rm audio_video.rm output.rm
```

Creating the Input File

Begin by creating a list of the URLs, titles, authors, or copyrights that you want to be shown during your presentation and the times within RealAudio or RealVideo clip when they should be displayed. The syntax for each entry should follow the format (with a space between each part of the command):

```
u starttime endtime EventURL
i starttime endtime Title
a starttime endtime Author
c starttime endtime Copyright
```

where:

u stands for URL event; each line starts with the letter u
i stands for title; each line starts with the letter i
a stands for author; each line starts with the letter a
c stands for copyright; each line starts with the letter c
starttime is the time into the clip when the new event is shown
endtime is the time into the clip when that event ends
EventURL (generally beginning with "http:" or "file:") is the Internet address for that event (usually an HTML document)

The time for **starttime** and **endtime** is:

```
[[[days:]hours:]minutes:]seconds[.tenths]
```

The lines of the input file must be in ascending order of start time. The end time should be at least one tenth of a second before the start time of the next event. The following example shows how an input file might look:

```
u 00:00:10.0 00:00:59.9 http://www.real.com/
u 00:01:00.0 00:02:00.0 http://www.mysite.com/page2/
```

This input file tells RealPlayer to send the Web browser to the RealNetworks home page ten seconds into the audio clip. One minute into the audio clip, the Web browser displays a page from "www.mysite.com."

The input file may also contain comment lines beginning with the # symbol. These comment lines are ignored by the event creation tool and are a good way to document the date that the file was created and the type of information found on each page.

Creating a RealAudio Events File

The RealAudio system allows you to create real-time on-demand multimedia presentations using the **Cevents32** utility (Windows) or **Cevents** (UNIX). Creating a synchronized multimedia event using the command line tool is a three-step process:

1. Use a text editor to create an input file specifying the display time for each URL.
2. Use the **Cevents32** or **Cevents** command line tool to generate a binary file from the text input file.
3. Place the resulting .rae file in the same directory as the .ra audio file.

Creating the Input File

Create the Input File as described on page 201.

Generating an Event File

After creating the text version of the input file, you must convert the file to a binary event file. This is done with the command line utility **Cevents32** (Windows) or **Cevents** (UNIX). It uses the following syntax:

Windows

```
cevents32 <InputTextFile> <OutputEventFile>
```

UNIX

```
cevents <InputTextFile> <OutputEventFile>
```

where:

`<InputEventFile>` is the file you just created
`<OutputEventFile>` is the same name as the file with which it will be associated.

For example, to a create synchronized multimedia presentation to accompany paradise.ra, you would generate paradise.rae using the following command:

```
cevents paradise.txt paradise.rae
```

Placing Event Files on Your Server

Copy the .rae file to the same directory as your .ra file. The .ra and .rae files must have the same name except for the file extension. RealServer automatically detects the file and send the event information to RealPlayer, which then sends it the Web browser.

Using Synchronized Multimedia from Local Files

RealPlayer can also read local synchronized multimedia files just as the Server does. In order for the local presentation to work with multiple platforms and with both Internet Explorer and Netscape Navigator, without hardcoding the directory structure, you need to follow these steps.

To play a synchronized multimedia file locally:

1. Place all .rpm/.ram, .ra, HTML and image files in one directory. The starting document should be named Index.html.

2. Do not use the slash mark (/) in your file names. Use the syntax described in “Creating the Input File,” above.

Note: If you will be using the presentation on a Macintosh computer, do not use relative path commands such as “../”.

.TXT file - convert into .rae file using Cevents compiler

```
u 10:00:45.00 &&media&&test2.html
```

.RAM and .RPM

```
file:1shapfut.ra
```

IMAGES

```
<IMG src="1shapfut.gif">
```

HREFs

```
<a href="rasld011.htm">
```

EMBEDs

```
<EMBED SRC="lecture5.rpm">
```

Using Synchronized Multimedia with Bandwidth Negotiation

When you deliver a Synchronized Multimedia presentation using Bandwidth Negotiation, you can choose to deliver the same event for all encoding formats, or you can choose to deliver different events for each encoding format.

You can create a single synchronized multimedia presentation for multiple formats, or you can create a different multimedia presentation for each encoding format. If you are working with previously created content, you can continue to use the previous method to create presentations.

For more information on this topic, see the *RealAudio and RealVideo Content Creation Guide.*

Chapter 9 RealServer Log Files

RealServer writes important status information to the following log files, which reside in the **logs** subdirectory of your RealServer installation:

Log	Default File Name	Purpose
Access Log	**pnaccess.log** or **pnaccess**	Records transactions by visitors.
Error Log	**pnerror.log** or **pnerror**	Records events and error messages about RealServer operation.
Ad Insertion Log	**pnad.log** or **pnad**	Records Ad Insertion-related events, including client transactions.

The log files are plain text files that can be read using any text editor or word processor. Each RealServer log file records transactions in a file format common to most Web servers. Transaction fields are delimited by spaces.

Access Log

The RealServer Access Log (**pnaccess.log**) displays the number of clients that are connected to your server, the names of the client machines, the clips they listened to, the times of day they connected, and errors that were generated by RealServer. This information can give you an idea of who your audience is and which clips are popular.

Reading the Access Log File

Two configuration parameters, **LoggingStyle** and **StatsMask,** control what is captured in the Access Log.

If **StatsMask** is configured to gather statistics type 3, note that the Access Log file size will grow rapidly. If you configure **StatsMask** to collect this information, be sure to review the log file frequently, as described in “Log File Maintenance.”

For more information on these settings, see Chapter 4, “Configuring and Maintaining RealServer.”

View the Access Log File by opening the **pnaccess.log** or **pnaccess** file in a word processor or text editor.

Syntax

The format for an Access Log record is:

```
<IP_address> <- -> <timestamp> "<GET filename>
<protocol>" <return_code> <bytes_sent>
[<client_ID_string>] [<client_unique_id>] <stat1>
<stat2> <stat3> <file_size> <file_time> <sent_time>
<resends> <failed_resends>
```

where:

`<IP_address>`

IP address of Client. For example:

```
      123.45.678.90
```

`<- ->`

Two hyphens for compatibility with Web server log formats.

`<timestamp>`

Time that Client accessed the file in the format:

```
[<dd>/<Mmm>/<yyyy>:<hh>:<mm>:<ss> <TZ>]
```

where TZ is the time zone expressed as the number of hours relative to the Coordinated Universal Time (Greenwich, England). For example:

```
[31/Oct/1996:13:44:32 -0800]
```

`<GET filename>`

File requested by Client. *Filename* is the relative path to the Server's **BasePath** value. For example:

```
GET /bands/fourfrosh/classics.ra
```

`<protocol>`

Protocol and version used by Client in the format:

```
PNA<type>/<number>
```

where:

`<type>` is **T** for TCP connections, **H** for PNAviaHTTP, **M** for Multicast, and blank for UDP connections.

Type appears only if the **LoggingStyle** configuration parameter is set to 1.

`<number>` is the PN protocol number. For example:

```
PNA/8
PNAT/8
```

`<return_code>`

Return code using HTTP standard error codes.

`<bytes_sent>`

Number of bytes transferred to Client during play. This field may be lower than the total size of the media file, indicating partial playback of the file. If this field is consistently low for some or all media files, it may mean that RealPlayers are able to connect to your server, but are unable to play files. Check your system Error Logs for messages relating to network system errors.

`[<client_ID_string>]`

Client ID string. This field is not part of the common Web server Access Log format; it is text sent by the Client that describes the version and type of Player being used. RealAudio Player versions 2 and 3 use the following format:

```
<platform>_<version>_<player>_<type>_<dist>_
<language>_<CPU>
```

where:

`<platform>` is the operating system that RealPlayer is running on—Win16, WinNT, Mac, and so on.

`<version>` is the operating system version number.

`<player>` is the version number of RealPlayer.

`<type>` is the type of RealPlayer.

`<dist>` is the distribution code of RealPlayer.

`<language>` is the code of RealPlayer. EN is US English.

`<CPU>` is the type of processor running the platform. 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 uses an ID string in the following format:

```
<platform><player>
```

The field descriptions are the same as the newer format. For example:

```
Win1.0.0
```

If the client is a splitter, the Client ID field contains the following string:

```
splitter
```

`<client-unique_id>`

Unique ID generated during RealPlayer installation that enables you to track details for individual players, allowing you to enhance the quality of the captured statistics. This entry is displayed only when LoggingStyle is set to 2.

This will also display the new entries for LoggingStyle 1.

<stat1>

Connection statistics sent by the Client when it completes playing a clip. These optional fields are sent only when the StatsMask configuration parameter is set to 1 or 3. The Player user can also set a preference value to block sending connection statistics. When the Client blocks connection statistics, when the Client is a splitter, or when StatsMask is set to 0, the **<stat1>** and **<stat2>** fields are replaced by **[UNKNOWN]**.

The connection statistics field starts with the string “Stat1” and has the following format:

```
[Stat1: <total> <order> <missing> <early> <late>
<audioformat>]
```

where:

<total> is the total number of packets received by the Client.

<order> is the number packets received by the Client out of order. These packets are reordered as they are being played by the Client.

<missing> is the number of packets requested by the Player, but that the Client did not receive. This is the most common problem reported in the Error Log. A low percentage of missing packets does not have a serious effect on quality; a high percentage seriously degrades media quality.

<early> is the number of requested packets received too early by the Client. If the Client receives packets too early, then older packets are discarded. This problem is rare, and it may indicate that the client’s machine is running too slow or has a bad Internet connection. However, if this problem shows up often, you need to investigate further.

<late> is the number of packets received too late by the Client. If the Client receives packets too late, the Player will have already played that portion of the media. This is a rare occurrence; if it happens often, your server’s Internet connection may not be fast enough.

<audioformat> is the name of the decoder used to play the clip. Possible values are:

dnet	RealAudio 3.0 formats

28.8	RealAudio 2.0 28.8 format
lpcJ	RealAudio 2.0 14.4 format

For example:

```
[Stat1: 641 0 0 0 0 dnet]
```

<stat2>

Extended connection statistics sent by the Player when it completes playing a clip. These statistics are supported by all Players.

These optional fields are sent only when the **StatsMask** configuration parameter is set to 2 or 3. The Player user can also set a preference value to block sending connection statistics. When the Client blocks connection statistics, when the Client is a splitter, or when StatsMask is set to 0, the **<stat1>** and **<stat2>** fields are replaced by **[UNKNOWN]**.

The extended connection statistics field starts with the string “Stat2” and has the following format:

```
[Stat2: <bandwidth> <available> <highest><lowest>
<average> <requested> <received><late> <rebuffering>
<type> <startup> <videoformat>]
```

where:

<bandwidth> is the bandwidth in bits per second of the clip.

<available> is the average bandwidth in bits per second available to the user while the clip was playing.

<highest> is the highest time in milliseconds between the Client requesting a resent packet and receiving the packet.

<lowest> is the lowest time in milliseconds between the Client requesting a packet be resent and receiving the packet.

<average> is the average time in milliseconds between the Client requesting a resent packet and receiving the packet for all resent packets.

<requested> is the number of resent packets requested by the Client.

<received> is the total number of resent packets received by the Client.

<late> is the number of resent packets received by the Client too late.

<rebuffering> is the rebuffering percentage for the clip.

<type> is the transport type for the connection. Values are: **0** (UDP), **1** (TCP), **2** (IP Multicast), and **3** (PNAviaHTTP)

<startup> is the time in milliseconds from the Client sending the first packet to the Server to the Client receiving the first packet from the Server.

<video format> is the name of the video decoder used to play the clip. Values are:
pnrv - RealVideo 1.0
clrv - RealVideo (Fractal)

For example:

```
[Stat2: 15234 15552 0 0 0 0 0 0 0 0 220 28.8]
```

<stat3>

Statistics sent by the client which capture behavior observed by a data renderer. These statistics are only sent to the RealServer when the RealPlayer preferences are set to send statistics. Moreover, these fields are only requested by the RealServer when the StatsMask configuration setting is set to 4, 5, 6, or 7. Pipe '|' characters are used as field separators in the Stat3 record. Multiple Stat3 records are delimited by a semi-colon character (;).

```
[Stat3:<timestamp>|<elapse time>|<action>|<(data)>;]
```

where:

<timestamp> Time when action occurred. Time is recorded in milliseconds, and is relative to the connect time of the client.

<elapse time> Elapse time of the clip when the behavior occurred, where time is recorded in milliseconds.

<action> Behavior, where values are:

ABORT client stop/application closure (not natural end of clip play)

RESUME resume play after a pause, seek or stop

PAUSE client pause

SEEKSTART client begin seek

RECSTART client begin record (RealPlayer Plus only)

RECEND end record (RealPlayer Plus only)

CLICK image map click-through

ADSTART elapse time of an advertisment clip when the client started play

ADEND elapse time of an advertisment clip when the client ended play

<(data)> Optional field containing a string which defines the action. The string found in this field is surrounded by parenthesis characters "(" and ")". Individual values found in the field are separated by the comma "," character.

For example:

```
[Stat3:6000|10000|PAUSE|;78600|10000|RESUME|;200000|
1350000|SEEKSTART|;2004500|3280000|RESUME|;]
```

Note: Note, a Stat3 record is not generated at either the initial play or natural end of .rm file stream. This information is directly derived from the main section of the Access Log.

Data Definitions Within the **Stat3** Record

CLICK Where the data field defines a CLICK action, the following values appear:

```
(<horizontal coordinate>, <vertical coordinate>,
<action_tag>)
```

where:

<horizontal coordinate> is the horizontal coordinate the client interacted with

`<vertical coordinate>` is the vertical coordinate the client interacted with

`<action_tag>` is the action comprising the image map values are:

`<PLAYER=" ">`

`<URL=" ">`

`<SEEK= >`

For example:

```
[Stat3:80000|80000|CLICK|(28,43,URL="http://www.
realnetworks.com");]
```

`ADSTART` Where the data field defines an ADSTART action, the following values appear:

`(<AD URL>,<elapse time>)`

where:

`<AD URL>` is the relative path and file name of the advertisment media

`<elapse time>` is the elapse time in milliseconds of the advertisment media when play began

For example:

```
[Stat3:1000|0|ADSTART|(/ads/mci.rm,0);]
```

`ADEND` Where the data field defines an ADEND action, the following values appear:

`(<AD URL>,<elapse time>,<ad duration>)`

where:

`<AD URL>` is the relative path and file name of the advertisment media

`<elapse time>` is the elapse time in milliseconds of the advertisment media when play ended

`<ad duration>` total duration time in milliseconds of the advertisment media

For example:

```
[Stat3:1000|0|ADSTART|(/ads/mci.rm,0);31000|0|AD
END|(/ads/mci.rm,30000);]
```

`<file_size>`

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. This field appears only if the LoggingStyle configuration parameter is set to 1.

For live broadcasts, file_size is always 0.

`<file_time>`

Total length, in seconds, of media stored in the media file. This field appears only if the LoggingStyle configuration parameter is set to 1.

For live broadcasts, file_time is always 0.

`<sent_time>`

Total length, in seconds, of the media sent to the Player. This field appears only if the LoggingStyle configuration parameter is set to 1.

`<resends>`

Number of packets successfully resent because of transmission errors. This field appears only if the LoggingStyle configuration parameter is set to 1.

`<failed_resends>`

Number of packets not successfully resent in time to correct transmission errors. This field appears only if the LoggingStyle configuration parameter is set to 1.

Example Access Log

The following example shows three Access Log entries:

```
172.16.2.139 - - [04/Nov/1996:14:45:57 -0700] "GET
newclips/realcool.ra PNA/8" 200 590976
[Win95_4.0_3.0.0.19_play32_PN01_EN_586] [Stat1: 2592 0
0 0 0 28.8][Stat2: 15234 15552 0 0 0 0 0 0 0 0 220
28.8] 590976 310 310 0 0
172.16.2.139 - - [04/Nov/1996:14:53:49 -0700] "GET
classic/xyz144.ra PNAT/8" 200 4
[Win95_4.0_3.0.0.19_play32_PN01_EN_586] [UNKNOWN] 5580
5 0 0 0
172.16.2.139 - - [04/Nov/1996:16:01:10 -0700] "GET
speeches/carter.ra PNA/5" 200 55680 [Win1.0.0] [Stat1:
229 0 0 0 0] 630020 630 55 0 0
```

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.

The default error log name is **pnerror.log**, but you can change the name of the file errors are recorded in by changing the **ErrorLogFilePath** specified in the configuration file. For information on this setting, see Chapter 4, “Configuring and Maintaining RealServer.”

View the text of the Error Log using a word processor or text editor.

Syntax

Error messages are recorded in the Error Log in the following format:

```
[Date] [Time] [Servername](ProcessID) : [Error
Message]
```

If the RealServer is running on a UNIX platform, the Error Log also includes information about extreme errors such as faulty streams, memory corruption, and product incompatibilities. The following is a report from a server that has had errors from which it recovered:

```
***13-Nov-97 13:50:47.175 pnserver(6905): TRAPPED
FAULT: Please File Bug Report
***13-Nov-97 13:50:47.177 pnserver(6905): Fault
Report: 5.0.1.1-rvserver-build-223 FreeBSD-2.1.x
0x00015516: 53ca6850 d8680001 68000153 000153f3
0cbe75e8 540b6800 6be80001 a1000cbe
0x00001b9c: 4074c085 0b6a006a 12c51be8 6a006a00
c512e806 006a0012 09e80a6a 6a0012c5

***13-Nov-97 13:50:47.177 pnserver(6905): TRAPPED
FAULT: Attempting Crash Avoidance...
***13-Nov-97 13:50:47.177 pnserver(6905): Fault caused
by type 9 client from 172.16.2.212
```

```
***13-Nov-97 13:50:47.178 pnserver(6905): TRAPPED
FAULT: Crash Avoidance Successful
```

The following is a sample entry from a server which was not able to recover from an error:

```
***13-Nov-97 13:49:23.105 pnserver(6890): FATAL ERROR:
Couldn't Handle Fault: Terminating...
***13-Nov-97 13:49:23.105 pnserver(6890): FATAL ERROR:
Please File Bug Report
```

If you have an entry that refers to a fatal error, contact the RealNetworks Technical Support Department for assistance.

Example Error Log

A sample error message looks like this:

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

You can also have RealServer send messages to your e-mail address to notify you when certain thresholds are exceeded.

Ad Insertion Log

The RealServer logs information about advertising presentation activity to the Ad Insertion Log (**pnad.log**). Each time an in-stream ad is inserted into a content clip, a log entry is generated in **pnad.log**, identifying the PLAYERID of the user, the IP address of the client, the content clip, the ad clip, and more. You can sort on any or all of these fields and compile reports. This information can give you an idea of who your audience is and which clips are popular.

The Ad Insertion Log does not conform to common Web logging formats.

Syntax

The syntax for one record in the **pnad.log** file is:

```
<IP_address> <- -> <timestamp> <GET ad_filename>
<PLAYERID> <content_URL> <off_set_time>
<ad_play_duration> <ad_insert_type> <status>
```

Parameter	**Description**
<IP_address>	IP address of the client
<- ->	Complies with common Web server log format.
<timestamp>	Client time that the ad was sent to the client, in the form: **<dd>/<mm>/<yyyy>:<hh>:<mm>:<ss> <TZ>** where TZ is the time zone expressed as the number of hours relative to Greenwich mean time.
<GET ad_filename>	Path and name of ad file relative to the base content path specified in the RealServer configuration file.
<PLAYERID>	General user ID for the RealPlayer (can be matched to the user if the user has registered at your site).
<content_URL>	URL of content clip broadcast to the RealPlayer.
<off_set_time>	Clock time (elapsed time), in milliseconds from

Parameter	**Description**
	beginning of content clip to point at which the ad was inserted.
`<ad_play_duration>`	The length of time (in milliseconds) the ad clip was displayed by the RealPlayer.
`<ad_insert_type>`	Type of ad insertion: • 1 Lead ad insertion, on-demand content • 2 In-line ad insertion, on-demand content • 3 End ad insertion, on-demand content • 4 Lead ad insertion, live content
`<status>`	Status of the ad insertion. Possible values are: • 0 Successful • 1 Failed, ad file not found • 2 Failed, incompatible content type

Example

The following is a sample Ad Insertion Log file:

```
172.16.8.194 - - [23/Sep/1997:20:43:40 -0700] GET
d:\inetpub\rvroot\build13\content/20kbs\usr_20.rm
3c633661-f971-11d0-fea1-85071a4854c3 20kbs/wc20.rm
30466 30466 1 0
172.16.8.194 - - [23/Sep/1997:20:44:18 -0700] GET
d:\inetpub\rvroot\build13\content/20kbs\mci_20.rm
3c633661-f971-11d0-fea1-85071a4854c3 20kbs/wc20.rm
101464 30132 2 0
172.16.8.194 - - [23/Sep/1997:20:45:35 -0700] GET
d:\inetpub\rvroot\build13\content/56kbs\usr_56.rm
3c633661-f971-11d0-fea1-85071a4854c3 56kbs/wc56.rm
30373 30373 1 0
172.16.8.194 - - [23/Sep/1997:20:51:07 -0700] GET
d:\inetpub\rvroot\build13\content/56kbs\usr_56.rm
3c633661-f971-11d0-fea1-85071a4854c3 56kbs/wc56.rm
30373 30373 1 0
172.16.6.228 - - [23/Sep/1997:20:51:53 -0700] GET
d:\inetpub\rvroot\build13\content/56kbs\usr_56.rm
```

```
f95cc2e1-2473-11d1-ff93-7d124ac71ce6 56kbs/wc56.rm
30373 30373 1 0
172.16.6.228 - - [23/Sep/1997:20:53:16 -0700] GET
d:\inetpub\rvroot\build13\content/56kbs\mci_56.rm
f95cc2e1-2473-11d1-ff93-7d124ac71ce6 56kbs/wc56.rm
181310 30512 2 0
172.16.8.194 - - [23/Sep/1997:20:54:11 -0700] GET
d:\inetpub\rvroot\build13\content/20kbs\usr_20.rm
3c633661-f971-11d0-fea1-85071a4854c3 20kbs/wc20.rm
30466 30466 1 0
```

Log File Maintenance

Because new records are appended to log files for each event (error or transaction) these files can become rather large quickly. To keep your log files a manageable size, it is recommended that you review them regularly and archive them to maintain a record of your server’s performance.

You should read your log files on a regular basis. How frequently you read them depends on the amount of traffic your RealServer handles and if you are encountering any problems.

Archiving Log Files

You can archive a log file by changing the filename in the RealServer configuration file. RealServer writes any new information to the new file, and the old file retains the archival information.

Using the Java Performance Monitor

Windows and UNIX

To use the Java Performance Monitor to change RealServer configuration settings:

1. Start the Java Performance Monitor and log on to the server.
2. Click **Configuration**.

3. In the **Config Variable Groups** list, double-click the Logging setting. On the **Config Variables** list, double-click the setting you want to change. The value appears on the far right.

   - **ErrorLogPath** for the Error Log
   - **LogPath** for the Access Log
   - **LogPath** for the Advertising Presentation Log

4. Type a new name for the log file and click **OK**.
5. When you have finished editing Configuration settings, click **Monitor** to close the Configuration window and return to the Java Performance Monitor.

Restart the RealServer to cause the new configuration settings to take effect.

Using the System Manager

Windows

1. Connect System Manager to the RealServer whose log file you want to rename.
2. On the **Server** menu, click **Configuration**.
3. Select the setting that you want to change:

   - **ErrorLogPath** for the Error Log
   - **LogPath** for the Access Log
   - **LogPath** for the Advertising Presentation Log

4. Type a new name for the log file and click **OK**.
5. RealServer starts writing to the new file.

Restart the RealServer to cause the new configuration settings to take effect.

UNIX

1. For UNIX servers, UNIX continues to write to the new, renamed file until you send a SIGHUP signal. Upon receiving the SIGHUP signal, RealServer closes the existing, now renamed log file. When the next event occurs, RealServer opens a new log file using the settings in the configuration file.

2. For example, to change your Access Log file from **pnaccess**, rename it **access1**. RealServer continues to write to **access1**. Once RealServer receives a SIGHUP signal, it closes **access1** and writes the next event to a new, empty file **pnaccess**.
3. If you do not want to keep your log files, simply delete the log file and issue a SIGHUP signal. Once RealServer receives this signal, it opens a new empty file with the same name.
4. To send a SIGHUP signal, use the **kill** command with the server's process id. If you are in the logs directory, use the following syntax:

```
kill -hup 'cat pnserver.pid'
```

Appendix A Server Commands

cevents

The **cevents** utility takes the supplied text file containing the multimedia event descriptions and converts it to a compiled events file.

Syntax

```
cevents <inputfile> <outputfile>
```

Example

Windows

```
cevent32.exe events.txt events.rae
```

UNIX

```
cevents events.txt events.rae
```

where

`-v`	Displays the version information of the **cevents** utility.
`<inputfile>`	Inputfile has the following format:

```
u <starttime> <endtime> <URL>
```

where each entry is on a single line with each value separated by spaces and where:

`<starttime>` is the start time of the event in HH:MM:SS.t format

`<endtime>` is the end time of the event in HH:MM:SS.t format

`<URL>` is the URL of the page for the event

Example

```
u 00:00:20.0 00:00:59.9 http://204.71.156.200/demo/banner.htm
u 00:01:00.0 00:02:00.0 http://204.71.156.200/demo/pictures.htm
u 00:02:05.0 00:03:00.0 http://204.71.156.200/demo/pictures2.htm
```

If you are using frames in your synchronized multimedia, you can target a specific frame by putting the name of your target frame in the event file as follows:

```
u 00:00:20.0 00:00:59.9
&&targetframename&&http://204.71.156.200/demo/banner.htm
u 00:01:00.0 00:02:00.0
&&targetframename&&http://204.71.156.200/demo/pictures.htm
u 00:02:05.0 00:03:00.0
&&targetframename&&http://204.71.156.200/demo/pictures2.ht
m
```

kill

Syntax

```
kill -HUP <processID>
```

where **<ProcessID>** is the process id of RealServer.

You can reconfigure a running server on a UNIX machine using the command-line interface. First, change the parameters you want by editing the **server.cfg** file. Then use the **kill** command with the **-HUP** flags. This forces the Server to reload with the new configuration settings.

If you do not know the process id, run **ps** to obtain it. The parameters for **ps** depend upon the version of UNIX you are using:

UNIX platform	Command
BSDI, FreeBSD, LINUX, SunOS	**ps -aux | grep pns**
AIX, HP-UX, DEC UNIX, IRIX, SOLARIS	**ps -ef | grep pns**

pnserver

The **pnserver** command starts RealServer.

Syntax

```
     pnserver [-v] [-n] [-p port] <configfile>
```

where:

`-v` Displays the version information of RealServer. This includes the platform, build and release tags used to identify a particular release.

`-n` (UNIX only) Do not detach from the command terminal. This prevents the server from becoming a daemon process.

`-p port` Use the supplied TCP port as the connection port for the server. This overrides any configuration file setting.

`<configfile>` Specifies a file of configuration settings for pnserver. If no file is specified, uses the settings in **server.cfg**. If another file is specified, settings in this file override values in **server.cfg**. For information on configuration settings, see Chapter 4, “Configuring and Maintaining RealServer.”

Example

Windows

```
     bin\pnserver server.cfg
```

UNIX

```
     bin/pnserver server.cfg
```

raconv

Bandwidth negotiation file converter. The **raconv** utility takes the supplied files and converts them to the Bandwidth Negotiation naming scheme and places them in the specified directory. More than one file name can be supplied.

Syntax

```
raconv [-v] [-f] <file names> directory
```

where:

`-v` Displays the version information of the utility. This includes the platform, build and release tags used to identify a particular release.

`-f` Forces raconv to overwrite existing files with the same filenames.

Example

The file newband.ra is encoded in three formats: RealAudio 2.0 - 28.8, RealAudio 3.0 - ISDN Mono, and RealAudio 3.0 - Dual ISDN Stereo. The files are stored in the following locations:

```
/usr/rawdata/old28_8/newband.ra
/usr/rawdata/isdnmono/newband.ra
/usr/rawdata/dualisdn/newband.ra
```

To create these files, issue the following commands:

```
raconv /usr/rawdata/old28_8/newband.ra /usr/rafiles
raconv /usr/rawdata/isdnmono/newband.ra /usr/rafiles
raconv /usr/rawdata/dualisdn/newband.ra /usr/rafiles
```

The **raconv** utility program creates a directory named **/usr/rafiles/newband.ra** that contains the files 28_8.36, dnet.50, and dnet.100.

rssm

System Manager enables remote monitoring and administration of RealServer from the UNIX command line. To connect System Manager to a Server, set hostname to the DNS name or IP address of the Server. If the Server is running on a port other then 7070, specify the port number.

System Manager can monitor a RealServer running on any platform. Information provided by System Manager includes the number and status of Player connections, System Manager connections, Unknown connections (connections currently being negotiated with the Server), and Total connections. This information can then be used to monitor activity on RealServer on a regular basis.

System Manager runs in two modes: interactive and non-interactive. When the System Manager is in the non-interactive mode, information is automatically appended to STDOUT every 5 minutes, unless that time span is modified by the **`-l`** command. The System Manager accepts commands from the command line; however, it does not prompt you.

The interactive mode is started with the **`-i`** command, which enables the System Manager to print prompts and accept commands from the command line.

Syntax

```
rssm [-v] [-l <update>] [-p <password>] [-c]
[-i] [-k] <hostname[:port]>
```

where

`-v` Displays the version information of the System Manager. This includes the platform, build and release tags used to identify a particular release.

`-l update` Sets the update period for output to the screen to update seconds.

`-p password` Provides the password required by System Manager to connect to the Server. If this option is not used the System Manager prompts for the password. This feature is not secure. The password is easily accessible to knowledgeable searchers. The password is required each time you want to start monitoring a Server. You can include **`-p <password>`** in the **rssm** command line in automatic monitoring scripts to avoid having to enter the password interactively.

`-c` Connects to the server to verify it is still accepting connections and then exits. Prints a message if the connection fails and the exit status is non-zero.

`-i` Starts interactive mode and permits entry of the commands listed in the command section.

`-k` Does DNS lookups on incoming IP addresses to translate them to full domain names. This command can slow down responses on System Manager. If you are experiencing delays in System Manager information or in response to commands, make sure that this feature is turned off.

Interactive Commands

After starting System Manager in interactive mode by using the **`-i`** command-line option, you can enter any of the following interactive commands at the System Manager prompt (>):

Command	Function
c	Displays the current configuration after it has been retrieved using the **t** command.
e	Resets peak usage value.
g	Displays the time that the peak usage value was last reset.
h or ?	Prints a list of commands.
i	Prints the Server's version number and platform.
k	Begins collating hostname information for connected clients by doing reverse DNS lookups on the IP numbers provided by the Server.
l	Provides the current list of connected clients.
n	Modifies a Server configuration variable.
o	Prints # of Players, Server Monitors, unknowns, and total connections to STDOUT every five minutes, or the number of seconds specified by the **-l** option on the command line. This command can be toggled to start and stop.
p	Prints Server license information.
s	Prints a single line of summarized status information.
u	Continuous display. Updates whenever a client status changes.
x	Exit the program.

System Manager displays the information about clients connected to the Server in the following format:

```
<client>  <name>
```

where:

<client> is the type of client connected (Manager or Player).

`<name>` is the domain name or IP address of that client.

For example, a client listing might look like:

```
Manager 204.71.154.93
Player  204.71.153.24
```

If you prefer to receive System Manager information in a report, use the **`-l`** option and append the output to a file. To do this, use the following command:

```
rssm -l<seconds> <hostname>[:port] >>
<reportname.txt>
```

where:

`<seconds>` is the number of seconds between reports.

`<hostname>` is the name of the computer you are collecting data from.

`<reportname.txt>` is the name of the report that the information is appended to.

Example

To monitor a RealServer in interactive mode, with updates every 20 seconds and fully qualified host names for clients, use the following command:

```
rssm -l 20 -k -i yourServer:7070
```

Appendix B Authentication Text File and Database Structure

This chapter describes the structure of the data storage text files, and of the database files. If you use a database for which there is no included template, create the data source to have the database file structure described herein.

For additional information on identifying which method RealServer is to use, see the **AuthDBPlugin** configuration setting in see Chapter 4, “Configuring and Maintaining RealServer”.

Text File

When you first install RealServer and the Authentication feature, no text files exist. After you start the RealServer for the first time, it creates the appropriate directory structure and initial text files. Every time you start the RealServer thereafter, it uses these directories and files.

The following directory structure is created under the **Commerce** directory:

Directory	Contents
dbdata	**ppvbasic.txt**
dbdata\users	This directory is initially blank
dbdata\guids	This directory is initially blank
dbdata\logs	**reglog.txt, accesslog.txt**
dbdata\redirect	This directory is initially blank

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 dbdata directory

This directory contains four subdirectories and one file: **ppvbasic.txt**.

The **ppvbasic.txt** file indicates to RealServer that this directory is the location for data storage.

Note: 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.

The dbdata\users Directory

The files in this directory are named *username*.txt, where *username* is also the User ID. This directory contains one file per registered user.

The first line of each *username*.txt file has the following form and is different from subsequent lines in the file:

```
password;uuid;uuid_writeable
```

where:

`password` If **AuthMode** is set to 1 (User-based), this stores the password. Otherwise an asterisk (*).

Note: Passwords are encrypted. See “Using the Password Tool” on page 146.

`uuid` If **AuthMode** is set to 0 (player-based) this stores playerID. Otherwise, this field is an asterisk (*). **`Uuid`** or **`password`** will be an asterisk *; depends on **AuthMode** configuration setting.

`uuid_writeable` A flag set and used by the Server: (0=playerID is in database, 1=record created, but playerID not yet registered)

The second and subsequent lines of each *username*.txt 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
```

Example

The example file user1.txt has the following content, when **AuthMode** is 0:

```
*;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
```

The dbdata\guids Directory

The files in this directory are given the names of the GUIDs from the registered RealPlayers, one per registered user. Each file contains only the name of the associated user name. For example, a file such as “1234this2is3my4guid89” contains the name of the user, “user1”.

The dbdata\logs Directory

This directory contains two files: **reglog.txt** and **accesslog.txt**.

Reglog.txt

Each line of this file represents the result of an attempt to register a visitor.

The **reglog.txt** file has the following format:

Syntax

```
status;userid;uuid;IP;register_time;url_redirect
```

where:

`status` Result of user’s attempt to connect.

0 = Success
1 = Failed (playerID not readable)
2 = Failed (playerID not unique—already registered under different username)

3 = Failed (3.0 Player or older)

`userid` Unique name of up to 50 characters.

`uuid` If **AuthMode** is set to 0 (Player-based) this stores playerID.

`ip` IP address from which user is attempting to connect.

`request_time` Time of connection request.

`url_redirect` If connection failed, URL to which user was redirected (see **redirect.txt**).

Example

```
0;user1;00001d00-0901-11d1-8b06-
00a024406d59;255.255.255.0;
14:07:63;*
2;user2; 00001d02-0901-11d1-8b06-
00a024406d59;255.255.255.0;
14:08:63;http://redirect.com/url.html
```

Accesslog.txt

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

Syntax

```
status;userid;uuid;ip;url;permission_type;
permission_on;start_time;end_time;total_time;
why_disconnect
```

where:

`status` Result of user’s attempt to connect. 0 = Success, 1=playerID not readable, 2=playerID non-unique (already registered under different username) 3=no

	username (unique record with username not yet created).
`userid`	Unique name of up to 50 characters.
`uuid`	If **AuthMode** is set to 0 (Player-based), this stores playerID.
`ip`	IP address from which user is attempting to connect.
`url`	Secured clip user is attempted to access.
`permission_type`	Permission type associated with access: 0=event-based, 1=calendar, 2=duration, 3=none
`permission_on`	Permission type associated with url: 0=file (individual clip), 1=directory, 2=none.
`start_time`	Time/date clip started playing.
`end_time`	Time/date clip stopped playing.
`total_time`	Total time clip played.
`why_disconnect`	0=client disconnected voluntarily, 1=server access expired (duration exceeded time banked or calendar date set).

Example

```
0;user1;1234this2is3my4guid89;255.255.255.0;/secure/
clip1.rm;0;0;14:07:63;14:08:64;00:01:01;0
```

The dbdata\redirect Directory

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.

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.

Database structure

This section describes the structure of the database templates included with RealServer. If you create your own database files to use with RealServer, use the format described here.

By default, RealServer uses text files (described in the preceding section) and not database files.

The database templates include five tables:

- **users**
- **permissions**
- **redirect**
- **register_log**
- **access_log**

The Users and Permissions tables are the most important; they contain the lists of who's registered and with what permissions.

users Table	
userid	User name of up to 50 characters. Ties to Permissions table.
password	If **AuthMode** is set to 1 (User-based), this stores the password. Otherwise blank. **Note:** Passwords are encrypted. See "Using the Password Tool" on page 146.
uuid	If **AuthMode** is set to 0 (Player-based) this stores playerID. Otherwise blank.
uuid_writeable	A flag set and used by the server: 0=playerID is in database, 1=record created, but playerID is not yet registered with the server.

permissions Table	
userid	User name of up to 50 characters. Ties to Users table.
url	URL of secure directory or clip.
url_type	Whether URL is directory or clip: 0 = directory, 1 = clip.
Permission_type	Access type: 0 = event-based, 1 = calendar, 2 = duration
expires	if permission_type = 1, this is expiration date/time, in format MM/DD/YYYY:HH:MM:SS. Otherwise blank.
debitted_time	if permission_type = 2, this is time remaining (in seconds) Otherwise blank.

redirect Table	
url	URL of any secure clip or directory.
url_redirect	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.

register_log Table	
status	Result of user's attempt to connect: 0 = Success

	1 = Failed (playerID not readable) 2 = Failed (playerID not unique—already registered under different username) 3 = Failed (3.0 Player or older)
userid	Unique name of up to 50 characters.
uuid	If **AuthMode** is set to 0 (Player-based) this stores playerID. Otherwise blank.
ip	IP address from which user is attempting to connect.
request_time	Time of connection request.
url_redirect	If connection failed, URL to which user was redirected (see Redirect Table, above).

access_log Table	
status	Result of user's attempt to connect: 0 = access to clip granted, 1=denied
userid	Unique name of up to 50 characters.
uuid	If **AuthMode** is set to 0 (Player-based) this stores playerID. Otherwise blank.
ip	IP address from which user is attempting to connect
url	Secured clip user is attempted to access.
permission_type	Permission type associated with access: 0=event-based, 1=calendar, 2=duration, 3=none
permission_on	Permission type associated with url: 0=file (individual clip), 1=directory, 2=none.
start_time	Time/date clip started playing.
end_time	Time/date clip stopped playing.
total_time	Total time clip played.
why_disconnect	0=client disconnected voluntarily, 1=server access expired (duration exceeded time banked or calendar date set).

Appendix C Creating and Serving Content for Mobile Players

Mobile Players such as the Audible Player by Audible, Inc. are an important new class of consumer products that can be used to access and playback RealAudio files. With these new products, you can take audio content in RealAudio format that has been downloaded to a client computer and transfer them to a hand-held Mobile Player that can then playback the content at your convenience. RealEncoder, RealPublisher, and RealServer support the creation and serving of content to Mobile Players such as the Audible Player.

Encoded files can be easily created using the RealEncoder or RealPublisher. These files can then be linked using .ram files on a Web page. When the user clicks on such a link, the RealAudio file is downloaded to the client computer using TCP at a rate faster than the streaming bit-rate. The rate at which this data transfer takes place is configurable by the RealServer administrator.

There are three steps to implementing Mobile Player support:

1. Encode audio content.
2. Create the .ram file to use on your Web site.
3. Configure the RealServer to allow Mobile Player support. This step need only be done once.

Encoding Content for Mobile Download

In RealEncoder or RealPublisher, click the **Mobile Playback** checkbox and encode audio files as usual. The Audible Player supports only the 6.5 Kbps voice codec that is available as part of the RealEncoder and RealPublisher products. Currently, (and in the near future) Mobile Players can only decode and play RealAudio files or .rm files with only-audio data.

If you are using the command-line encoder, you will need to use the **`/P`** option. Valid arguments for **`/P`** are “0” (disabled) and “1” (enabled).

You can also use the **rmedit** tool to modify a pre-existing .rm file created using RealEncoder 5.0 and enable it for Mobile Playback. See the *RealAudio and RealVideo Content Creation Guide* for more information on encoding files with these options.

Creating .ram Files for Mobile Download

To make content downloadable for Mobile Playback, you will need to create a .ram file on your Web server that points to the encoded .rm file and contains the “**`mobileplayback`**” .ram file tag. Once the appropriate .rm file has been created you can create a .ram file that contains an entry of the form:

```
pnm://my.server.com/basepath/file.rm?mobileplayback=”1”
```

where **`my.server.com`** is the name of the machine that is running your RealServer, **`basepath`** is the location on the machine that contains your .rm files and **`file.rm`** is the name of the file you want to make available for Mobile Download. You can also add other .ram file tags such as **`starttime`** and **`endtime`** to the entry. In addition, you can also create .ram files with multiple URLs as before. With multiple Mobile Download enabled URLs, the files are downloaded in succession to the client computer. If you insert URLs without the **`mobileplayback`** .ram file tag, these URLs are streamed to the client player instead of being downloaded.

Leaving the **`mobileplayback`** tag out of the .ram file enables you to make the same .rm file streamable (instead of downloadable) to the client computer.

Configuring Servers for Mobile Download

Add the **MobilePlaybackOversendRate** configuration setting to the **server.cfg** file. For information on how to add this setting, see Chapter 4, "Configuring and Maintaining RealServer."

The syntax for this variable is as follows:

MobilePlaybackOversendRate

The multiple of the streaming rate that is used to determine the effective bit rate at which the server transfers files to Mobile Playback enabled RealPlayers via TCP. The file is sent via TCP at the effective bit rate or the network bandwidth, whichever is lower.

Default value	10
Range of values	1-100

Syntax

```
MobilePlaybackOversendRate <number>
```

Example

To serve 6.5 Kbps encoded files at 19.5 Kbps:

```
MobilePlaybackOversendRate 3
```

Index