RealMedia Production Guide

RealSystem G2 Preview Release

RealNetworks, Inc

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

http://www.real.com

contents

contents

contents

contents

Introduction

Welcome to RealSystem, the most sophisticated system for streaming multimedia files across a network. This manual will help you produce your multimedia presentation, whether you want to stream a video from your home page or create a multimedia extravaganza with video, audio, animation, still images, and text.

Note

The HTML version of this manual, available at http://www.real.com, contains RealMedia examples you can view with RealPlayer.

How to Use this Manual

This manual tells how to assemble a RealMedia presentation. Although it gives tips on producing great content, the more you know about producing audio, video, and graphics, the faster you'll put together a great streaming presentation. If you know the basics of HTML, you'll find it easy to pick up SMIL, the language used to assemble a RealMedia presentation.

Chapter 1: What's New in RealSystem G2?

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

Chapter 2: Creating a RealMedia Presentation

This chapter walks you through the steps you take to put together a streaming multimedia presentation. It also explains how to target a bandwidth connection.

Chapter 3: Producing Audio

This chapter gives you pointers on capturing and digitizing high-quality audio. It then gives you the background you need to encode the audio source for streaming.

Chapter 4: Producing Video

Read this chapter to learn how to capture high-quality video and optimize it for RealMedia streaming.

Chapter 5: Producing Animation

Macromedia's Flash animation paired with a RealAudio soundtrack produces dazzling animated presentations. This chapter tells you how to create RealFlash content.

Chapter 6: Assembling a Presentation with SMIL

After you create your multimedia files, you write a simple file that pulls the presentation together. This chapter tells you how to write this file to specify when and how each part of your presentation plays.

Chapter 7: Playing a Presentation in a Web Page

To integrate your presentation seamlessly in your Web page instead of playing it back in RealPlayer, follow the instructions in this chapter.

Chapter 8: Inserting Ads in your Presentation

Using RealServer, you can insert ads into your streaming presentation. This chapter explains how to prepare your advertising media and set up ad rotation.

Chapter 9: Delivering Your Presentation

This chapter provides step-by-step instructions for moving your presentation files to RealServer and linking your Web page to them. It also tells how you can use a standard Web server for simple media streaming.

Chapter 10: Broadcasting Live

Refer to this chapter if you plan to broadcast an audio or video event live.

Chapter 11: Using RealMedia Logos

Look here for guidelines on using RealMedia logos on your Web pages.

Appendix A: File Type Reference

This appendix provides a simple reference for file types used in RealMedia streaming.

Conventions in this Manual

The following table explains the conventions used in this manual.

Convention	Meaning
variables	Italicized text represents variables. Substitute values appropriate for your situation.
[options]	Square brackets indicate optional values you may or may not need to use.
choice 1|choice 2	Vertical lines separate values you can choose between.
...	Ellipses indicate nonessential information omitted from the example.

Additional RealSystem Resources

In addition to this manual, you may need the following RealNetworks resources, available at **http://www.real.com**:

- *RealText Content Creation Guide*

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

- *RealPix Content Creation Guide*

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

- *RealServer Administration Guide*

  The basic reference for the RealServer administrator, this manual explains how to set up, configure, and run RealServer to stream multimedia. You need this manual only if you are running RealServer yourself.

- RealMedia SDK

  The RealMedia SDK lets you integrate applications with the RealMedia system. You need the SDK and its documentation to create a new plug-in for RealServer or RealPlayer. A knowledge of programming is required to use the SDK.

Chapter 1: What's New in RealSystem G2?

If you're familiar with RealSystem, this chapter gives you a quick look at the many changes that affect how you produce streaming media in RealSystem G2.

RealSystem's New Features

Based on the most advanced technology available, the open, end-to-end architecture of RealSystem G2 changes the nature of streaming multimedia across a network. With RealSystem G2, you have far more possibilities for creating Web-based multimedia than before.

New File Streaming Possibilities

Popular File Formats Stream Natively

With RealSystem, you can now stream many popular audio and video formats. Stream WAV without conversion to RealAudio, for example, AVI without conversion to RealVideo.

Additional Information

See "Choosing Media to Stream" on page 9.

Text and Slide Shows Now Stream

RealText and RealPix let you stream text and create streaming slide shows that use special effects such as fades.

Additional Information

See "Choosing Media to Stream" on page 9.

Open Plug-In Architecture Streams New File Types

RealSystem's open architecture lets RealNetworks' development partners create plug-ins to stream virtually any file type. Automatic download of plug-ins ensures that RealPlayer 6.0 users can play new RealMedia streaming file types as soon as they are introduced.

Additional Information

Visit **http://www.real.com** for information on becoming a RealNetworks partner.

Advancements in RealAudio and RealVideo

New Audio Codecs Provide Superior Sound Quality

RealSystem G2 introduces a new family of RealAudio Codecs that provides fast encoding, superior sound, and the ability to encode a single file for delivery at different bit rates.

Additional Information

See "Supporting Multiple Bandwidth Connections" on page 13 for an overview. "Choosing a RealAudio Codec" on page 24 lists the new Codecs.

Multiple Audio Streams Play through the Same Codec

RealPlayer 6.0 removes the restriction that two RealAudio streams played simultaneously must be encoded with different Codecs. The new RealPlayer can now play multiple streams that use the same Codec.

Easier Presentation Assembly

SMIL Files Coordinate Presentations

For presentations that include more than one file, you create a SMIL file to specify how and when each file plays. SMIL, which stands for Synchronized Multimedia Integration Language, is a standardized language that uses a simple mark-up similar to HTML to coordinate a streaming presentation.

Additional Information

For a look at SMIL features, see "Writing a SMIL File" on page 17. Refer to Chapter 6 beginning on page 43 for instructions on using SMIL.

Bandwidth Negotiation through Multiply Encoded Files or SMIL

RealSystem G2 introduces simpler methods for supporting multiple bandwidth connections. New Codecs allow a single RealMedia file to contain multiply encoded versions of the source. Or you can let RealPlayer choose between different versions of a presentation based on bit-rate parameters in the SMIL file. Either way, you need just one link on your Web page and your encoded files do not need to conform to any RealSystem naming conventions.

Additional Information

See "Supporting Multiple Bandwidth Connections" on page 13 for an overview.

RAM Files Created Automatically

RealSystem still uses RAM files (extensions `.ram` and `.rpm`) to launch presentations, but RealServer's RAMGEN feature can generate these files automatically. In your Web page, you link to the SMIL file and include in the URL special parameters that cause RealServer to generate the RAM file and download it to the Web browser.

Additional Information

See "Linking your Web Page to your Presentation" on page 75.

Enhanced Protocol Support

RTSP Protocol Now Used

Because it still supports the PNA protocol, RealServer 6.0 is backwards compatible with RealSystem 3.0 through 5.0. But it introduces as its primary protocol the RealTime Streaming Protocol (RTSP), an open, standards-based protocol for multimedia streaming. Because of this, URLs that point to media on RealServer now begin with `rtsp://`.

RealSystem Interoperates with RTP-Based Servers and Clients

When communicating with RealPlayer 6.0, RealServer 6.0 uses RealTime Streaming Protocol (RTSP) as its control protocol and RealNetworks' proprietary RDP as its packet protocol. But because RealSystem supports international standards for streaming media, RealServer and RealPlayer interoperate with RTP-based media servers and clients. The following table lists the protocols used with different mixes of servers and clients.

Server	Client	Control Protocol	Packet Protocol
RealServer 6.0	RealPlayer 6.0	RTSP	RDP
RealServer 6.0	RTP-based client	RTSP	RTP
RTP-based server	RealPlayer 6.0	RTSP	RTP
RealServer 6.0	RealPlayer 3.0 to 5.0	PNA	PNA
RealServer 3.0 to 5.0	RealPlayer 6.0	PNA	PNA

Compatibility with Previous Releases

RealSystem is fully compatible with RealSystem 3.0 through 5.0:

- You do not need to modify a presentation created for an earlier version of RealSystem. Users with the latest version of RealPlayer can still view your presentation, whether or not the RealServer used to stream the presentation has been upgraded to version 6.0.

- You can make some RealSystem G2 presentations playable with RealPlayer 3.0 through 5.0. If you choose not to do so, users with older versions of RealPlayer are asked to upgrade when they try to view the presentation.
- To include new features such as RealText in an existing presentation, you need to update the presentation. This includes creating a SMIL file and changing the URL in your Web page. You also need to make sure that your RealServer has been upgraded to the latest version.

Chapter 2: Creating a RealMedia Presentation

RealSystem gives you the power to create compelling, complex multimedia presentations streamed over a network. It includes RealServer, the most advanced streaming media server available, and RealPlayer, the world's most popular desktop application for playing streaming media files.

RealSystem *streams* multimedia presentations over a network. This lets the presentation start playing back shortly after the computer begins to receive data. In contrast, a Web server *downloads* files, and the computer must receive the files in their entirety before starting playback. A downloaded video clip, for example, may not begin playback for several minutes. A streamed video clip, in contrast, begins to play within seconds.

Choosing Media to Stream

RealMedia gives you many choices for combining media into a streaming presentation. In choosing which formats to stream, you need to take a careful look at bandwidth as described in the next section. This helps you decide, for example, what size to make a streaming video.

Audio

Chapter 3 beginning on page 21 discusses the audio formats you can stream:

- RealAudio
- AU
- WAV

Video

Chapter 4 beginning on page 29 describes the video formats you can stream:

- RealVideo

- AVI
- Vivo

Animation

RealFlash, which pairs Macromedia Flash animation with a RealAudio soundtrack, lets you stream animated presentations over a network. See Chapter 5 beginning on page 33 for details.

Images

RealMedia lets you stream any JPEG image along with video or audio. In addition, you can use RealPix to create eye-catching slide shows with special effects such as dissolves and zooms. For more information, download *RealPix Content Creation Guide* from **http://www.real.com**.

Text

RealMedia includes RealText, streaming text you can format and display at specific times within a presentation. You can use RealText to add subtitles to a video, for example, or lay out text from a live source to create a real-time stock ticker. For details, see *RealText Content Creation Guide*, available at **http://www.real.com**.

Additional Media Types

RealSystem easily extends to stream nearly any type of file or live event. Check **http://www.real.com** for the availability of plug-ins that let RealSystem stream additional video and audio formats, as well as exciting new types of media.

Targeting Bandwidth

A network connection's bandwidth is a crucial factor in developing a streaming multimedia presentation. To let Web users with 28.8 Kbps modems view your presentation, for example, you have to develop a presentation that requires less than 28.8 Kilobits of data per second.

As the first step in developing your presentation, target a bandwidth as described below and create content with that connection speed in mind. This helps ensure that the presentation streams smoothly. If you don't hit your bandwidth mark, the presentation may pause intermittently as RealPlayer waits for the necessary data to arrive.

What is Bandwidth?

Total bandwidth is the upper limit on how much data can pass through a network connection per second. Internet bandwidth is described in Kilobits per second (Kbps). A 28.8 Kbps modem, for example, can receive data at any speed up to 28.8 Kbps. Bandwidth is analogous to a speed limit, such as 60 m.p.h. A

presentation's bit rate is analogous to car speed. Based on variables such as weather and traffic, a car may be able to travel only 30 m.p.h. Due to network congestion and server load, a 28.8 Kbps modem may receive 11 Kbps of data one minute, 25 Kbps of data another.

When you drive on a highway, you have no control over weather and traffic that makes you slow down. Under good conditions, though, you can observe the speed limit. Likewise with your presentation, you have no control over server load and network congestion when a Web user views your presentation. You can, however, ensure that your presentation does not exceed the user's bandwidth. On the highway, breaking the speed limit gets you a ticket. On the Internet, exceeding bandwidth stalls your presentation.

For example, a 28.8 Kbps connection can still play a presentation that requires a 56 Kbps stream. But the modem takes around two seconds to receive the data that RealPlayer has to play every second. In other words, data has to be displayed at a rate faster than which it comes in over the modem. Consequently, RealPlayer does not begin playback until it receives and stores (“buffers”) enough data to play the presentation without halting. For a long presentation, this may take a few minutes. Viewers are not likely to wait for such a slow starting presentation.

Designing content suitable for viewers' available bandwidth is crucial to delivering a compelling multimedia presentation. Because most Internet users have 28.8 Kbps modems, content available to the public should target that bandwidth. If your presentation is for high-speed intranet use only, you may be able to target a higher minimum bandwidth. Delivering content suited for low bandwidth ensures that your presentation flows smoothly for all viewers, helping you reach the largest audience possible.

Choosing a Target Bandwidth

The target bandwidth of a RealMedia presentation is the maximum bandwidth available for a particular connection, such as 28.8 Kbps. The presentation's total bit rate must be at or below the target bit rate. The total bit rate comprises two main parts:

- Cumulative bit rate consumed by all streaming files (the presentation bit rate).
- 25% of target bit rate for overhead (noise, data loss, and packet overhead). This is an approximation. Overhead can vary depending on the type of connection and general network conditions. A 56 Kbps modem typically requires more overhead than a 56 Kbps ISDN connection, for example.

If your target bit rate is 28.8 Kbps, for example, take 75% of that rate as the bandwidth available for your streaming files. For a 28.8 Kbps connection, you have approximately 20 Kbps total for your presentation. The following table

lists the recommended maximum presentation bit rate for streaming files over different network connections.

Bit Rates Available for Streaming Files

Target Connection Speed	Suggested Maximum Bit Rate for Streaming Files
14.4 Kbps modem	10 Kbps
28.8 Kbps modem	20 Kbps
56.0 Kbps modem	34 Kbps
56.0 Kbps ISDN	45 Kbps
112 Kbps dual ISDN	80 Kbps

Developing a Bandwidth Strategy

Once you know the bit rate available for your streaming files, you can begin to develop your bandwidth strategy. If you want to stream just one file for your presentation, your strategy is straightforward. Suppose you want to create an audio file that Web users with 14.4 Kbps modems can play. You can simply create a RealAudio file that consumes 8 Kbps of bandwidth. Then anyone with a 14.4 Kbps or higher connection can listen to your presentation.

An exciting part of RealMedia, though, is that it allows you to put together different types of data in one presentation, such as slide shows with audio voiceovers, or video with scrolling subtitles. When multiple files play together in your presentation, you need to consider how much presentation bandwidth to allot to each file. This is also true if you have a single file that has multiple streams, such as a video that contains a visual track and an audio track.

Suppose you want to stream a RealVideo clip at 28.8 Kbps. How much bandwidth should you give to the visual track and to the audio track? The answer depends on the content. Because music has a greater frequency range than voice, a music video requires more audio data than a “talking heads” interview. Hence a soundtrack with music consumes more bandwidth than one that uses just speech.

The more you increase the audio track’s bandwidth, however, the more you have to decrease the visual track’s bandwidth. If you start with a huge video source file, your RealVideo encoding tool may discard a lot of the source data to make the encoded RealVideo data fit a certain bandwidth. Although the RealVideo file will be playable, you may not like the results. Motion might appear too jerky, for example, or fast-moving images might not resolve visually.

The point here is that you always need to think about bandwidth as you create your streaming media content. Your bandwidth target greatly affects how you create content. If you know you’ll have only a small bandwidth for video, for

example, you can optimize the visual content to display in a small window at a slow frame rate. You may need to jettison panoramic and fast action shots that won't fare well under these constraints.

This manual provides tips for creating content with bandwidth targets in mind. With some practice, you will quickly learn how to balance bandwidth requirements with presentation quality.

Supporting Multiple Bandwidth Connections

As described above, you can create a RealAudio file consuming 8 Kbps of bandwidth that anyone with a 14.4 Kbps or higher connection can play. This file will have good quality sound, but the same source file encoded for 16 Kbps will have better sound. Encoded at 32 Kbps, the file will have even greater frequency response and dynamic range.

To provide good content for users with slower connections and great content for those with faster connections, you can use two methods, and even mix them depending on your needs. With the first method, you create a single file that targets different bandwidths. In the second method, you create separate files for each bandwidth target and let RealPlayer choose which set of files to play. Either way, you add to your Web page just one link for all visitors. You don't need separate links for modem and ISDN connections, for example.

Encoding Single Files for Multiple Bandwidths

With RealAudio and RealVideo, you can encode different versions of a source file for different bandwidths. For example, you can encode a music file in RealAudio for 28.8 Kbps modems, 56 Kbps modems, and 112 Kbps dual ISDN connections. In your Web page, you link to this single file. When a user clicks the link to play the file, RealPlayer communicates its available bandwidth to RealServer, which then chooses the encoding to use.

RealServer and RealPlayer can even adjust this choice to compensate for network conditions. If a fast connection becomes bogged down because of network traffic, RealServer seamlessly switches to a lower bandwidth encoding to prevent the presentation from stalling. When the network congestion clears, RealServer switches back to the higher bandwidth encoding.

Letting RealPlayer Choose between Multiple Files

If your presentation uses file types other than RealAudio or RealVideo, you can create multiple versions of the files for different bandwidths. When you assemble your final presentation, you use the SMIL file to designate a bandwidth connection for each of the different file groups. When a user clicks your Web page link, RealPlayer receives the SMIL file and chooses which group to play based on its own connection speed.

Because each connection speed uses a different set of files, RealServer cannot switch between the different encodings as it can with a single, multiply encoded file. RealServer employs other techniques, however, to compensate for network congestion. Its advanced stream thinning capabilities let it drop low priority data to temporarily decrease the presentation bandwidth. When the congestion clears up, it continues to stream all the presentation data.

Additional Information

For an overview of SMIL, see "Writing a SMIL File" on page 17. "Setting Bandwidth Choices" on page 52 explains how to use a SMIL file to designate different bandwidth groups.

Working with Source Files

After you choose your media types and bandwidth target, you gather and edit your source material. The quality of your source and tools greatly affects the outcome of your streaming presentation. For example, a streaming soundtrack reflects the quality of the microphones used to capture the audio, as well as the sophistication of the hardware and software used to digitize and edit the sound file. Always use high quality source materials and good editing tools for your source files.

Additional Information

See Chapters 3 and 4 for tips on capturing high quality audio and video.

Synchronizing Files to a Timeline

Streaming multimedia presentations follow a timeline. In a two-minute video, for instance, each frame corresponds to a specific point in a two-minute timeline. The video's soundtrack is typically two minutes long as well. The first second of audio meshes with the first second of the video, and so on through both tracks' timelines. Video production software lets you coordinate the visual and audio tracks to a single timeline. You then create one file that contains these two internally coordinated tracks.

If you produce separate files through different software programs, however, you need to be aware of how timelines relate. Suppose you create a video file that has just a visual track. Through another software program you then produce a soundtrack. You need to note in this case how the clips' timelines relate. Should they start playing together? Or should one clip be delayed? When you assemble the presentation, you can use the SMIL file to specify playback times for each file.

Additional Information

For an overview of SMIL, see “Writing a SMIL File” on page 17.

Encoding and Compressing Source Files

Editing tools typically save data in a proprietary format, letting you export the data to a different format as needed. For example, most image editing programs let you export a graphic as a JPEG or GIF. If your editing program does not export the source file to the streaming format you want, you need an additional encoding tool to convert the format.

In addition, you typically need to compress the streaming file. File formats such as RealVideo are always compressed, so the conversion process involves compression. Other file formats, such as WAV, are uncompressed until you compress them. So with these file formats, compression involves another step after converting the source file to a streaming format.

Additional Information

The chapters on producing the various file types explain the compression you need to use for the each file type.

RealNetworks Encoding Tools

RealNetworks encoding tools convert files to the RealAudio or RealVideo format. RealNetworks provides free tools for converting popular sound and video formats. It also sells advanced encoding tools that help you synchronize timelines, create HTML pages, and transfer pages to a server. In addition, plug-ins for popular programs such as Adobe Premiere and Microsoft PowerPoint let you save presentations directly as RealVideo. RealNetworks’ encoding tools are easy to use, and let you quickly build the presentation you want.

Additional Information

Check **http://www.real.com** for the tool that’s right for you.

How Compression Works

Media source files are often too large to stream, even at high bandwidths. Compression reduces the data in the file, making the file smaller. To illustrate how this works, imagine your computer screen as a rectangular grid of small squares, 640 squares wide and 480 squares high. When the computer paints the screen, it acts like a workman who puts a tile in each square to create a mosaic. If you are telling the workman how to fill in the mosaic, you could say:

1. “Put a red tile in column 1, row 1.”
2. “Put a red tile in column 2, row 1.”

3. "Put a red tile in column 3, row 1."
4. And so on until row 1 is tiled. Then start all over for row 2, row 3, and so forth.

Suppose that the entire top row should be red tiles. You can convey easily this information without all the repetition:

1. "Put red tiles in all columns of row 1."

This drastically reduces the amount of information needed to get the job done. File compression works much the same way. Although the many kinds of compression technology are diverse in their approaches and complexities, they all use some means to cut down the amount of data stored in a file while still retaining good playback quality.

Notes on Converting and Compressing Files

Keep the following in mind when converting formats or compressing files:

- Always keep a copy of the source file. If you need to change your presentation, you typically have to change the source file, then redo the conversion/compression.
- File compression affects how much bandwidth the file consumes. So you need to consider the compression you'll use when you target bandwidth.
- "Lossy" compression types such as JPEG, RealAudio, or RealVideo discard some source file data. So using too high of a compression rate can degrade quality to unacceptable levels. You may need to experiment when using these compression types to find a good balance between file size and presentation quality.
- Don't compress a file more than once. The file size will not decrease further, and the file may become unplayable.
- The quality and content of a video can affect the compressed file size. For more on this, see "Shooting a Video" on page 29.

Using a Container Format

When you have converted your presentation files to their streaming formats, you can combine the streaming files into a single file, called a container file. Popular container formats include:

- Advanced Streaming Format (extension `.asf`)
- RealMedia File Format (extension `.rmf`)

RealSystem can stream either of these container formats. However, using a container format is not necessary. Keeping your files and in their native

streaming formats and putting the presentation together with a SMIL file as described below gives you greater flexibility.

Writing a SMIL File

With your files in their streaming formats, you put the presentation together with SMIL. Pronounced “smile,” SMIL stands for Synchronized Multimedia Integration Language. A SMIL file is not necessary to stream just one file. But when you have multiple files, SMIL’s simple mark-up language specifies how and when the files play. Here are some of the many advantages of using SMIL:

- Avoid Additional Encoding for Container Formats

  Because your files stay separate, you avoid the additional step of merging them into a container file. To change your presentation, you edit the SMIL file rather than merging the files again into a different container file.

- Use Files in Different Locations

  Because the SMIL file lists a separate URL for each file, you can put together presentations using files in any locations. You can use a video file from one server, for example, and an audio file from another.

- Support Multiple Languages

  A SMIL file can list different language options for audio or text files. To create a video with sound tracks in different languages, for example, you produce one video file with no soundtrack, then create audio files in each language. Your Web page needs just one link to the SMIL file. When a visitor clicks that link, the visitor’s RealPlayer chooses the soundtrack to receive based on its language preference.

- Support Multiple Bandwidths

  The SMIL file can also list presentation choices for different bandwidths. RealPlayer then chooses which files to receive based on its available bandwidth. You can thereby support multiple connection speeds through a single hypertext link, rather than separate links for modem users, ISDN users, T1 users, and so on.

- Put Together Customized Presentations

  Because a SMIL file is a simple text file, you can generate it automatically for each visitor. You can therefore create different presentation parts, then assemble a customized SMIL file based on preferences recorded in the visitor’s browser.

- Time and Control the Presentation

  The SMIL file lets you easily control the presentation timeline. You can delay an audio track by 2.5 seconds, for example, without changing the encoded audio file.

- Lay Out the Presentation

  When your presentation includes multiple elements, such as two videos playing simultaneously, you can use SMIL layout tags to align the videos.

- Include Ads

  For commercial presentations, the SMIL file lets you insert ads into the presentation as needed.

Additional Information

Chapter 6 beginning on page 43 explains the SMIL file syntax. Chapter 8 beginning on page 73 explains how to set up ad rotation.

Putting Files on RealServer

When your presentation is ready to go, you move the streaming media files and SMIL file to RealServer for testing and delivery. To make your presentation accessible, you simply create a link to the SMIL file in your Web page. When a user clicks that link, RealPlayer launches and plays the presentation in its own window.

You can also play the presentation directly in your Web page. To do this, you use RealPlayer's Netscape plug-in or ActiveX Control, adding mark-up to your Web page to specify how the presentation displays and what RealPlayer controls appear on the page.

Additional Information

See Chapter 9 beginning on page 78 for information on moving files to RealServer. For more on playing back a presentation in a Web page, see Chapter 7 beginning on page 59.

Note

If you use an Internet Service Provider (ISP), make sure your provider has the latest version of RealServer available.

Viewing your Presentation

With RealPlayer and the browser of your choice installed, you simply click the presentation link in your Web page. RealPlayer buffers presentation files for a few seconds, then begins to play the presentation back in its own window or your browser. Free RealPlayer downloads are available from RealNetworks at **http://www.real.com**.

Additional Information

For advice on testing, see "Testing your Presentation" on page 81.

Downloading RealPlayer Plug-Ins

RealPlayer can play virtually any streaming file because of its plug-in technology. RealPlayer plug-ins function like Web browser plug-ins. When RealPlayer receives a streaming RealVideo movie, for example, it uses its RealVideo plug-in to play the streaming data on your computer screen. If RealPlayer doesn't have a plug-in needed to play a certain streaming file, it downloads that plug-in from the Internet.

Plug-in downloading lets you confidently develop presentations using the latest streaming file types available for RealSystem. If visitors to your Web page don't have a plug-in needed to play your presentation, they can quickly download it and view your presentation. Because RealPlayer is the world's most popular application for viewing streaming media, you can be sure that your RealMedia presentation can reach the widest audience possible.

Additional Information

For more information about developing RealPlayer plug-ins or building RealPlayer capabilities into another application, visit **http://www.real.com**.

Chapter 3: Producing Audio

RealSystem can stream many types of audio in addition to the RealAudio format. Not all audio files stream equally well, however. This chapter describes the types of audio files you can stream, explaining how to prepare or encode your files. It also provides tips for capturing high quality audio source.

Recording High Quality Audio

The following are tips for creating high quality audio source files. Although geared for RealAudio, these guidelines will help you no matter which audio format you stream.

Additional Information

For pointers on video, see “Producing High Quality Video” on page 29.

Use Professional Recording Equipment

Every piece of equipment in the audio chain, from the microphone, to the sound card, to the sound editing software, affects sound quality. If you intend to provide commercial audio content, invest in professional audio equipment and software. Poor quality equipment can add hiss and distortion, degrading sound clarity.

Use Quality Source Files

If you are not recording your own sound, be sure to use high-quality audio source files. Use sources from CD or DAT, for example.

Digitize Sound before Encoding

If you are not broadcasting live, capture or “digitize” the sound to a supported file format such as a WAV, QuickTime, or AIFF whenever possible. Digitizing the sound before encoding the file allows you to use a sound editor to adjust the signal amplitude to maximize the available dynamic range. If you do not adjust the signal, the resulting streamed files may sound flat.

Minimize Source File Size

Keep the audio source file as small as possible. This makes it easier to encode the file in a streaming format. Cut any unnecessarily long silences from the beginning or end of the file to conserve space.

Set Input Levels Correctly

Setting correct input levels is crucial. All audio equipment has a signal-to-noise ratio, a ratio between the loudest possible sound the equipment can reproduce without distortion and its inherent noise. To work with the loudest input sounds possible, set the input level to use the full range of available amplitude without distortion. This distortion is known as "clipping," and is audible as a high frequency crackling noise.

When digitizing with a sound card, do several test runs and adjust levels on the mixer page of your sound card utilities so the input approaches but does not exceed the maximum. Most mixer pages graphically display input. Make sure there are no peaks above maximum. Be conservative, though. You never know when someone will get excited and speak much louder, or when a crowd at a sports event will roar.

Sound files that do not use the full amplitude range will produce low quality streaming files. If the amplitude range is too low, use your sound editor's **Increase Amplitude** or **Increase Volume** command to adjust the range before encoding the file. Most sound editors have a **Normalize** function that maximizes levels automatically. However, you get better quality if you set the levels correctly when recording.

Eliminate DC Offset

Eliminate DC offset either while recording content or later with an sound editor. This removes low frequency noise.

Equalize Frequencies

Equalization (EQ) changes the tone of the incoming signal by "boosting" (turning up) or "cutting" (turning down) certain frequencies. Using EQ, you can emphasize frequencies you want and cut frequencies that contain noise or unwanted sound. In addition, EQ can compensate for RealAudio Codecs that do not have flat frequency responses (that is, Codecs for which certain frequencies are not as loud after encoding).

Normalize Audio Files

As the last step before encoding the file, normalize the source file to 95% of the maximum sound volume. This lets you feed your encoding tool the loudest distortion-free files possible. If your machine's normalization option does not let you specify a percentage, turn down the overall volume after you normalize by using your software's **Volume** or **Amplify** option.

Prevent Clipping

If your original audio file signal exceeds the acceptable amplitude range, the file may be "clipped." Clipping can give rise to clicks or pops on playback. If your source file contains a clipped signal, your streaming file may have high-frequency background noise or static. Lowering the input volume helps reduce clipping.

Prepare Volume Levels for Live Broadcasts

When broadcasting live audio, you have less opportunity to manipulate the input signal. Be sure that volume levels are prepared and tested before encoding live input.

Producing RealAudio

RealNetworks pioneered streaming audio with RealAudio, the first streaming media product for the Internet. Since its debut in 1995, RealAudio has become the standard for network audio, delivering stereo sound over 28.8 Kbps modems, with near-CD quality sound at ISDN and LAN speeds. RealAudio files use the file extension `.ra`.

Audio Input Formats

RealAudio is a compressed format suitable for streaming over low to high network speeds. Because RealAudio is compressed, you typically start with a sound file in a digitized, uncompressed format such as WAV or AIFF. You then create a RealAudio file from this source file through an encoding tool. Your encoding tool should be able to accept some or all of these input formats:

- Audio Interchange File (`.aif`)
- Audio (`.au`)
- QuickTime (`.mov`)
- Sound (`.snd`)
- Waveform (`.wav`)

Additional Information

See "RealNetworks Encoding Tools" on page 15 for more on RealAudio encoding tools. Refer to "Streaming Other Audio Formats" on page 27 for information on streaming digitized audio files without conversion to RealAudio.

Using RealAudio Codecs

RealAudio uses a "lossy" compression scheme that disregards parts of the audio source file to achieve a highly reduced file size. A RealAudio file encoded from

a WAV file, for example, is typically smaller than the WAV by a factor of ten or more. Although discarding audio information during encoding lowers the file's frequency response and dynamic range, carefully choosing Codecs minimizes the impact of compression.

An encoding tool uses a Codec to compress the original sound file and create a RealAudio file. RealPlayer uses the same Codec to decompress the streamed RealAudio file for playback. When you encode a RealAudio file, you choose a Codec (or series of Codecs) based on two criteria:

1. Bandwidth

   As "Targeting Bandwidth" on page 10 explains, you need to decide how much bandwidth each part of your presentation will consume. When you have a bandwidth target for your audio component, you can choose a Codec that encodes RealAudio at or below that target.

2. Audio Content

   RealNetworks provides different Codecs for music and spoken voice. Voice Codecs, for example, focus on the standard frequency range of the human voice. Music Codecs have broader frequency ranges to capture more of the high and low frequencies.

Choosing a RealAudio Codec

When you encode your audio file with a RealNetworks encoding tool, the tool selects the Codec or Codecs automatically. You simply set your bandwidth targets and some other parameters, such as whether you want earlier versions of RealPlayer to be able to play your presentation. The tool then selects the best Codec or Codecs to use based on your input.

Additional Information

Refer to the manual or online help for your RealNetworks encoding tool for step-by-step instructions on how to encode a RealAudio file.

The following table lists all RealAudio Codecs. It is provided as a general reference. Check your RealAudio encoding tool for the Codecs it supports. See below for information about interpreting the table.

Audio Codec	6	5	4	3	2	1	Sample Rate	Resp.	Comments
Low Bandwidth Audio Codecs									
5 Kbps Voice	X	X	–	–	–	–	8, 16, or 32 kHz	4 kHz	
6.5 Kbps Voice	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	
8 Kbps Voice	X	X	X	X	X	X	8, 16, or 32 kHz	4 kHz	Superseded by 8.5Kbps Voice Codec.

(Table Page 1 of 2)

Audio Codec	6	5	4	3	2	1	Sample Rate	Resp.	Comments
8 Kbps Music–G2	X	–	–	–	–	–	8, 16, or 32 kHz	4 kHz	Generation 2 Codec. Use with multiply encoded files.
8 Kbps Music	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	DolbyNet Codec.
8.5 Kbps Voice	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	
11 Kbps Music–G2	X	–	–	–	–	–	11.025, 22.05, or 44.1 kHz	5 kHz	Generation 2 Codec. Use with multiply encoded files.
12 Kbps Music	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	DolbyNet Codec.
Medium Bandwidth Audio Codecs									
15.2 Kbps Voice–RealAudio 2.0 Mono	X	X	X	X	X	–	8, 16, or 32 kHz	4 kHz	Superseded by 16 Kbps Voice Codec.
16 Kbps Voice–Mono Wideband	X	X	–	–	–	–	16 or 32 kHz	8 kHz	Highest bit rate Codec for voice.
16 Kbps Music–G2 Mono	X	–	–	–	–	–	22.05 or 44.1 kHz	8 kHz	Generation 2 Codec. Use with multiply encoded files.
16 Kbps Music–Mono Low Response	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	DolbyNet Codec.
16 Kbps Music–Mono Medium Response	X	X	X	–	–	–	11.025, 22.05, or 44.1 kHz	4.7 kHz	Suitable for pop/rock music. DolbyNet Codec.
16 Kbps Music–Mono High Response	X	X	X	–	–	–	11.025, 22.05, or 44.1 kHz	5.5 kHz	Suitable for classical music. DolbyNet Codec.
20 Kbps Music–G2 Mono	X	–	–	–	–	–	22.05 or 44.1 kHz	10 kHz	Generation 2 Codec. Use with multiply encoded files.
20 Kbps Music–Stereo	X	X	X	X	–	–	8, 16, or 32 kHz	4 kHz	DolbyNet Codec.
High Bandwidth Audio Codecs									
32 Kbps Music–G2 Mono	X	–	–	–	–	–	44.1 kHz	16 kHz	Generation 2 Codec. Use with multiply encoded files.
32 Kbps Music–Mono	X	X	X	–	–	–	11.025, 22.05, or 44.1 kHz	8 kHz	DolbyNet Codec.
32 Kbps Music–Stereo	X	X	X	–	–	–	8, 16, or 32 kHz	5.5 kHz	DolbyNet Codec.
40 Kbps Music–Mono	X	X	X	X	–	–	11.025, 22.05, or 44.1 kHz	11 kHz	DolbyNet Codec.
40 Kbps Music–Stereo	X	X	X	X	–	–	8, 16, or 32 kHz	8 kHz	DolbyNet Codec.
80 Kbps Music–Mono	X	X	X	X	–	–	11.025, 22.05, or 44.1 kHz	20 kHz	DolbyNet Codec.
80 Kbps Music–Stereo	X	X	X	X	–	–	8, 16, or 32 kHz	16 kHz	DolbyNet Codec.

(Table Page 2 of 2)

The preceding table includes the following information:

- 6, 5, 4, 3, 2, 1

  An "X" in these columns indicates that a file encoded with this Codec can be played by RealPlayer 6.0, 5.0, 4.0, and so on.

- Sample Rate

  You can use any sampling rate in your audio source file. However, the suggested sampling rates in the table above ensure that the audio stays synchronized with other media and prevents pitch shifting in audio resampling. The highest sampling rate provides the fullest sound.

- Resp.

  This column lists the Codec's frequency response in kHz.

Notes on RealAudio Encoding

The following sections give tips for encoding RealAudio files.

Multiple Encoding in a Single File

You can create a single file encoded for multiple bandwidths only with the Codecs introduced in RealSystem G2. In the preceding table, these Codecs are marked as playable only by RealPlayer 6.0. To support multiple bandwidths when encoding with other Codecs, you must encode a separate file for each Codec. You then use a SMIL file to specify bandwidth choices.

Additional Information

For more on bandwidth selection through SMIL, see "Setting Bandwidth Choices" on page 52.

Backwards Compatibility with Earlier Versions of RealPlayer

When you use a RealNetworks encoding tool to encode a file for multiple bandwidths, you can specify backwards compatibility with earlier versions of RealPlayer. The tool encodes the file for your selected bandwidths with the G2 Codecs. It also includes in the file an encoding at your lowest bandwidth selection that uses an older Codec.

For example, you can encode a single file at 8, 16, and 32 Kbps using the G2 Codecs. In the encoding tool, you can choose backwards compatibility to create an additional 8 Kbps stream with an older Codec. Depending on its connection speed, RealPlayer 6.0 receives the 8, 16, or 32 Kbps G2 stream. Earlier versions of RealPlayer receive the 8 Kbps stream encoded with the older Codec regardless of their connection speeds.

Streaming Other Audio Formats

RealSystem can stream audio formats other than RealAudio. Because these formats may not be as highly compressed as RealAudio, they may not be good for low bandwidth connections. The following sections give you guidelines for preparing files in these formats. This information will help you decide if you should stream from the native format or convert the file to RealAudio.

Note

The following sections will be added later.

AU

WAV

Chapter 4: Producing Video

This chapter covers the types of video that RealSystem can stream. It will help you decide whether to convert your video source file to RealVideo or stream from another video format. It also provides tips for capturing high quality video.

Producing High Quality Video

Because video loses image quality when compressed, you need to start with high quality video source. This section gives you tips on which video source formats to use, as well as pointers on staging, shooting, and digitizing the video. Although geared for RealVideo, these guidelines will help you no matter what video format you choose.

Additional Information

For pointers on producing audio, see “Recording High Quality Audio” on page 21.

Shooting a Video

Observe the points below if you intend to shoot a new video rather than use existing content. These guidelines will help you produce high-quality video with a low compressed file size.

Stage According to the Video’s Final Size

To reach Web users with 28.8 Kbps modems, you need to produce a small video, such as 176 x 144 pixels. If you show an instructor pointing to a chart, for example, you may need to zoom in on the chart to make the text legible.

Minimize Scene Changes and Movement

The less that changes from frame to frame, the more the video file will compress. Most compression technologies reuse existing data when frames are similar. So a video with relatively stationary subjects (“talking heads”) will compress more than a music video

with rapid scene changes and a lot of movement. You can do the following to cut down on unnecessary movement:

- Use a mounted rather than hand-held camera. This greatly reduces the movement you inadvertently introduce into the scene when recording.
- Don't have an object that moves rapidly fill the entire frame. Keep in mind, though, that your streaming video may be a few square inches in size, so you don't want to pull the camera back too far.

Of course, you don't want to eliminate all dynamic elements! When you do include rapid movement, give enough time for objects to resolve. Because of low frame rates and high compression, objects coming to a rest may appear blurry at first. If you have a dialog box popping up on a computer screen, for example, show that box stationary for a few seconds so that the image resolves.

Use Light, Uniform Colors and Good Lighting

Bright lighting at a constant exposure keeps the foreground detail crisp and cuts down the compressed file size. Use uniform, light colors in backgrounds and clothing. Dark colors and complex textures such as paislies and stripes add to the file size. They can also degrade the video with unwanted visual effects.

Digitizing Video

Keep the following points in mind when digitizing a video source.

Use a High Quality Source Format

Whether you shoot a video yourself or digitize existing material, it's important to use a high-quality video format. The following are common video formats in order of descending quality:

1. Betacam-sp, also known simply as Beta. This format is common among video production professionals.
2. Satellite television services (for example, Direct TV), which can produce video on the level of Beta.
3. Laserdisc.
4. S-VHS or Super-VHS.
5. VHS.

Use S-video

Video playback devices commonly have two output types, S-video and composite. Use S-video, which generally produces better results.

Digitize to an Uncompressed Format

If possible, digitize in an uncompressed format. This gives you greater flexibility when editing the video. It produces larger source files, however.

Producing RealVideo

RealNetworks introduced RealVideo with RealSystem 4.0. A RealVideo file uses the file extension `.rm` and typically includes a soundtrack encoded in RealAudio.

Additional Information

For more on RealAudio, see "Producing RealAudio" on page 23.

Video Input Formats

RealVideo is a compressed format, so you typically start with a digitized, uncompressed format such as AVI. You then convert this file to RealVideo for streaming. Check your encoding tool's supported input formats, which should be some or all of the following:

- AVI (`.avi`)

  You can typically convert compressed or uncompressed AVI to RealVideo. RealNetworks recommends using the uncompressed format whenever possible.

- QuickTime (`.mov`)

Additional Information

See "RealNetworks Encoding Tools" on page 15 for more on RealVideo encoding tools available from RealNetworks.

Using RealVideo Codecs

Like RealAudio, RealVideo uses a "lossy" compression scheme that discards parts of the source file during encoding. When you encode RealVideo, you simply choose an encoded video bandwidth from the encoding tool's list of supported bandwidths. You also choose a RealAudio Codec for the audio track.

Additional Information

See "Using RealAudio Codecs" on page 23 for general information on Codecs.

Streaming Other Video Formats

RealSystem can stream other video file formats in addition to RealVideo. Because these file formats may not be as highly compressed as RealVideo, they may not work well with low bandwidth connections. The following sections give you guidelines for steaming these other video formats. This information will help you decide if you should stream the native format or convert the file to RealVideo.

Note

The following sections will be added later.

AVI

Vivo

Chapter 5: Producing Animation

This chapter tell how to create RealFlash presentations optimized for different bandwidths. It also provides tips for creating streaming Flash content. For exciting examples of streaming animation, visit the RealFlash showcase at **http://www.real.com/showcase/animation/index.html**.

Benefits of Using RealFlash

RealFlash makes it easy to put animation on the Web. It combines the power of Macromedia Flash with the clarity of RealNetwork's RealAudio. The results are visually arresting animations with superb sound. RealFlash is well-suited for linear presentations that have continuous audio and images synchronized along a timeline, including:

- Full-length, television-like cartoons for entertainment and education
- Internet or intranet demonstrations, training courses, and product overviews
- Product advertisements
- Movie trailers
- Karaoke

Tools for Creating RealFlash

You need the following tools to create and stream RealFlash:

- Macromedia Flash 2.0

  You use Flash to create animations and import sound, synchronizing the two to a single timeline. This chapter provides tips for optimizing streaming animations, but you need to refer to the Flash documentation from Macromedia for information about using Flash.

- Sound capture and editing tools

  You should use professional quality hardware and software to capture and process the sound file encoded in the RealAudio format.

Additional Information

For more on audio production, see Chapter 3 beginning on page 21.

- RealNetworks Production Tools

  To encode your sound file as RealAudio, use a RealNetworks tool available from **http://www.real.com**.

- RealFlash Optimization Kit

  This kit contains the RealFlash Bit Rate Calculation Spreadsheet and the RealFlash Bandwidth Tuner. These tools help you create and optimize your presentation. You will need Microsoft Word and Microsoft Excel to use the spreadsheet.

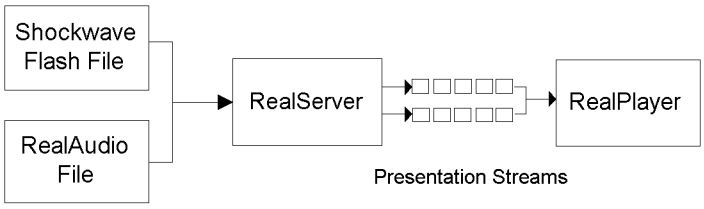
- RealServer and RealPlayer

  RealServer is required to deliver your RealFlash presentation. Web users view your presentation through RealPlayer. Free RealPlayer downloads are available from RealNetworks at **http://www.real.com**.

Preparing a RealFlash Presentation

A RealFlash presentation consists of two separate files streamed together, a Flash animation file and a RealAudio soundtrack. To create these components, you develop animation in Flash and synchronize it with an imported sound file, such as a WAV or AIFF file. You then export a Shockwave Flash file that contains the animation and generate a RealAudio file from the soundtrack. RealServer streams the presentation to RealPlayer, ensuring that animation and sound stay synchronized.

RealFlash Presentations Consist of Flash and RealAudio



Choosing a Target Bandwidth

When you begin to develop your RealFlash presentation, target an audience connection speed and create content with that bandwidth in mind. This helps ensure that both the Flash animation and the RealAudio file stream smoothly. If your target bit rate is 28.8 Kbps, for example, you have approximately 20 Kbps of bandwidth to divide between the RealAudio soundtrack stream and the Flash animation stream.

Additional Information

For an overview of bandwidth considerations, see "Targeting Bandwidth" on page 10.

The good news is that designing Realflash content for low bandwidths does not diminish the quality of your animation. RealFlash transmits vector information that the viewer's machine then renders. So unlike with bitmap animations, the quality of Flash animation depends more on the machine's CPU and graphics capabilities than the amount of data downloaded. A well-designed 28.8 Kbps RealFlash animation can have the same visual impact as an animation requiring a significantly higher connection speed.

Dividing Bandwidth Between Flash and RealAudio

Once you have determined the combined bit rate for Flash and RealAudio, you need to divide the rate between the Flash and RealAudio components. Your animation usually determines this division because it typically consumes more bandwidth. Although you may not have a final bandwidth figure until you create, export, and tune your animation, you should start with a target estimate. The table below lists commonly used RealAudio and Flash bit rate combinations for a 28.8 Kbps connection.

Common Bandwidth Divisions between RealAudio and Flash at 20Kbps

Presentation Type	RealAudio	Flash
Emphasis on animation with good quality spoken soundtrack	5 Kbps	15 Kbps
Emphasis on animation with high quality spoken soundtrack	6.5 Kbps	13.5 Kbps
Emphasis on animation with very high quality spoken soundtrack	8.5 Kbps	11.5 Kbps
Emphasis on animation with good quality music soundtrack	8 Kbps	12 Kbps
Emphasis on high quality music soundtrack with animation	12 Kbps	8 Kbps

If sound quality takes precedence, start by selecting the RealAudio Codec that supplies high quality audio while leaving enough bandwidth for good quality animation. Because RealAudio bandwidth consumption is flat, a soundtrack using an 8.5 Kbps Codec, for example, will consistently consume 8.5 Kbps of bandwidth. The remainder of the combined bandwidth is available for Flash.

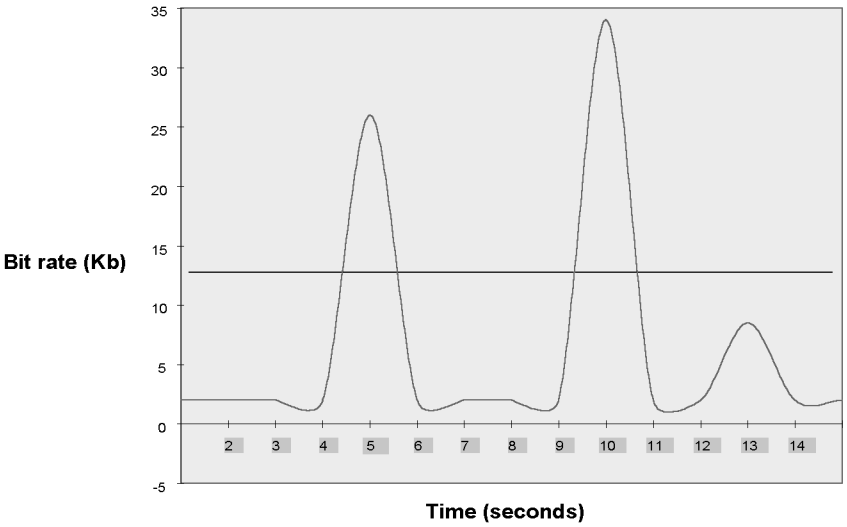
Additional Information

See "Choosing a RealAudio Codec" on page 24 for a list of Codecs.

Maximizing Flash Efficiency

Unlike RealAudio, Flash does not consume bandwidth at an even rate. This is the nature of vector-based animation. At the start of a scene, for example, groups and symbols used in the scene are downloaded. This requires a lot of data transfer. After that, only "lightweight" instructions for manipulating the groups and symbols are needed. This process results in bandwidth consumption like that shown in the following figure.

Flash Sample Bit Rate Requirement



This graph shows a sample Flash presentation consuming an average of 12 Kbps of bandwidth. Around five and ten seconds into the presentation, the bandwidth requirement spikes because the presentation needs more than 12 Kb of data. These spikes typically correspond to scene changes or the introduction of new objects in a key frame. RealPlayer responds by buffering the data as it comes in, potentially delaying playback until all necessary data has arrived.

As you create your Flash animation, you need to minimize the spikes that may cause RealPlayer to halt the presentation while it buffers data. There are two ways you can do this:

1. As you create your animation, minimize the overall bit rate requirement of the Flash stream by keeping the ratio of file size to presentation length as low as possible. This doesn't eliminate spikes, but it helps keep the spikes smaller. The guidelines below explain how to do this.
2. After you export your animation file, use the RealFlash Bit Rate Calculation Spreadsheet included with the RealFlash Optimization Kit to view frame-by-frame bandwidth needs. You can then modify the Flash source file or use the tuning utility to change the streaming file's bandwidth consumption.

Keeping Flash Files Small

The following are recommendations for keeping Flash file size down as you develop the animation:

- Reduce successive key frames.

  Excessive key frame changes increase bandwidth consumption. Minimize the number of key frames and simplify the objects within key frames.

- Use symbols instead of groups.

  Flash stores a symbol once and can refer to it repeatedly, each reference adding little to the file size. However, it stores a group definition each time the group is used. Using a group three times, for example, stores the same data in the file three times. Using symbols instead of groups can therefore reduce file size significantly.

- Simplify Flash elements.

  Simplify the elements drawn in or imported into Flash. Under **Modify>Curves**, use the **Smooth** and **Straighten** commands on lines and curves to strip away unneeded point and path information. This reduces the data stored for each element. Use **Optimize** to optimize the data reduction while maintaining acceptable screen appearance. Because screen resolution is lower than print resolution, you can eliminate minute details without compromising appearance.

- Adjust JPEG quality when exporting.

  When exporting **.fla** files to **.swf** files, set the JPEG quality to no greater than 50, possibly as low as 30.

Minimizing CPU Usage

Bandwidth is not the only consideration when optimizing Flash files. Flash's vector-based animation differs from raster or bitmap animation in that the user's machine must perform complex calculations to display the animation. Operations that require many calculations on top of the normal load may

adversely affect playback. The following are ways to reduce the RealFlash CPU requirements:

- Set a frame rate of seven (7) fps for 28.8 Kbps connections.

  Using a frame rate of 7 fps provides acceptably smooth motion without overburdening most processors.

- Optimize tweening.

  Tweening interpolates the motion between key frames. Interpolating multiple objects and color effects at the same time will adversely affect playback. Other actions related to tweening that slow down playback are changing large areas of the screen between frames and using gradient fills.

- Decrease the number and size of objects simultaneously moving on screen.

  The CPU must redraw areas where action occurs, thus consuming CPU cycles. To minimize this, localize tweening to a small portion of the screen so that the entire screen does not have to be redrawn. File size remains the same, but only one part of the screen is redrawn.

Using Interactive Commands

Although RealFlash is best suited for linear presentations, you can add interactivity through certain Shockwave commands. RealFlash maps these commands to RealPlayer functions. At the end of your RealFlash presentation, for example, you might have a graphic that says, “Click here to visit our home page.” The Shockwave **Get URL** command used with this graphic corresponds to a RealPlayer command that displays the URL in the browser window.

Shockwave and RealPlayer Interactive Commands

Shockwave Command	RealPlayer Mapping	Notes
Play	Play	(none)
Stop	Pause	Presentation pauses until action is performed or Play button pressed.
Goto	Seek	RealPlayer seeks to the designated frame and buffers the defined presentation preroll. Because seeking requires buffering, do not use Goto to advance from one scene to the next in a linear presentation.
Get URL	(internal)	Displays URL in browser window. Because the user has to return to RealFlash manually, use this only at the end of a presentation.

Creating a RealFlash Presentation

The following sections explain the steps for creating a RealFlash presentation. Refer to the manuals for the tools you use for step-by-step instructions on carrying out each task.

Importing an Audio Source

When you create your Flash animation, you import your audio source (WAV or AIFF) and synchronize it with the animation timeline, thereby creating a soundtrack. Flash provides different methods for incorporating sound into an animation. For RealFlash presentations, use the stream synchronization setting.

Exporting Shockwave Flash

You export your Flash animation to a Shockwave Flash (**.swf**) file for use with RealSystem. This creates a compressed version of the animation suitable for streaming. When you export the Shockwave Flash file, you disable the audio stream. You later export the soundtrack separately and convert it to RealAudio. Here are tips on exporting Shockwave Flash:

- Set JPEG quality between 30 and 50. This helps to keep the file size down.
- Click the **Generate Size Report** checkbox. This creates the Flash movie report you can use with the tuning spreadsheet. RealNetworks highly recommends that you use the movie report and spreadsheet to evaluate your RealFlash presentation for bandwidth efficiency.

The ratio of Shockwave file size to presentation length is a good indication of the overall bandwidth requirement. Convert the file size to Kilobits and divide by the number of seconds in the animation to get the average bandwidth. This number should be below your allowable bit rate for Flash. For example, to find the average bandwidth of a 325 Kilobyte file that plays for 3 minutes, multiply 325 by 8 to get 2624 Kilobits. Then divide by 180 seconds to get an average bandwidth of 14.6 Kbps.

Converting File Size to Kilobits

Using This Measurement	Do This to Get Kilobits
Megabytes	Multiply by 8192
Kilobytes	Multiply by 8
bytes	Divide by 128
bits	Divide by 1024

Tip

You can also find the average bit rate of a Shockwave Flash file by dropping it onto RealPlayer and observing the RealPlayer status bar.

Keep in mind that even a presentation with an acceptable average bandwidth may stall during playback because it contains bandwidth spikes. Use the spreadsheet to find out where spikes occur.

Tuning Shockwave Flash

After you generate a Shockwave Flash file and create the movie report, you can use the RealFlash Bit Rate Calculation Spreadsheet to examine the file's bandwidth consumption frame-by-frame. You can also use the RealFlash Bandwidth Tuner to view bandwidth statistics and adjust the file's streaming bit rate.

Additional Information

These tools are part of the RealFlash Optimization Kit available at **http://www.real.com**. See the tuner online help for instructions on using the spreadsheet and tuner.

Exporting Audio

After you have created and tuned your Shockwave Flash file, you export the Flash soundtrack and convert it to the RealAudio format with a RealNetworks encoding tool. In Flash, you export the movie as a Windows AVI or Macintosh QuickTime file, setting 32x21 as the height and width attributes to minimize disk space usage and file creation time.

After you save your RealFlash audio as an AVI or QuickTime movie, encode it in the RealAudio format with a RealNetworks tool, using the file extension `.rm`. Choose a Codec that fits your presentation's bandwidth and content requirements. Here are some guidelines for selecting a Codec:

- When animation is complex, use low bit rate Codecs targeted for voice.
- Use higher bit rate Codecs when emphasizing music or narration. The lowest bit rate for a music Codec is 8 Kbps.
- To ensure a high-quality visual presentation, you may need to increase the bit rate for a complex animation. This requires you to select a lower bit rate Codec to stay within the acceptable bandwidth range.

Additional Information

See “Choosing a RealAudio Codec” on page 24 for a list of Codecs.

Creating a SMIL File

When your Shockwave Flash and RealAudio files are complete, you create a SMIL file that lists the URLs for these files. Chapter 6 beginning on page 43 explains how to create the SMIL file. In its simplest form, the SMIL file specifies that the two files play in parallel:

```
<smil>
  <body>
    <par>
      <audio src="rtsp://realserver.company.com/sound.rm"/>
      <animation src="rtsp://realserver.company.com/cartoon.swf"/>
    </par>
  </body>
</smil>
```

Transferring Files to RealServer

When the presentation is ready, you move the Shockwave Flash, RealAudio, and SMIL files to their designated locations on RealServer. You then create a link in your Web page to the SMIL file. For instructions on how to do this, see Chapter 9 beginning on page 75.

Chapter 6: Assembling a Presentation with SMIL

When your multimedia presentation contains multiple clips–such as two videos played together–you use Synchronized Multimedia Integration Language (SMIL) to coordinate the parts. Pronounced “smile,” SMIL uses a simple but powerful mark-up language for specifying when and how files play.

When you complete your SMIL file, you put it on RealServer and link your Web page to it. Persons viewing your page then click the link to view your presentation. If you have just one file in your presentation, such as a single video file, you don’t need to create a SMIL file, however. Just link your Web page to that file.

Additional Information

See Chapter 7 starting on page 59 for information on laying out your presentation in a Web page instead of in RealPlayer. See Chapter 9 beginning on page 75 for instructions on linking your Web page to your SMIL file.

Creating a SMIL File

You can create a SMIL file (file extension `.smi`) with any text editor or word processor that can save output as plain text with line breaks. If you are familiar with HTML mark-up, you will pick up SMIL quickly. In its simplest form, a SMIL file lists multiple media files played in sequence:

```
<smil>
  <body>
    <audio src="rtsp://realserver.company.com/one.ra"/>
    <audio src="rtsp://realserver.company.com/two.ra"/>
    <audio src="rtsp://realserver.company.com/three.ra"/>
  </body>
</smil>
```

SMIL General Rules

SMIL has many similarities to HTML, but also some important differences. When you create your SMIL file, keep the following general rules in mind:

- The SMIL file must start with a <smil> tag and end with the </smil> closing tag. All other mark-up appears between these two tags:

```
<smil>
 ...all other SMIL mark-up...
</smil>
```

- The <body> and </body> tags are required, but <head> and </head> tags are optional.
- SMIL tags and attributes must all be lowercase.
- A tag that does not have a corresponding end tag (for example, the <smil> tag has the end tag </smil>), must close with a forward slash. For example:

```
<audio src="first.ra"/>
```

- Attribute values, such as "`first.ra`" shown above, must be enclosed in double quotation marks.
- Save your file with the file extension `.smi`. Do not include spaces in the file name. For example, you can have the file `my_presentation.smi` but not the file `my presentation.smi`.
- As in HTML, you can add a comment to a SMIL file like this:

```
<!-- This is a comment -->
```

- This document indents tags to various levels to illustrate the SMIL structure, but this is not required. Indenting your own SMIL files like the examples here will help you keep track of the file functions, though.

Specifying File Locations

To specify a file used in your presentation, you use in the body of the SMIL file a tag that describes the file type and location:

```
<audio src="rtsp://realserver.company.com/audio/first.ra"/>
```

The tag begins with a file type attribute listed in the following table.

File Type	Used For
animation	Animation files, such as Shockwave Flash files used in a RealFlash presentation.
audio	Audio files.
img	Still images such as GIFs or JPEGs.
ref	Any file type not covered by other attributes, such as a multimedia synchronization file.

(Table Page 1 of 2)

File Type	Used For
text	Static text files.
textstream	Streaming RealText files.
video	Video or other files that display continuous motion.

(Table Page 2 of 2)

Although the tag must start with one of these type attributes, the attributes do not affect playback because RealPlayer determines the file type through other means. Specifying a text file as "audio," for example, does not adversely affect playback.

Following the file type, the src attribute lists the file location. How you specify this location varies depending on whether you plan to stream the presentation over a network with RealServer or access the presentation locally.

Linking to Files on RealServer

When a RealMedia presentation streams over a network, the media files reside on RealServer. Each **src** attribute in the SMIL file provides a URL to a media file:

```
src="rtsp://realserver.company.com:6060/audio/first.ra"
```

The following table explains the URL components:

URL Component	Meaning
rtsp://	This designates RealServer's RTSP streaming protocol. In contrast, URLs in Web pages start with "http://".
realserver.company.com	The address varies for each RealServer. Instead of a name, it may use a TCP/IP address such as `172.2.16.230`. It typically uses an identifier such as "realserver" instead of "www".
:6060	This is the port RealServer uses for RTSP connections. If the port number is required, separate it from the address with a colon.
/audio/	The RealServer administrator sets up the directory structure, which may be more than one level deep as shown in this example.
first.ra	This is the file name you choose.

Contact your RealServer administrator to get the RealServer address, RTSP port, and directory structure.

Using Relative URLs

If your presentation includes many files that are on the same RealServer, you can make each URL relative to a base target that you define in the header:

```
<head>
  <meta name="base" content="rtsp://realserver.company.com/"/>
</head>
<body>
  <audio src="audio/first.ra"/>
  <audio src="audio/second.ra"/>
  <audio src="rtsp://realserver.real.com/audio/third.ra"/>
</body>
```

Because the third file has a full URL specified for it, the base target is ignored. For the first two files, however, the src values are appended to the base target, effectively giving the files these URLs:

```
rtsp://realserver.company.com/audio/first.ra
rtsp://realserver.company.com/audio/second.ra
```

Tip

The relative syntax for SMIL files works like relative links in HTML. You can find additional information about this topic in an HTML reference.

Linking to Local Files

If your presentation files will reside on the user's local computer (as with a multimedia tutorial included with a software application, for example), you include the SMIL file locally as well. The **src** attributes in the SMIL file list presentation files in this format:

```
src="audio/first.ra"
```

This example is a local, relative link to a file that resides one level below the SMIL file in the `audio` directory. For local access, you typically want to use relative links because you cannot be sure where users will place files on their machine.

Alternately, you can use absolute, local links to specify exact locations. The syntax for absolute links is the same as with HTML. It varies with operating systems, however, and you should be familiar with the directory syntax for the system you're using. For example, the following absolute link syntax works for Windows machines, but not UNIX or Macintosh:

```
src="file://c:\audio\first.ra"
```

Warning

Microsoft Internet Explorer 3.0 tries to display local SMIL files as HTML. To support this browser, omit the <head> tag. This problem does not occur with Netscape Navigator

or Internet Explorer 4.0. Nor does it occur when you stream files from RealServer to Internet Explorer 3.0.

Adding a Header

The SMIL file can have a header section that defines aspects of the entire presentation, such as its title:

```
<smil>
   <head>
     ...all header information...
   </head>
   <body>
     ...all body information...
   </body>
</smil>
```

In the SMIL file header, you typically provide author, title, and copyright information that shows up in the RealPlayer status panel. To do this, you use <meta> tags that have name and content attributes as shown here:

```
<head>
  <meta name="author" content="Jane Morales"/>
  <meta name="title" content="Multimedia My Way"/>
  <meta name="copyright" content=”(c)1998 Jane Morales"/>
</head>
```

Within the body, you can override header elements as needed by adding author, title, and copyright attributes to source tags:

```
<body>
  <video src="first.rm"/>
  <video src="second.rm"
   author="Sam Clark"
   title="Planning is the Key"/>
</body>
```

When the second file in this example plays, the author and title displayed in RealPlayer change to new values, but the copyright stays the same. You can also specify values for groups of files by including the author, title, and copyright attributes in <seq> and <par> tags instead of the source tags.

Organizing a Presentation

With the SMIL <seq> and <par> tags you can specify how and when each clip plays. The following sections explain how to play files in sequence or parallel, as well as how to add timing information to tune the presentation.

Playing Files in Sequence

To play clips in sequence, use the <seq> tag. In the following example, the second clip begins when the first clip finishes.

```
<seq>
  <audio src="audio/newsong.ra"/>
  <audio src="audio/oldsong.ra"/>
</seq>
```

If your presentation included just the files above, you wouldn't need to use the <seq> tag. You could simply list the files in order and RealPlayer would play them in sequence. The <seq> tag is used mostly in combination with <par>.

Playing Files Simultaneously

You can play two or more clips at the same time by using the <par> ("parallel") tag. For example, the following plays a synchronized group of RealAudio, RealVideo, and RealText clips:

```
<par>
  <audio src="audio/newsong.ra"/>
  <video src="video/newsong.rm"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

RealSystem ensures that the clips stay synchronized. If some video frames don't arrive, for example, the system either drops those frames or delays playback of the other files until the frames arrive. If another part of the presentations follows in sequence after a <par> group, that part plays when all files in the <par> group finish.

Specifying Timing

Timing elements let you specify when a file starts playing and how long it plays. All timing elements are optional. If you do not set them, clips start and stop according to their normal timelines and their positions within <par> and <seq> groups. The easiest way to designate a time is with shorthand markers of **h**, **min**, **s**, and **ms** as illustrated in the following table.

Shorthand Example	Value
2.5h	2 hours, 30 minutes
2.75min	2 minutes, 45 seconds
15.55s	15 seconds, 550 milliseconds
670.2ms	670.2 milliseconds

You can also express time elements in an "normal play tme" format that includes an "npt:" prefix:

```
"npt=hh:mm:ss.xy"
```

Here, **hh** is hours, **mm** is minutes, **ss** is seconds, **x** is tenths of seconds, and **y** is hundredths of seconds. In this example:

```
"npt=02:34.0"
```

the time value is 2 minutes, 34 seconds. If the value does not include a decimal point, RealPlayer takes the last value to be seconds. So it reads the following value as 2 minutes, 34 seconds rather than 2 hours, 34 minutes:

```
"npt=02:34"
```

Setting Begin Times

You can use a begin attribute like the following to start a clip at a specific point within the presentation timeline:

```
begin=”20.5s”
```

In a <par> grouping, this starts the clip 20.5 seconds after the group becomes active. In a <seq> grouping, this adds 20.5 seconds of blank time before the clip begins. The begin attribute works for images, audio, video, or any other file type.

Setting Duration Times

The duration attribute is for media clips with or without internal timelines. It controls how long the clip appears in the presentation:

```
begin=”20.5s” dur="14.5s"
```

Applying a begin time of 20.5s and a duration of 14.5s to a graphic image, for example, makes the image appear 20.5 seconds after its part of the presentation begins. The graphic disappears 14.5 seconds later.

Setting a Fill

The fill attribute determines what happens when the duration time elapses. It can have either of these values:

- remove (default)

  Remove the clip. When used with a still image, the image disappears once the duration time has elapsed.

- freeze

  Freeze the clip on its last frame. When used with a video, for example, the video’s last frame freezes on the screen.

Suppose that you have a 20-second video and specify a 30-second duration with a "freeze":

```
dur="30s" fill="freeze"
```

After the video plays, its last frame displays for 10 seconds. The video disappears when the duration time expires:

Setting Clip Begin and End Times

The clip-begin and clip-end attributes are for files that have internal timelines, such as audio or video files. They specify the internal timing marks where playback begins and ends:

```
clip-begin=”10.5s” clip-end=”50.7s”
```

In this example, the clip starts playing at its internal 10.5-second mark rather than at its normal beginning. It stops playing when it reaches its 50.7-second mark, playing for a total of 40.2 seconds.

Clip Timing Example

The following example shows two audio files with different timing options:

```
<par>
  <audio src="song1.ra" clip-begin="30.4s” clip-end=”60.4s"/>
  <audio src="song2.ra" begin="28s" clip-begin=”2.4s”
   clip-end=”13.7s”/>
</par>
```

The timing options modify the <par> tag so that the two clips start at different times. The first clip begins to play immediately, but starts at 30.4 seconds into its timeline. Because it ends at 60.4 seconds into its timeline, it plays for exactly 30 seconds.

The second clip is delayed for 28 seconds. That means it overlaps the first clip by 2 seconds. It starts at 2.4 seconds into its timeline and ends at 13.7 seconds into its timeline, thus playing for 11.3 seconds. The total playing time for this group is 30 seconds for the first clip, plus 11.3 seconds for the second clip, minus the 2 second overlap: 39.3 seconds.

Combining <seq> and <par> Tags

You can combine and nest <seq> and <par> tags as you need them. Note that the organization of these tags greatly affects the presentation playback.

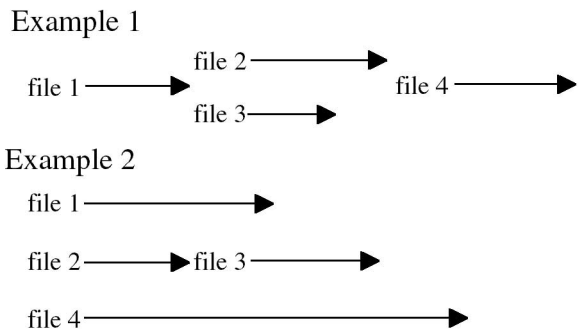
```
<seq>
  file 1
  <par>
    file 2
    file 3
  </par>
```

```
  file 4
</seq>
```

In the example above, file 1 plays first. When it finishes, file 2 and file 3 play together. When both file 2 and file 3 have finished, file 4 plays. You get very different results, though, if you switch the <seq> and <par> groupings:

```
<par>
  file 1
  <seq>
    file 2
    file 3
  </seq>
  file 4
</par>
```

In this example, file 1, file 2, and file 4 all begin at the same time. When file 2 finishes, file 3 starts. The following figure illustrates the difference between these different groupings.



Switching Between Alternate Choices

With the <switch> tag, you can specify multiple options that RealPlayer can choose between:

```
<switch>
  <type src="location1" test-attribute="value1"/>
  <type src="location2" test-attribute="value2"/>
</switch>
```

The <switch> tag specifies any number of choices. RealPlayer looks at choices in order, evaluating the test attribute and its value to determine which file to choose. If the test attribute is system-language for example, each switch statement lists the location of a file for a different language. RealPlayer then chooses a file according to its language preference setting.

Setting Language Choices

The following table shows the languages that RealPlayer currently supports.

Language	Test-Attribute Value
English	en
French	fr
German	gr
Japanese	jp

The following example shows a video slide show with separate audio narrations in English, French, and German. Based on its language preference and the system-language value in the SMIL file, RealPlayer determines which file to play:

```
<par>
  <video src="rtsp://realserver.company.com/slides/seattle.rm"/>
  <!--select audio based on RealPlayer language-->
  <switch>
    <audio src="english/seattle.ra" system-language="en"/>
    <audio src="french/seattle.ra" system-language="fr"/>
    <audio src="german/seattle.ra" system-language="gr"/>
  </switch>
</par>
```

Setting Bandwidth Choices

To take advantage of high bandwidth connections available to some users, you can encode different versions of your files for different bit rates. You then use the <switch> tag to define the choices RealPlayer can make based on its available bandwidth. As shown below, you group files with <par> tags, using the system-bitrate attribute to list the approximate bandwidth (in Kbps) each group consumes:

```
<switch>
  <par system-bitrate=75000>
    <!--for dual isdn and faster-->
    <audio src="audio/newsong1.ra"/>
    <video src="video/newsong1.rm"/>
    <textstream src="lyrics/newsong1.rt"/>
  </par>
  <par system-bitrate=47000>
    <!--for single isdn-->
    <audio src="audio/newsong2.ra”/>
    <video src="video/newsong2.rm"/>
    <textstream src="lyrics/newsong2.rt"/>
  </par>
  <par system-bitrate=20000>
    <!--for 28.8 modems-->
```

```
    <audio src="audio/newsong3.ra"/>
    <video src="video/newsong3.rm"/>
    <textstream src="lyrics/newsong3.rt"/>
  </par>
</switch>
```

Always list system bit rate options from highest to lowest. RealPlayer evaluates options in the order listed, selecting the first viable option even if subsequent options suit it better. So if the 28.8 Kbps option is first, a RealPlayer with a dual-ISDN connection will choose that option because it is the first viable option listed.

Also make sure that the last option satisfies the lowest bit rate connection you want to support. If you do not list an option suitable for 28.8 Kbps modems, for example, RealPlayers connected through those modems will not play the presentation.

Using Multiply Encoded RealMedia Files

With RealAudio or RealVideo files encoded for multiple bit rates, you may or may not need to use the <switch> tag:

- When the presentation consists soley of the multiply encoded RealAudio or RealVideo file, simply link to that file within the SMIL file. The media file then streams at the rate appropriate for RealPlayer's connection speed.
- Use the <switch> tag when combining a multiply encoded RealAudio or RealVideo file with other streaming files that do not support multiple bandwidth connections. All options within the <switch> tag will list the same RealAudio or RealVideo file. But each option will list a unique version of the other media file encoded for a specific bandwidth.

Laying Out Multiple Clips

If your presentation plays only one clip at a time, you do not need to specify a layout. Each clip automatically plays in the main RealPlayer window, the window resizing automatically for each new clip. When your presentation displays several clips at a time, however, you can define separate playback areas ("regions") within the main RealPlayer window through the <layout> tag:

1. In the SMIL file header, you use the <region> tag to name each area and define its size and location within the main RealPlayer window. (See "Defining the Layout" below.)
2. In the file body, you use region attributes to specify which source files play in which areas. (See "Assigning Clips to Regions" on page 57.)

Defining the Layout

When you lay out regions, you use a simple coordinate system measured across and down from the top, left-hand corner of RealPlayer's main window. Offset measurements are in pixels, with zero pixels as the default. The following example defines a root-layout region that sets the overall window size, and also defines two regions for displaying video and text:

```
<head>
  <layout>
    <root-layout background-color="maroon"
     height=”230” width=”250”/>
    <region id="videoregion" top="5" left="5"
     height=”180” width=”240”/>
    <region id="textregion" top="200" left="5"
     height=”20” width=”240”/>
  </layout>
</head>
```

Defining the Background Region

You specify the size of the entire playback area with root-layout. You cannot play media files in the root-layout area, however. The example shown above creates a playback area 230 pixels high by 250 pixels wide. When the presentation begins, the RealPlayer main window expands to this size. Other regions measure their top and left offsets from the upper, left-hand corner of the root-layout area.

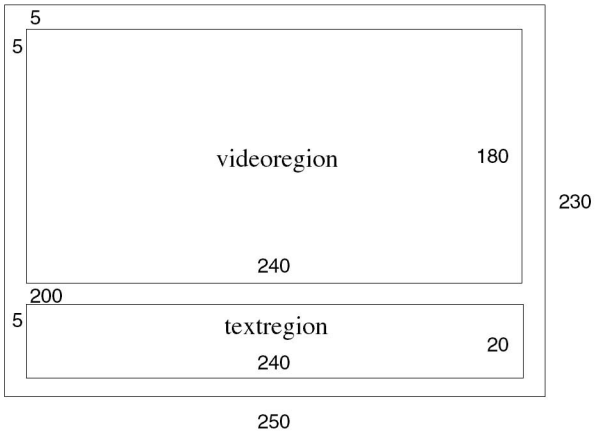
Note

If you leave root-layout out, RealPlayer calculates the total playback area size based on the sizes and offsets of the defined regions. RealNetworks recommends that you always define root-layout, however.

Defining Playback Regions

You create playback regions for media files with <region> tags. These regions must lay within the background area defined by root-layout. Any part of a region outside the background area is cut off. The example above shows two playback areas named “videoregion.” and “textregion.” Both areas are offset 5 pixels to the right of the background area's left edge. The video region displays

5 pixels down from the top of the root-layout area, and the text region displays 200 pixels down. The following figure illustrates this placement.



Note

Layout values expressed as percentages are not supported in this preview release. Use pixel values instead.

Each region must specify a height and width in pixels. The media clip played in a region may be a different size than the region, however. In this case, you use the **fit** attribute to determine how RealPlayer fits the file to the region:

```
<region id="videoregion" top="5" left="5"
height=”180” width=”240” fit="meet"/>
```

The **fit** attribute uses one of the values described in the following table. Note that in no case will media display outside the region's boundaries.

Attribute	Action
fill	Scale the clip's height and width so that it fills the region exactly.
hidden (default)	Place the clip at the region's upper, left-hand corner. If the clip is smaller than the region, fill remaining space with the region's background color. If the clip is larger than the region, crop out the area that doesn't fit.
meet	Place the clip at the region's upper, left-hand corner. Scale the clip and preserve its height/width ratio until one dimension is equal to the region's size and the other dimension is within the region's boundaries. Fill empty space with the region's background color.
scroll	Place the clip at the region's upper, left-hand corner. Scale clip to normal size and add scroll bars for height and width as necessary.
slice	Place the clip at the region's upper, left-hand corner. Scale the clip and preserve its height/width ratio until one dimension is equal to the region's size and the other dimension overflows the region's boundaries. Crop the overflow.

Note

For this preview release, only the default value of "hidden" is supported for the **fit** attribute.

When scaling media inside a region, keep in mind that different types of content scale with different results. A video scaled larger than its encoded size may not look good. Vector-based media such as RealFlash animation scale more easily to fit different region sizes, however.

Adding a Background Color

In the SMIL layout, you can specify background colors for any region:

```
<layout>
  <root-layout id="background" background-color="maroon"/>
  <region id="videoregion" background-color=”silver”.../>
  <region id="textregion" background-color=”#C2EBD7”.../>
</layout>
```

The default background color for all regions is black. When a media clip plays, it overlays and hides the background color for its region. The background color shows through, however, if the clip contains transparency.

For the color value, you can use any Red/Green/Blue hexadecimal value (#RRGGBB) supported by HTML, as well as one of the following predefined color names, listed here with their corresponding hexadecimal values:

white (#FFFFFF)	silver (#C0C0C0)	gray (#808080)	black (#000000)
yellow (#FFFF00)	fuschia (#FF00FF)	red (#FF0000)	maroon (#800000)
lime (#00FF00)	olive (#808000)	green (#008000)	purple (#800080)
aqua (#00FFFF)	teal (#008080)	blue (#0000FF)	navy (#000080)

Note

For this preview release, background colors are supported only for the root-layout area.

Overlapping Clips

If regions overlap, you can determine which regions appear in front with the z-index attribute. The following example creates a video region that overlaps an image region:

```
<head>
  <layout>
    <root-layout background-color="gray"
     height=”220” width=”280”/>
    <region id="imageregion" top="10" left="10"
     height=”200” width=”260” z-index=”0”/>
    <region id="videoregion" top="20" left="20"
     height=”180” width=”240” z-index=”1”/>
```

```
  </layout>
</head>
```

This example defines a gray root-layout area 220 pixels high by 280 pixels wide. A smaller image region is centered within this gray background. Its z-index of zero makes it display behind all other regions, but not behind the root-layout area. The video region centered in the image region appears on top of that region because of its higher z-index value. Another region could overlap the video region by using a z-index of 2 (or 5, or 8, or 29, for instance).

Here are points to observe when using the z-index:

- The root-layout area always behind beneath all other regions and does not use a z-index.
- The z-index values start at 0 (zero) and proceed through the positive integers without limit. You can't use a negative value such as -4.
- The default value of 0 (zero) applies if you don't specify the z-index.
- Using strictly sequential values such as 0, 1, 2, 3, 4 helps you keep track of the layers, but is not necessary. A sequence such as 0, 1, 6, 19, 34 works just as well. And leaving gaps in the sequence makes it easier to insert layers later.
- Nonoverlapping clips can have the same z-index values. Side-by-side videos can both have a z-index of 3, for example
- When overlapping clips (say, clip A and clip B) have the same z-index, the following rules determine which clip appears in front:
  - If clip B starts later in the presentation than clip A, it appears in front of clip A.
  - If both clips start at the same time, the clip listed later in the SMIL file appears in front.

Assigning Clips to Regions

After you define the layout in the header section, you use region attributes within the source tags to attach each source to a region:

```
<body>
  <par>
    <video src="video.rm" region="videoregion"/>
    <audio src="audio.ra"/>
    <textstream src="text.rt" region="textregion"/>
  </par>
</body>
```

In the example above, the video and text files are assigned to the video and text regions defined in the header. You can "reuse" regions by assigning sequential clips to them. For example, you can play a video clip in a region, then display

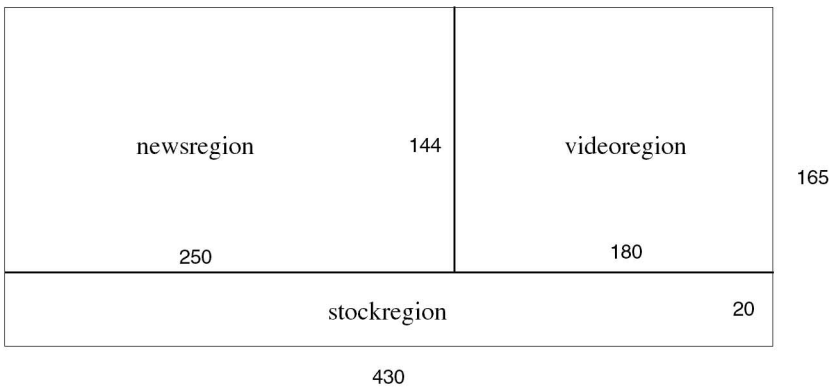
another clip in that region after the first clip finishes. Don't assign the same region to two clips that play at the same time, however. You don't assign audio files to regions at all because audio does not require a display area.

SMIL Layout Example

The following example displays three regions: a news region, a video region, and a stock ticker region. The news and video regions are arranged side by side at the top of the main RealPlayer window. The stock ticker region appears below them.

```
<smil>
  <head>
    <!--presentation with 2 text files and 1 video file-->
    <meta name="title" content="Music of the Week"/>
    <layout>
      <root-layout height=”165” width=”430”/>
      <region id="newsregion" top=”0” left=”0”
       height=”144” width=”250”/>
      <region id="videoregion" top=”0” left=”250”
       height=”144” width=”180”/>
      <region id="stockregion" top=”145” left=”0”
       height=”20” width=”430”/>
    </layout>
  </head>
  <body>
    <par>
       <!--play these 3 clips simultaneously-->
       <textstream src="news.rt" region="newsregion"/>
       <video src="newsvid.rm" region="videoregion"/>
       <textstream src="stocks.rt" region="stockregion"/>
    </par>
  </body>
</smil>
```

The following figure shows the design of these regions.



Chapter 7: Playing a Presentation in a Web Page

As Chapter 6 explains, playing your presentation back in RealPlayer simply requires a hypertext link from your Web page to a SMIL file. But you can also have a presentation play back directly in your Web page, even adding controls such as fast forward and pause.

Choosing the Netscape Plug-in or ActiveX Control

To provide Web page playback, RealPlayer includes a plug-in for browsers that support the Netscape plug-in architecture:

- Netscape Navigator 3.0 and 4.0
- Microsoft Internet Explorer 3.0 and 4.0

It also has an ActiveX control that provides playback capabilities within these products:

- Internet Explorer 3.0 and 4.0
- Visual Basic applications

Because they both have the same capabilities, you can use either the plug-in or the ActiveX control depending on which products you need to support. The following sections describe the basics of using the plug-in or the control, then explain each option you can set.

Tip

Familiarity with RealPlayer and HTML will make it easier to use the plug-in or control.

Using <EMBED> Tags for the Netscape Plug-In

To use RealPlayer's Netscape plug-in, you add <EMBED> tags to your Web page HTML. Each <EMBED> tag has three required parameters (SRC, WIDTH, HEIGHT), and can include many optional parameters. The basic <EMBED> tag looks like the following (the SRC value has been omitted for simplicity):

```
<EMBED SRC="..." WIDTH=300 HEIGHT=134>
```

This tag creates a playback area 300 pixels wide by 134 pixels high within the Web page. Parameters typically have the form `PARAMETER=value`. The parameter names can be any case, though this manual shows them uppercase. Except for file names, which must typically be lowercase, parameter values are not case-sensitive. Unless they are URLs, parameter values do not need to be inside quotation marks.

Supporting Other Browsers

To accommodate browsers that do not support the Netscape plug-in, use <NOEMBED> to define a standard hypertext link to your presentation. The unembedded link follows the <EMBED> tag:

```
<EMBED SRC="..." WIDTH=300 HEIGHT=134>
<NOEMBED><A HREF="...">Play with RealPlayer.</A></NOEMBED>
```

In this example, browsers that can play the embedded presentation hide the text between <NOEMBED> and </NOEMBED>. Other browsers ignore the preceding <EMBED> tag and display just the hypertext link. The user then clicks the link to play the presentation in RealPlayer.

Using <OBJECT> Tags for the ActiveX Control

You embed the RealPlayer ActiveX control in HTML pages with the <OBJECT> tag. This tag uses an ID that you select, such as `ID=RVOCX`, and must have the following class ID that identifies RealPlayer:

```
CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
```

The <OBJECT> tag also sets the width and height of the playback area within the browser. A typical <OBJECT> tag looks like this:

```
<OBJECT ID=RVOCX
CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
WIDTH=300 HEIGHT=134>
... parameters ...
</OBJECT>
```

This tag creates a playback area 300 pixels wide by 134 pixels high within the Web page. Between <OBJECT> and </OBJECT>, you can define any number of additional parameters in this form:

```
<PARAM NAME="name" VALUE="value">
```

PARAM, NAME, and VALUE markers can be any case, though this manual shows them uppercase. Parameter values are not case-sensitive except for file names, which must typically be lowercase. Always enclose parameter values in double quotation marks.

Setting Basic Parameters

Both the Netscape plug-in and ActiveX control use the same basic tag parameters. As explained above, however, the tag syntax for the plug-in and the control differs. The following sections explain the basic parameters you can include in each <EMBED> or <OBJECT> tag.

SRC

The SRC parameter, which is required for each <EMBED> tag, gives the presentation's source URL surrounded by double quotes. The directory names cannot contain spaces. Here is an example of the SRC parameter within the <EMBED> tag:

```
SRC="http://realserver.company.com:8080/ramgen/sample.smi?embed"
```

The `?embed` option at the end of the URL causes RealPlayer to play the presentation back in the Web page. If you do not include this option, the presentation plays back in RealPlayer.

For the ActiveX control, the <OBJECT> tag's CLASSID parameter eliminates the need to include the `?embed` option in the URL:

```
<PARAM NAME="SRC"
VALUE="http://realserver.company.com:8080/ramgen/sample.smi">
```

Additional Information

For more on presentation URLs, see "Linking your Web Page to your Presentation" on page 75.

WIDTH and HEIGHT

Required for each <EMBED> or <OBJECT> tag, the WIDTH and HEIGHT parameters set the size of the playback area. If you leave them out, the playback area may appear as a tiny icon because streaming media presentations do not size automatically. The values for WIDTH and HEIGHT are in pixels by default, so a width of 300 creates a playback area 300 pixels wide. Setting WIDTH and HEIGHT to 0 (zero) hides the playback area.

You can also express WIDTH and HEIGHT as percentages of the browser window size. For example, a width of 50% makes the presentation area half the browser width. Keep in mind that different types of content scale with different results. A video scaled larger than its encoded size may not look good. Vector-based media such as RealFlash animation scale more easily to fit different playback areas, however.

CENTERED

The default value for CENTERED is false, which causes the media file to fill the entire playback area. If you set CENTERED to true, the media file is centered within the playback area and displays at its encoded size. So by using

CENTERED=true, you can create a large playback area with WIDTH and HEIGHT and still have the media file play at its normal size.

Adding RealPlayer Controls

With the CONTROLS parameter, you can place RealPlayer controls such as a Play/Pause button on your Web page. A visitor to your page can then control the presentation playback just as if using RealPlayer as a separate application. The following example for the Netscape <EMBED> tag displays the Play/Pause button:

```
<EMBED SRC="..." WIDTH=300 HEIGHT=134 CONTROLS=PlayButton>
```

For the ActiveX control, you define a CONTROLS parameter within the <OBJECT> tag structure:

```
<OBJECT ID=RVOCX CLASSID="...” WIDTH=312 HEIGHT=140>
<PARAM NAME="SRC" VALUE="...">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
</OBJECT>
```

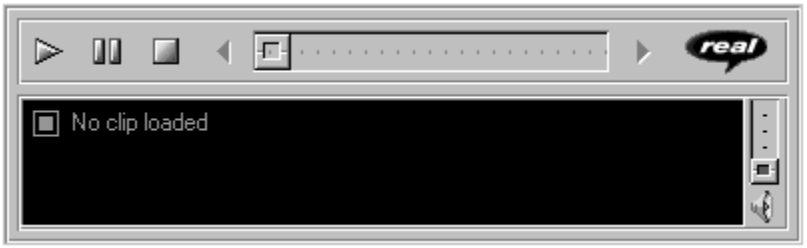
The following sections describe each RealPlayer control you can use. A tag’s WIDTH and HEIGHT parameters set the control’s size. The standard pixel widths and heights given below produce embedded controls approximately the same sizes as the RealPlayer controls. Specifying different pixel sizes scales the controls larger or smaller. You can also use percentage values for sizes, but this is recommended only for the image window.

Additional Information

For information on using more than one control in your Web page, see “Using Multiple Controls” on page 66.

Full Controls

All



Displays the RealPlayer Control Panel, Information and Volume Panel, and Status Bar. The control name "default" also works.

Standard pixel width:	400
Standard pixel height:	120

ImageWindow

Displays the image window. This is available only for display presentations such as video or animation. Even if no other controls are visible on the page, the user can typically right-click (on Windows) or hold down the mouse button (Macintosh) on the playback area to display a menu of choices such as Play/Pause and Stop.

Standard pixel width:	none
Standard pixel height:	none

Individual Controls and Sliders

ControlPanel

Displays the Play/Pause button, the Stop button, Fast Forward and Rewind controls, and the position slider.

Standard pixel width:	400
Standard pixel height:	36

PlayButton

Displays the Play/Pause button.

Standard pixel width:	44
Standard pixel height:	26

PlayOnlyButton

Displays the Play button without Pause.

Standard pixel width:	26
Standard pixel height:	26

PauseButton

Displays the Pause button without Play.

Standard pixel width:	26

Standard pixel height: 26

StopButton

Displays the Stop button.

Standard pixel width: 26
Standard pixel height: 26

FFCtrl

Displays the Fast Forward control.

Standard pixel width: 26
Standard pixel height: 26

RWCtrl

Displays the Rewind control.

Standard pixel width: 26
Standard pixel height: 26

MuteCtrl

Displays the Mute control button.

Standard pixel width: 26
Standard pixel height: 26

MuteVolume

Displays the Mute control and volume slider.

Standard pixel width: 26
Standard pixel height: 88

VolumeSlider

Displays the volume slider.

Standard pixel width: 26
Standard pixel height: 65

PositionSlider

Displays the clip position slider.

Standard pixel width: 240
Standard pixel height: 22

Information Panels

TACCtrl

Displays the Title, Author, and Copyright control. See also "Suppressing Labels" on page 66.

Standard pixel width:	tbd
Standard pixel height:	tbd

InfoVolumePanel

Displays the Title, Author, and Copyright information panel, as well as the volume slider. See also "Suppressing Labels" on page 66.

Standard pixel width:	300
Standard pixel height:	88

InfoPanel

Displays the Title, Author, and Copyright information panel. See also "Suppressing Labels" on page 66.

Standard pixel width:	300
Standard pixel height:	88

Status Panels

StatusPanel

Displays the status panel, which shows informational messages, current place in the presentation timeline, and total clip length. If you do not embed a status panel in your page, error messages display in the browser's status bar.

Standard pixel width:	tbd
Standard pixel height:	tbd

PositionField

Displays the field of the status bar that shows current position in the presentation timeline and total clip length.

Standard pixel width:	tbd
Standard pixel height:	tbd

StatusField

Displays the message text area of the status bar.

Standard pixel width:	tbd
Standard pixel height:	tbd

StatusBar

Displays the status field, position field, and channels (stereo/mono).

Standard pixel width:	tbd
Standard pixel height:	tbd

Suppressing Labels

When you use a control that includes the Title, Author, and Copyright fields, you can include the NOLABELS option to suppress that information. Here is an example for the Netscape plug-in:

```
<EMBED ... CONTROLS=All NOLABELS=true>
```

and an example for the ActiveX control:

```
<OBJECT ...>
<PARAM NAME="CONTROLS" VALUE="All">
<PARAM NAME="NOLABELS" VALUE="true">
</OBJECT>
```

Using Multiple Controls

The CONSOLE parameter defines a name that unifies various <EMBED> or <OBJECT> tags so that the controls work together. For example, you could create three separate <EMBED> or <OBJECT> tags to define an image window, a Play button, and a Stop button. By using three separate tags, you can can set the size of each control and specify its layout. You could put each control in a different table cell, for example.

Each <EMBED> or <OBJECT> tag for a single presentation defines the same CONSOLE name, or uses one of these predefined names:

- `_master` links all embedded controls on the page.
- `_unique` links to no other embedded controls on the page.

You can have multiple console names for separate presentations. For a page that has two video presentations, for example, you can define console names of

`video1` and `video2`. All controls for `video1` interoperate and all controls for `video2` interoperate. But a volume slider for `video1`, for example, will not affect `video2` controls.

Notes on Using Consoles

Note the following when grouping multiple controls with CONSOLE attributes:

- Every <EMBED> tag must have a SRC attribute. Tags linked by a console name should have the same SRC value.
- With the ActiveX control, only one <OBJECT> tag in a console group needs to have a SRC value.
- If the <EMBED> or <OBJECT> tags in a console group have different SRC values, the first valid source that RealPlayer finds among those choices becomes the console source. This may not always be the first source listed.
- Clicking a Play button for one console stops playback for other consoles. This allows multiple consoles to play separate audio tracks or to use the same image window.

Multiple Controls Example

The following examples for the <EMBED> and <OBJECT> tags set up an image window and two sets of controls (a Play button and Stop button) for separate videos, `sample1.rm` and `sample2.rm`. By using the predefined console name `_master`, the image window links to both control sets. The control sets use different console names, however, so they do not link to each other. Clicking each Play button therefore starts a different video.

Because each <EMBED> tag must have a source attribute, the image window in the Netscape plug-in example simply uses the same source as the first Play button. AUTOSTART is not used, however, so the viewer simply clicks either Play button to start a video. Clicking the other Play button stops the first video and plays the second video.

Netscape Plug-in Sample Mark-up

```
<EMBED SRC="http://realserver.company.com/sample1.rm"
WIDTH=176 HEIGHT=128 CONTROLS=ImageWindow CONSOLE=_master>

<H4>Video 1</H4>
<EMBED SRC="http://realserver.company.com/sample1.rm"
WIDTH=50 HEIGHT=50 CONTROLS=PlayButton CONSOLE=video1>
<EMBED SRC="http://realserver.company.com/sample1.rm"
WIDTH=50 HEIGHT=50 CONTROLS=StopButton CONSOLE=video1>

<H4>Video 2</H4>
<EMBED SRC="http://realserver.company.com/sample2.rm"
WIDTH=50 HEIGHT=50 CONTROLS=PlayButton CONSOLE=video2>
```

```
<EMBED SRC="http://realserver.company.com/sample2.rm"
WIDTH=50 HEIGHT=50 CONTROLS=StopButton CONSOLE=video2>
```

ActiveX Control Sample Mark-up

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=176 HEIGHT=128>
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
<PARAM NAME="CONSOLE" VALUE="_master">
</OBJECT>

<H4>Video 1</H4>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50 HEIGHT=50>
<PARAM NAME="SRC"
VALUE="http://realserver.company.com/sample1.rm">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
<PARAM NAME="CONSOLE" VALUE="video1">
</OBJECT>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50 HEIGHT=50>
<PARAM NAME="CONTROLS" VALUE="StopButton">
<PARAM NAME="CONSOLE" VALUE="video1">
</OBJECT>

<H4>Video 2</H4>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50 HEIGHT=50>
<PARAM NAME="SRC"
VALUE="http://realserver.company.com/sample2.rm">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
<PARAM NAME="CONSOLE" VALUE="video2">
</OBJECT>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50 HEIGHT=50>
<PARAM NAME="CONTROLS" VALUE="StopButton">
<PARAM NAME="CONSOLE" VALUE="video2">
</OBJECT>
```

Setting Automatic Playback

The AUTOSTART and LOOP parameters let you set the content to start playing automatically and loop continuously. Here is an example of both parameters used in the Netscape plug-in:

```
<EMBED SRC="..." WIDTH=50% HEIGHT=50% AUTOSTART=true
LOOP=true>
```

And an example for the ActiveX control:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50% HEIGHT=50%>
<PARAM NAME="SRC" VALUE="...">
<PARAM NAME="AUTOSTART" VALUE="true">
<PARAM NAME="LOOP" VALUE="true">
</OBJECT>
```

AUTOSTART

When set to true, the AUTOSTART parameter starts the content playing as soon as it loads. When you have multiple <EMBED> or <OBJECT> tags linked by a CONSOLE name, you need to set AUTOSTART to true in just one of the tags. Leaving AUTOSTART out or setting its value to false means the presentation does not start until the user starts it by, for example, clicking an embedded Play button.

LOOP

If this parameter is set to true, the presentation continuously loops until the viewer stops it. When you have multiple <EMBED> or <OBJECT> tags linked by a CONSOLE name, you need to set looping in just one of the tags. The default value of false applies if you leave the loop parameter out. In this case the presentation stops after it plays the first time. The user can always play the presentation again, however, by clicking the Play button.

Working with SMIL Layouts

As “Laying Out Multiple Clips” on page 53 explains, you can use your presentation’s SMIL file to define separate playback regions for the multiple parts of a presentation. This lets you lay out two videos side-by-side, for example. When playing back a presentation in a Web page, you can define the layout in SMIL or through the Netscape plug-in or ActiveX Control.

Defining the Layout with SMIL

Controlling the layout through SMIL is the easier method. You set up the regions and their relative placements in the SMIL file. You then use the Netscape plug-in or ActiveX control to create a playback region in the Web page large enough to accommodate all the regions. This SMIL file will then produce the same layout when played through the Web page or RealPlayer.

The sample layout shown in “SMIL Layout Example” on page 58 defines three regions, creating a total playback area 430 pixels wide by 165 pixels high. To accommodate this in your Web page, you define an area at least as large as this through the <EMBED> or <OBJECT> tag. Here are examples for the Netscape plug-in:

```
<EMBED SRC="..." WIDTH=430 HEIGHT=165 CONTROLS=ImageWindow
CONSOLE=_master>
```

and ActiveX control:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=430 HEIGHT=165>
<PARAM NAME="SRC" VALUE=”...”>
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
<PARAM NAME="CONSOLE" VALUE="_master">
</OBJECT>
```

The SRC parameter provides the URL to the SMIL file. You can then use additional <EMBED> or <OBJECT> tags linked to the `_master` console to provide RealPlayer controls for the presentation.

Defining the Layout with the Plug-in or Control

The second method omits the <layout> tag from the SMIL file header. It simply associates each presentation file with a SMIL region. For example, you would modify the sample layout shown in "SMIL Layout Example" on page 58 to look like the following:

```
<smil>
  <head>
    <!--presentation with 2 text files and 1 video file-->
    <meta name="title" content="Music of the Week"/>
  </head>
  <body>
    <par>
       <!-- play these 3 clips simultaneously -->
       <textstream src="news.rt" region="newsregion"/>
       <video src="newsvid.rm" region="videoregion"/>
       <textstream src="stocks.rt" region="stockregion"/>
    </par>
  </body>
</smil>
```

You then define separate playback areas for each presentation file through <EMBED> or <OBJECT> tags, using REGION parameters to associate each tag with a SMIL region. For example, the <EMBED> tag that plays `news.rt` would look like this:

```
<EMBED SRC="http://www.company.com/sample.smi"
WIDTH=250 HEIGHT=144 CONTROLS=ImageWindow
REGION=newsregion CONSOLE=_master>
```

The <OBJECT> tag would look like this:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=250 HEIGHT=144>
<PARAM NAME="SRC"
VALUE="http://www.company.com/sample.smi">
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
<PARAM NAME="REGION" VALUE="newsregion">
<PARAM NAME="CONSOLE" VALUE="_master">
</OBJECT>
```

You define an <EMBED> or <OBJECT> tag like these for each presentation file. The SRC parameter in each tag lists the same SMIL file. You can also use additional <EMBED> or <OBJECT> tags linked to the `_master` console to provide RealPlayer controls for the presentation.

Note

Keep in mind that this method works best when the presentation plays back in a Web page. The SMIL file will still work with RealPlayer, but because the file does not define the layout, RealPlayer automatically creates a layout.

Using Advanced Parameters

The following are advanced parameters typically used when integrating the Netscape plug-in or ActiveX control into playback environments more complex than simple HTML Web pages.

AUTOGOTOURL

You can use the AUTOGOTOURL parameter if the presentation plays back within a Java applet or VisualBasic application. The parameter determines how URLs in the presentation are handled. The default value of true applies if you leave the parameter out. In this case any URL embedded in the presentation goes to the browser. If you set this parameter to false, RealPlayer sends the URL to the VisualBasic application or Java applet with the `OnGotoURL()` call.

NAME

NAME is an optional parameter for the Netscape plug-in <EMBED> tag:

```
<EMBED NAME=vid SRC="..." WIDTH=300 HEIGHT=134>
```

If you give the plug-in instance a name, you can refer to it through a JavaScript command such as this:

```
<Input Type="button" Value="play"
onClick=”document.vid.play()”>
```

Note

With the ActiveX control, you refer to the ID instead of a name.

WINDOWED

This parameter is used with the Netscape plug-in only. Its default value is true. If you set WINDOWED=false in the <EMBED> tag, the plug-in is windowless. A windowless plug-in does not require a native window. It can be opaque or transparent and can be invoked in HTML layers.

Additional Information

See the Netscape developer reference at **http://developer.netscape.com/library/documentation/communicator/plugin/index.htm**.

Parameter Reference

The following table summarizes the <EMBED> and <OBJECT> tag parameters you can use.

Parameter	Works with <EMBED>	Works with <OBJECT>	Function	Refer to
AUTOGOTOURL	yes	yes	Sends URLS to applet or VisualBasic application when set to false.	page 71
AUTOSTART	yes	yes	Starts playback automatically.	page 69
CENTERED	yes	yes	Centers clip in playback area.	page 61
REGION	yes	yes	Associates clip with SMIL regions.	page 70
CONSOLE	yes	yes	Links multiple instances of a tag.	page 66
CONTROLS	yes	yes	Adds RealPlayer controls to presentation.	page 62
LOOP	yes	yes	Makes presentation loop continuously.	page 69
NAME	yes	no	Provides reference for JavaScript.	page 71
NOLABELS	yes	yes	Suppresses title, author, and copyright fields in controls.	page 66
SRC	yes	yes	Specifies source file.	page 61
WIDTH and HEIGHT	yes	yes	Sets size of playback area.	page 61
WINDOWED	yes	no	Makes plug-in windowless when set to false.	page 71

Chapter 8: Inserting Ads in your Presentation

RealSystem G2 is backwards compatible with the RealSystem 5.0 ad rotation features. Enhanced ad rotation features will be added in a subsequent release of RealSystem G2.

Chapter 9: Delivering Your Presentation

When you finish crafting your multimedia presentation, you place the files on RealServer for streaming. This chapter explains how to link your Web page to your presentation. It also describes how a Web server can stream some RealMedia presentations.

Moving Files to RealServer

When your media files and SMIL file are ready, transfer them to RealServer and place them in the directories prepared by the RealServer administrator. Check your SMIL file to ensure the following:

- It follows the guidelines described in “SMIL General Rules” on page 44.
- URLs for media source files begin with `rtsp://` when streaming from RealServer and `http://` when streaming or downloaded from a Web server.
- Locations of media files are correct.

Tip

RealNetworks’ publishing tools include features for transferring files to RealServer automatically. See **http://www.real.com** for details.

Linking your Web Page to your Presentation

With your files on RealServer, link your Web page to the SMIL file with an HTML hypertext link that looks like this:

```
<a href=”http://realserver.company.com:8080/ramgen/
sample.smi”>
```

If the presentation plays back directly in the Web page, the URL looks like this:

```
<a href=”http://realserver.company.com:8080/ramgen/
sample.smi?embed”>
```

The following table explains the components of these URLs.

URL Component	Meaning
http://	This makes the browser contact RealServer through the HTTP protocol. (Web browsers do not use the RTSP protocol.)
realserver.company.com	This address varies for each RealServer. It typically uses an identifier such as “realserver” instead of “www”.
:8080	This is the port RealServer uses for HTTP connections. It can vary for each RealServer. Separate the port and address with a colon.
/ramgen/	As “Using RAMGEN” explains, this parameter causes RealServer to generate a RAM file automatically.
sample.smi	This is the SMIL file for your presentation. If you have just one file to stream, you can link directly to that file instead of a SMIL file.
?altplay=*file.ext*	This optional parameter specifies an alternate presentation created for older versions of RealPlayer. See “Listing Alternate Presentations” on page 77.
?embed	This makes the presentation play back in the Web page. See Chapter 7 starting on page 59 for more information.

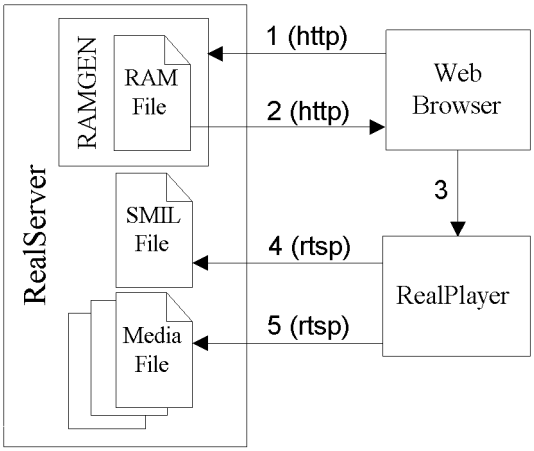
Tip

Contact your RealServer administrator to get the RealServer address, HTTP port, and RAMGEN directory structure.

Using RAMGEN

The `/ramgen/` parameter in your Web page hyperlink causes RealServer to generate a RAM file and download it to the viewer’s Web browser. This step is needed because browsers may not be configured to launch RealPlayer when they receive a SMIL file. But browsers are configured to launch RealPlayer when they receive a RAM file with the `.ram` or `.rpm` extension. RealPlayer then receives the RAM file and can use it to get the SMIL file and presentation

files from RealServer. Because the RAM and SMIL files are small, this interaction occurs quickly.



The figure above illustrates the process of requesting a presentation.

1. Using the HTTP protocol, the Web browser requests the SMIL file from RealServer.
2. The URL to the SMIL file causes RealServer's RAMGEN utility to generate a RAM file automatically and download it to the browser.
3. The RAM file extension (`.ram` or `.rpm`) causes the Web browser to launch RealPlayer as a helper application.
4. RealPlayer receives the RAM file and request the SMIL file from RealServer using the RTSP protocol.
5. With the information in the SMIL file, RealPlayer requests and receives the streaming media files.

Note

Use RAMGEN even when linking to a single file type (such as a `.rm` file) that automatically launches RealPlayer.

Additional Information

If you cannot use RAMGEN to generate the RAM file automatically, you can write the RAM file as described in "Creating a RAM File Manually" on page 80.

Listing Alternate Presentations

When you update existing content for RealSystem G2, you can keep the earlier content available for older versions of RealPlayer. If you have a RealVideo 5.0 presentation, for example, your Web page links to a RAM file. You could modify the presentation by incorporating advanced SMIL features. In your Web

page link, you then use RAMGEN options that cause viewers to link to either the older presentation file or the new SMIL file depending on the RealPlayer version they use:

```
<a href="http://realserver.company.com:8080/ramgen/
sample.smi?altplay="old_sample.rm">
```

This link instructs RealServer to generate a RAM file for `sample.smi` if the viewer has the latest version of RealPlayer. Older versions of RealPlayer receive a RAM file for the older `old_sample.rm` file. (Note that the URL specifies the presentation file, not the old RAM file.) If your previous link specified options such as an end time, you can include those options in the new URL after the altplay parameter. The following shows an end time set for `old_sample.rm`:

```
<a href="http://realserver.company.com:8080/ramgen/
sample.smi?altplay=old_sample.rm?end=7:45">
```

Tip

It is not necessary to keep older content available. If you do not use altplay, viewers with older versions of RealPlayer are prompted to upgrade when they click the link to the SMIL file.

Combining RAMGEN Options

The question mark operator ("?") separates RAMGEN options from the main URL. To use multiple RAMGEN options, you use a question mark before the first option, then separate the remaining options with ampersands (&). For example, the following link uses altplay and embed (the order of options does not matter):

```
<a href="http://realserver.company.com:8080/ramgen/
sample.smi?embed&altplay=old_sample.rm">
```

Streaming from a Web Server

With HTTP streaming, you can stream RealMedia clips from a Web server. You can use HTTP streaming if, for example, your Internet Service Provider (ISP) does not have RealServer. This method is not as robust as streaming from RealServer, but it provides a reasonable method for providing short clips to a limited number of users. HTTP streaming is not recommended for long presentations or presentations viewed simultaneously by large numbers of people, however.

If you plan to stream files from a Web server, configure your server's MIME types as described below. Then transfer your files to the Web server and write a RAM file as described in "Creating a RAM File Manually" on page 80.

Limitations on HTTP Streaming

There are several limitations on presentations streamed by Web servers:

- RealMedia File Types Only

  Except for RealFlash, the RealMedia formats are designed to stream from a Web server as well as from RealServer. A Web server cannot stream other file formats streamed by RealServer, however. A Web server will download rather than stream a WAV file, for example.

- No Multifile, Synchronized Presentations

  Your presentation must consist of a single file or multiple files played in sequence. A Web server cannot deliver multiple streams synchronized to the same timeline. For this reason it cannot play back a RealFlash presentation, for example.

- Limited User Control of Playback Features

  Web server streaming prevents the use of some RealPlayer features. For example, a Web server cannot jump to a new position in a presentation. When the viewer moves the RealPlayer position slider forward, playback stops because the Web server cannot jump to that new position in the file's timeline. The file continues to stream at its normal pace and RealPlayer resumes playback once the stream reaches the requested timeline position.

- No Automatic Detection of Modem Speed

  With HTTP streaming, RealPlayer can't automatically detect the modem speed and determine which version of a presentation to play. Instead, you need to encode separate files optimized for various connection speeds and provide separate links that viewers can select.

- No Live Broadcast

  Live streaming of presentations is not possible because Web servers can stream only presentations stored on disk.

Configuring Web Server MIME Types

To stream RealAudio and RealVideo clips from a Web server, you must define the following MIME types in the server. Some Web servers are preconfigured with these MIME types. If you are using an ISP, ask the ISP's Web server administrator to configure these MIME types for you:

- For files with `.ra`, `.rm`, or `.ram` extensions:
  - `audio/x-pn-RealAudio`
  - `video/x-pn-RealVideo`
- For files with `.rpm` extensions:

- `audio/x-pn-RealAudio-plugin`
- `video/x-pn-RealVideo-plugin`

Creating a RAM File Manually

Whenever possible, have RealServer create the RAM file automatically as described in "Using RAMGEN" on page 76. In some cases, though, you may need to create a RAM file manually:

- Streaming from a RealServer machine not set up to use RAMGEN.
- Streaming a RealAudio or RealVideo file from a Web server.
- Playing back presentation files that reside on a local, desktop machine.

➤ **To create a RAM file:**

1. Open any editor that can save files as plain text. On the top line, enter the URL of the SMIL file or the single media file in the presentation. The following example links to a SMIL file on a RealServer machine that doesn't use RAMGEN:

   `rtsp://realserver.`*`company`*`.com:6060/sample.smi`

   For Web server streaming, you specify the HTTP protocol and the Web server name:

   `http://www.`*`company`*`.com/sample.smi`

   On a local machine, the following specifies a file that resides one level below the RAM file in the `media` directory:

   `file://media/sample.smi`

   Additional Information

   These URLs are like those used in a SMIL file to designate media files. For more information on general URL syntax, see "Specifying File Locations" on page 44.

2. When streaming from RealServer, you can support older versions of RealPlayer just as RAMGEN does with the altplay attribute. You add the marker:

   `--stop--`

   after the RTSP URL and specify the URL for the older file just as in previous versions of RealSystem. Here's an example:

   ```
   rtsp://realserver.company.com:6060/sample.smi
   --stop--
   pnm://realserver.company.com:7070/old_sample.rm
   ```

   Note that this second URL specifies the older `pnm://` protocol and designates the port that RealServer uses for that protocol. Contact the RealServer administrator for that port number.

3. Save the file as text-only with a `.ram` extension (played in RealPlayer) or a `.rpm` extension (played in the Web browser).
4. Link your Web page to the `.ram` or `.rpm` file.

Testing your Presentation

The following are guidelines for making sure your presentation works well and reaches its target audience:

1. Test your presentation in "real world" conditions. If you target 28.8 Kbps connections, for example, request the presentation from RealServer over a 28.8 dial-up modem.
2. Make sure that your presentation works well for an "average" CPU. RealNetworks recommends testing a presentation on both Pentium and Power Macintosh 90 MHz machines. Do not rely on MMX technology to enhance playback. Not all viewers will have MMX machines.
3. Test that the presentation plays to completion successfully (minimal or no buffering after the initial preroll) and that the presentation parts are synchronized.
4. When embedding a presentation in a Web page, verify that the playback window has the correct location and controls.

Chapter 10: Broadcasting Live

Information on preparing content for a live broadcast will be added at a later date. See the documentation for RealNetworks tools products for information on broadcasting RealVideo and RealAudio events with RealNetworks tools. You can get tools and documentation from **http://www.real.com**.

Chapter 11: Using RealMedia Logos

This chapter to be included later.

Appendix A: File Type Reference

The following table provides a quick reference to file types commonly used in RealMedia production. This is not a definitive list of all file types used in RealSystem.

RealSystem File Types

Extension	File Type	Reference
RealMedia Streaming File Types		
.ra	RealAudio	“Producing RealAudio” on page 23
.rm	RealVideo (may include RealAudio track)	“Producing RealVideo” on page 31
.rp	RealPix mark-up	“Images” on page 10
.rt	RealText mark-up	“Text” on page 10
Files Types Also Streamed by RealSystem		
.au	Audio file	“AU” on page 27
.avi	Windows standard video file	“AVI” on page 32
.jpg	JPEG file used in RealPix	“Images” on page 10
.stg	STiNG file used in RealPix	“Images” on page 10
.swf	Shockwave Flash file for RealFlash	“Producing Animation” on page 33
.viv	Vivo video file	“Vivo” on page 32
.wav	Waveform audio file	“WAV” on page 27
RealSystem Information Files		
.ram	RAM file to launch RealPlayer	“Linking your Web Page to your Presentation” on page 75
.rpm	RAM file to launch RealPlayer for an embedded presentation	“Linking your Web Page to your Presentation” on page 75
.smi	SMIL file for putting presentations with multiple file types together	“Assembling a Presentation with SMIL” on page 43
Common Source Files Converted to Streaming File Formats		
.aif	Audio Interchange Format source for RealAudio	“Producing RealAudio” on page 23
.fla	Flash source file for Shockwave Flash	“Producing Animation” on page 33
.mov	QuickTime movie source for RealVideo	“Producing RealVideo” on page 31
.snd	Sound source file for RealAudio	“Producing RealAudio” on page 23

GLOSSARY

B

bandwidth
The upper limit on the amount of data, typically expressed as Kilobits per second (Kbps), that can pass through a network connection per second.

bit
The smallest unit of measure of data in a computer. A bit has a binary value, either “0” or “1.”

bit rate
The number of bits transmitted per second. A 28.8 Kbps modem, for example, can transmit or receive around 29,000 bits per second.

buffering
The process of receiving and storing presentation data before playing it back. RealMedia presentations have an initial buffering time called “preroll”. Once the presentation starts, buffering should be minimal or the presentation may pause.

byte
A common unit of data measurement. One byte is composed of eight (8) bits.

C

client
General term for a software application that receives data from a server. A Web browser is a client of a Web server. RealPlayer is a client of RealServer.

clip
A single file in a presentation. The term typically refers to media files with internal timelines, such as audio or video.

Codec
Compressor/decompressor. Codecs convert between an uncompressed (raw) format and a compressed format such as RealAudio. They reduce the amount of bandwidth a streaming file consumes.

D

download
To send a file over a network so that the entire file must arrive before the file contents display. Contrast to “stream”.

E

encoding
The method of converting and compressing a file into a smaller, streaming format. For example, you use an encoding tool and a Codec to encode sound files as RealAudio.

H

HTTP
The protocol used by Web servers to communicate with Web browsers. RealServer can use HTTP, but it streams files to RealPlayer with RTSP.

I

ISP
Internet Service Provider. A company that provides access to the Internet.

Many ISPs have RealServer available to stream media.

K

Kilobit
A common unit of data measurement equal to 1024 bits. A Kilobit is usually referred to in the context of bit rate per unit of time, such as Kilobits per second (Kbps).

Kilobyte
A common unit of data measurement equal to 1024 bytes.

P

PNA
A proprietary protocol RealServer 6.0 supports for backwards compatibility with RealSystem versions 3.0 through 5.0.

preroll
Buffering that occurs at the start of the presentation. A presentation should have a preroll under 10 seconds.

presentation
A group of clips coordinated through SMIL and streamed from RealServer to RealPlayer.

R

RDP
The proprietary data package protocol RealServer 6.0 uses (along with RTSP) when communicating with RealPlayer 6.0. Contrast to “RTP”.

RealPlayer
RealNetworks client designed to play multimedia presentations streamed by RealServer.

RealServer
RealNetworks software used to stream multimedia presentations to RealPlayer.

real-time

RTP
The open, standards-based data package protocol RealServer 6.0 uses (along with RTSP) to communicate with RTP-based clients. Contrast to “RDP”.

RTSP
The open, standards-based control protocol RealServer 6.0 uses to stream files to RealPlayer 6.0 or any RTP-based client. Contrast to “HTTP”.

S

server
1. A software application such as a Web server or RealServer that sends files over a network.

2. The computer that runs server software.

SMIL
Synchronized Multimedia Integration Language. A mark-up language similar to HTML that describes how and when each clip in a presentation is played.

stream
v. To send a media file over a network so that it begins playing back before all file data has arrived.

n. A flow of a single type of data, measured in Kilobits per second (Kbps). A presentation’s soundtrack is one stream, for example.

U

URL

Universal Resource Locator. A location description that lets a Web browser or RealPlayer receive a file stored on a Web server or RealServer.