RealSystem G2 Production Guide

BETA 1 Release

real

™

RealNetworks

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

http://www.real.com

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 simply 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 multimedia examples you can view with RealPlayer.

How to Use this Manual

This manual tells how to assemble a RealSystem G2 presentation. Although it gives tips on producing great content, the more you know about producing audio, video, and graphics in general, 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 RealSystem G2 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 changes in RealSystem G2.

Chapter 2: Creating a RealSystem Presentation

This chapter explains streaming media and walks you through the steps you take to put together a RealSystem G2 presentation.

Chapter 3: Targeting Bandwidth

This chapter explains how to target a bandwidth connection, an important step for creating a streaming media presentation.

Chapter 4: Producing Audio

This chapter gives you pointers on capturing and digitizing high-quality audio. It then gives you the background you need to encode a RealAudio file.

Chapter 5: Producing Video

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

Chapter 6: Producing Animation

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

Chapter 7: Assembling a Presentation with SMIL

After you create your multimedia files, you write a SMIL 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 8: 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 9: 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 10: 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 11: Broadcasting a Presentation

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

Appendix A: File Type Reference

This appendix provides a quick reference for file types used in RealSystem G2 streaming.

Appendix B: SMIL Language Codes

If you create content in different languages, you use codes in the SMIL file to indicate language choices. This appendix lists the codes you can use.

Conventions in this Manual

The following table explains the conventions used in this manual.

Notational Conventions

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 G2 Resources

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

- *RealText Authoring Guide*

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

- *RealPix Authoring Guide*

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

- *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.

- RealSystem G2 Software Development Kit (SDK)

  The RealSystem G2 SDK lets you integrate applications with RealSystem. You need the SDK and its documentation to create a new plug-in for RealServer or RealPlayer. 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. With RealSystem G2, you have far more possibilities for creating Web-based multimedia than before.

New File Streaming Possibilities

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 and zooms.

Additional Information

See "Choosing Media to Stream" on page 10.

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 G2 users can play new RealSystem streaming file types as soon as they are introduced.

Additional Information

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

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. Note, however, that these file formats typically do not stream over networks as easily and robustly as RealAudio and RealVideo.

Additional Information

See "Choosing Media to Stream" on page 10.

Advancements in RealAudio and RealVideo

New RealAudio SmartStream Codecs Provide Superior Sound Quality

RealSystem G2 introduces a new family of RealAudio SmartStream 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 24 for an overview. "Choosing RealAudio Codecs" on page 31 lists the new SmartStream codecs.

Multiple Audio Streams Play through the Same Codec

RealPlayer G2 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 14. Refer to Chapter 7 beginning on page 51 for instructions on using SMIL.

Bandwidth Negotiation through Multiply Encoded Files or SMIL

RealSystem G2 introduces simpler methods for supporting multiple bandwidth connections. New SmartStream codecs allow a single RealAudio 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 24 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 97.

Enhanced Protocol Support

RTSP Protocol Now Used

Because it still supports the PNA protocol, RealServer G2 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 G2 now begin with `rtsp://`.

RealSystem Interoperates with RTP-Based Servers and Clients

When communicating with RealPlayer G2, RealServer G2 uses RealTime Streaming Protocol (RTSP) as its control protocol and RealNetworks' proprietary RDP as its packet protocol. But because RealSystem G2 also 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.

RealSystem Protocols

Server	Client	Control Protocol	Packet Protocol
RealServer G2	RealPlayer G2	RTSP	RDP
RealServer G2	RTP-based client	RTSP	RTP
RTP-based server	RealPlayer G2	RTSP	RTP
RealServer G2	RealPlayer 3.0 to 5.0	PNA	PNA
RealServer 3.0 to 5.0	RealPlayer G2	PNA	PNA

Compatibility with Previous Releases

RealSystem G2 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 G2.
- 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 RealSystem Presentation

RealSystem G2 gives you the power to create compelling, complex multimedia presentations streamed over a network. This chapter gives you an overview of steps you take to build a streaming media presentation.

Streaming Media with RealSystem G2

RealSystem G2 includes RealServer G2, the most advanced streaming media server available, along with RealPlayer G2 and RealPlayer Plus G2, the world's most popular desktop applications for playing streaming media clips. Before you begin to develop your presentation, download RealPlayer G2 from **http://www.real.com**. Also make sure that your company, organization, or Internet Service Provider (ISP) has RealServer G2 available to stream your presentation.

Tip

With RealPlayer G2 installed, watch some streaming media presentations at **http://www.timecast.com** and **http://www.real.com/showcase**. These presentations demonstrate the many possibilities of streaming media.

Streaming vs. Downloading

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 displaying them. A downloaded video clip, for example, may not begin playback for several minutes. But a streamed video clip begins to play within seconds.

Working with Timelines

A static Web page has no timeline: images and text download without a preset order. The page is complete once all media arrives. A streaming multimedia presentation, though, has a timeline that coordinates events. If you stream a slide show with an audio narration, for example, the audio and images need to stay synchronized to the same timeline.

Although RealSystem ensures that media clips stay synchronized, timelines introduce complexity into streaming multimedia. Persons viewing multimedia over a network have bandwidth constraints, which are limits on how much data can get to them each second. Your presentation may stall if at some point in its timeline it requires more bandwidth than the viewer has available. This happens because RealServer needs to transmit within a certain time frame more data than the viewer's RealPlayer can receive.

When you develop a streaming media presentation, you want to ensure that it flows smoothly to your target audience. When video or audio stalls, your audience is not likely to wait long for it to restart. Also, you want to make sure that the presentation begins to play back quickly. Viewers are rarely willing to wait more than a few seconds for something to happen after they click a link.

Additional Information

Chapter 3 beginning on page 19 explains how to produce streaming multimedia with bandwidth in mind. This is the primary concern for creating a smoothly flowing presentation.

Choosing Media to Stream

RealSystem G2 gives you many choices for combining media into streaming presentations. In choosing which media formats to stream, carefully consider bandwidth requirements as described in Chapter 3 beginning on page 19. This helps you decide, for example, what size to make a streaming video.

Audio

Chapter 4 beginning on page 27 discusses the audio formats you can stream:

- RealAudio
- AU
- WAV

Video

Chapter 5 beginning on page 37 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. See Chapter 6 beginning on page 41 for details.

Images

With RealPix you can stream images to create eye-catching slide shows with special effects such as dissolves and zooms. For more information, download *RealPix Authoring Guide* from **http://www.real.com**.

Text

RealText streams text 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 Authoring 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.

Working with Source Files

After you choose your media types and bandwidth target, you produce your source files with editing tools. Sound editing software, for example, lets you cut parts of an audio clip and optimize the clip for streaming. Producing streaming media for RealSystem does not require you to use specific editing tools. You must simply ensure that your editing tools can save files in streaming formats, or in formats you can easily convert to streaming formats.

Tip

The quality of your source materials affects the outcome of your streaming presentation. See Chapters 4 and 5 for tips on capturing high quality audio and video.

Understanding File Formats

There are three basic types of file formats you work with when developing a streaming presentation:

- Proprietary formats

  Many editing programs save files in a proprietary format used only by that program. Sound Forge sound editing software, for example, uses the proprietary Sonic Foundry format (file extension .sfr). RealSystem cannot stream most proprietary formats.

- Standard or “open” formats

  Editing programs typically let you save or export files to a standard, “open” format. Sound editing programs usually let you export files to the Windows-standard WAV format or the Macintosh-standard AIFF format. RealSystem can stream several standard formats, but these formats are typically not optimized for streaming.

- Streaming formats

  Formats such as RealAudio and RealVideo are highly compressed formats optimized for network streaming. These formats give the best results. You can convert a file from a standard format to a streaming format with an encoding tool. Some editing programs can also directly export files to streaming formats.

Encoding and Compressing Source Files

If your editing program does not export files to the streaming format you want, you can use an encoding tool to convert the file to a streaming format. Streaming formats such as RealVideo are always compressed, so the conversion process automatically compresses the file as well.

RealNetworks Encoding Tools

RealNetworks encoding tools convert many standard file formats to RealAudio and RealVideo. RealNetworks provides free tools for converting popular sound and video formats. It also sells advanced encoding tools that help you create HTML pages and transfer files to servers. 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 worker who puts a tile in each square to create a mosaic. If you are telling the worker 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.

Tips 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 37.

Synchronizing Files to a Timeline

As noted earlier, 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 your SMIL file to specify playback times for each file.

Additional Information

For specifics on SMIL timing, see "Specifying Timing" on page 59.

Writing a SMIL File

With your files in their streaming formats, you put your 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 Using Container Formats

  Because RealSystem can stream many media formats, you do not need to merge files into container formats such as these:

  - Advanced Streaming Format (ASF, file extension `.asf`)
  - RealMedia File Format (RMFF, file extension `.rmf`)

  Although RealSystem G2 can stream either of these container formats, keeping files in their streaming formats and putting the presentation together with a SMIL file gives you greater flexibility. To change your presentation, for example, you simply edit the SMIL file rather than merge 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 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 lay out the clips.

- Include Ads

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

Additional Information

Chapter 7 beginning on page 51 explains the SMIL file syntax. Chapter 9 beginning on page 95 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 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. See Chapter 10 beginning on page 100 for information on linking your Web page to your presentation files on RealServer.

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. For more on playing back a presentation in a Web page, see Chapter 8 beginning on page 79.

Note

If you use an Internet Service Provider (ISP), make sure your ISP has RealServer G2 available. Earlier versions of RealServer, up through RealServer 5.0, do not work with SMIL files and many of the G2 streaming file types.

Viewing your Presentation

With RealPlayer G2 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 playback, see “Testing your Presentation” on page 104.

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 playing streaming media, you can be sure that your RealSystem G2 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: Targeting Bandwidth

Targeting a network connection's bandwidth is crucial for delivering a successful streaming media presentation. Web users with 28.8 Kbps modems, for example, need to view presentations that require less than 28.8 Kilobits of data per second. As the first step in building your presentation, target a bandwidth and create content with that connection speed in mind. This helps ensure that the presentation streams smoothly.

Note

The only time you do not need to consider bandwidth is when all files in your presentation reside on users' local computers rather than on RealServer.

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, 22 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 someone 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 faster than 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 that long.

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.

What is Preroll?

Preroll is the initial data that RealServer sends to RealPlayer before playback begins. Before it delivers a presentation, RealServer looks at the clip sizes and the timeline. Weighing these against the connection bandwidth, RealServer determines how much data RealPlayer must receive starting to play the presentation. The preroll helps ensure that once RealPlayer commences playback, it does not need to halt the presentation while it receives more data.

As a rule of thumb, you want the preroll under 15 seconds, ideally under 10 seconds. If your presentation requires more bandwidth than the user's connection can supply, the presentation can still play back but it requires a lengthy preroll, and users are not likely to watch long while nothing happens on screen. Sticking to your bandwidth target helps keep presentation preroll to an acceptable level.

Choosing a Target Bandwidth

The target bandwidth of a RealSystem G2 presentation is the maximum bandwidth available for a particular connection, such as a 28.8 Kbps modem. The presentation's total bit rate must be at or below the target bit rate. The total bit rate comprises two main parts:

- Maximum bit rate consumed by all streaming files. For a multifile presentation, this can vary over time. You therefore want to consider what part of your presentation consumes the most bandwidth and use that peak point as your guideline.

- 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 various network connections.

Bit Rates Available for Streaming Files

Target Connection Speed	Recommended 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. Things become more complex, though, when you combine different clips into one presentation. But with some practice, you will quickly learn how to balance bandwidth requirements with presentation quality.

Understanding Clip Bandwidth Characteristics

The first step in planning your multimedia presentation is to understand the bandwidth characteristics of your clip or clips. The following sections describe characteristics for RealMedia file types. If you stream other types of clips, make sure you understand each clip's bandwidth characteristics.

RealAudio and RealVideo

RealAudio and RealVideo consume bandwidth at a flat rate. The codec used to create the file determines the bandwidth consumption. If you have a RealAudio file encoded with a 8 Kbps codec, for example, that file will steadily consume 8 Kbps of bandwidth as long as it plays.

Additional Information

"Choosing RealAudio Codecs" on page 31 and "Using RealVideo Codecs" on page 39.

RealFlash

The Shockwave Flash component of a RealFlash presentation has a spiky bandwidth characteristic, meaning it consumes a lot of bandwidth at certain points in its timeline, little bandwidth at other points. RealNetworks provides tools that help you optimize RealFlash for a specific bandwidth.

Additional Information

If you plan to create streaming animation, read "Preparing a RealFlash Presentation" on page 43.

RealPix

RealPix consists of still images streamed to RealPlayer. Its bandwidth consumption depends on the number and size of the images, as well as how quickly you stream them. RealPix therefore gives you a lot of control over bandwidth usage through your choice of images and how you construct the RealPix timeline.

Additional Information

For more on RealPix and its bandwidth characteristics, refer to *RealPix Authoring Guide* available at **http://www.real.com**.

RealText

RealText consists of a text file that contains the RealText mark-up. Because it uses just a simple text file, RealText consumes little bandwidth. This makes it easy to add RealText to any presentation.

Additional Information

For information on creating RealText, see *RealText Authoring Guide* available at **http://www.real.com**.

Delivering a Single Media Clip

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. However, when a file has multiple streams, such as a video that contains a visual track and an audio track, you need to consider how much bandwidth goes to each stream.

Suppose you want to stream a RealVideo clip at 28.8 Kbps. How much bandwidth should you give to the visual track and how much 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 even when you stream just a single file, your bandwidth target 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.

Additional Information

See "Shooting a Video" on page 37.

Developing Multimedia Presentations

An exciting part of RealSystem G2 is that you can develop multimedia presentations, such as slide shows with audio voice-overs, or video with scrolling subtitles. When multiple clips play together, you need to consider how much presentation bandwidth to allot each file. While it's exciting to create presentations that include video, slide shows, audio, and scrolling text all at once, viewing such presentations over slow modems may be difficult.

When developing a streaming multimedia presentation, keep the following tips in mind:

- Consider the presentation timeline carefully to eliminate bandwidth bottlenecks. These typically occur when two or more high-bandwidth clips play simultaneously. You may need to omit high-bandwidth pairings, combining high-bandwidth with low-bandwidth clips instead.
- Stagger the times when clips begin to play back. Every clip requires a certain preroll before RealPlayer can play it. Your presentation will play more smoothly if RealServer does not need to send more than one clip's preroll at a time.
- Start presentations with low-bandwidth clips. Use RealText to display credits, for example. Or begin with a highly compressed RealAudio narration before bringing in video. RealSystem takes advantage of the extra bandwidth to begin streaming higher bandwidth data to RealPlayer "behind the scenes."
- Test presentations in "real world" circumstances, replicating your audience's bandwidth conditions. Clips may play back fine when the files are on your desktop computer. The presentation may bog down, though, when you stream the clips over a modem.

Supporting Multiple Bandwidth Connections

To reach a wide audience, you need to provide content available for slow connections. If you're publishing a RealAudio file, for example, you can encode it at 8 Kbps so that anyone with a 14.4 Kbps or higher connection can play it. This file will have good quality sound. But the same source file encoded for 16 Kbps will have better sound. Encoded for 32 Kbps delivery, 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 the RealAudio SmartStream codecs, 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.

Additional Information

For more on SmartStream, see “Choosing RealAudio Codecs” on page 31.

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 presentation, you use a 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 decrease the presentation bandwidth temporarily. When the congestion clears up, it continues to stream all the presentation data.

Additional Information

“Setting Bandwidth Choices” on page 62 explains how to use a SMIL file to designate different bandwidth groups.

CHAPTER 4: Producing Audio

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 higher speeds. RealSystem G2 can stream other audio formats as well. This chapter explains how to prepare and encode your sound files. It also provides tips for capturing high quality audio.

Steps for Streaming RealAudio

To create a streaming RealAudio file, you follow these basic steps:

1. Capture audio.

   Unless you start with a digitized audio source file, you capture audio from a voice or music source, such as a person speaking into a microphone or a CD you play through your computer.

   Additional Information
   "Recording Audio" on page 28 provides guidelines for capturing audio.

2. Digitize and optimize audio.

   You next digitize the source audio to a common file format, such as WAV or AIFF. With a sound editor, you can then optimize the audio for streaming. (If you are broadcasting live, however, you encode the streaming audio directly from the source.)

   Additional Information
   See "Editing Digitized Audio Files" on page 30 for tips on sound editing.

3. Encode to RealAudio.

   With you digitized file optimized or your live broadcast ready to go, you encode your source in the RealAudio format. When you do this, you choose a codec or set of codecs that target a network bandwidth.

   Additional Information

   "Producing RealAudio" on page 31 explains RealAudio and its codecs.

4. Deliver RealAudio.

   With your presentation ready to go, you make your RealAudio file or broadcast available through your Website. If you are combining sound with another streaming file type, such as RealPix, you write a SMIL file that assembles all the pieces.

   Additional Information

   Chapter 7 starting on page 51 explains how to create a SMIL file. See Chapter 10 beginning on page 97 for instructions on linking your Web page to a RealAudio file. For more on live broadcasting, read Chapter 11 starting on page 107.

Recording Audio

Whether you start with existing recordings or capture audio live, the following tips will help you create great audio source files or broadcasts.

Use High Quality Source Media

If you plan to stream existing material, start with the best source possible. Use the cleanest recording with the least amount of unwanted noise. Compact disc (CD) and digital audio tape (DAT) are good source media, although well-recorded analog sources such as records, reel-to-reel tapes, and chrome (type II) cassettes can sound just as good. Try to avoid "consumer grade" recording media such as Type I cassettes and VHS tapes.

When capturing audio, record to high quality source media such as CD or DAT. Alternately, you can capture audio directly to a WAV or AIFF file through your computer's sound card.

Choose Professional Recording Equipment

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

Use Shielded Cables

It is important to use high quality, shielded cables. Unshielded cables increase the chance of introducing line noise and Radio Frequency Interference (RFI) into recordings. Keep audio cables physically separated from power cords to minimize the introduction of noise. Also be sure to ground all equipment properly.

Set Input Levels Correctly

Setting correct input levels is crucial. All audio equipment has a signal-to-noise ratio, the ratio between the loudest possible sound the equipment can reproduce without distortion and its inherent noise floor. This distortion is known as “clipping,” and is audible as a high-frequency crackling noise.

To get the best signal-to-noise ratio, set the input level on each audio device you use so that it utilizes its full range of available amplitude without distortion during the piece’s loudest sections. For example, the audio signal chain may start with the microphone, continue through a small mixing desk, go to a compressor, and finally end at the sound card. For each piece of equipment, set levels as close as possible to 0 dB without going over.

When digitizing with a sound card, attenuate each device to work at its peak efficiency, checking at each point for signal distortion. Perform several test runs and note input levels in your sound editor. Make sure there are no peaks above maximum amplitude. Adjust levels with your sound card mixer so the input approaches but does not exceed the maximum. Be conservative, though. Levels might suddenly increase if, for instance, an interviewee suddenly speaks loudly or a crowd at a sports event roars.

Prepare Volume Levels for Live Broadcasts

When broadcasting live audio, prepare and test volume levels before encoding live input. It is beneficial to have a dynamics compressor (gain compression, not data compression), which is an outboard piece of audio equipment that automatically “rides” the gain level. It maximizes input levels and reduces the chance of distortion by decreasing signal gain that rises above a specified threshold.

Use Optimum RealAudio Sampling Rates

RealAudio codecs have optimum sampling rates that produce the best sound. The “RealAudio Codecs” table starting on page 32 lists the recommended sampling rates for each codec.

Editing Digitized Audio Files

If you are not broadcasting audio live, capture and “digitize” the audio to a supported file format such as a WAV, QuickTime, or AIFF. Digitizing the audio before encoding a streaming file lets you edit the audio and maximize its available dynamic range. The following sections give some basic sound editing and optimization tips for digitized audio.

Eliminate DC Offset

DC Offset is extremely low frequency (inaudible) noise that results from equipment grounding problems. If you don’t remove it, it can skew the results of subsequent sound editing. Use your sound editor’s **DC Offset** function immediately after recording a digital audio file.

Tip

If your sound editing program allows it, eliminate DC offset during recording. This saves you an editing step.

Normalize Audio Files

Set sensible input levels when recording, then use normalization to maximize levels after recording. Your streaming files sound best when your digitized source has the highest possible gain without clipping. Digital audio files that do not utilize their full amplitude range produce low quality streaming files. If the amplitude range is too low, use your sound editor to adjust the range and increase the amplitude.

Tip

Most sound editors have a **Normalize** function that maximizes levels automatically.

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 certain frequencies and cut frequencies that contain noise or unwanted sound. 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). You can therefore use EQ to make a RealAudio file sound as close as possible to the initial recording.

Producing RealAudio

RealAudio is a compressed format suitable for streaming over the Internet or intranets. RealAudio files use the file extension `.ra`. 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`)

Choosing RealAudio Codecs

RealAudio uses a “lossy” compression scheme that discards parts of the audio source file to achieve a highly reduced file size. A RealAudio file encoded from a WAV source file, for example, may be 10 to 20 times smaller than the WAV file. Although discarding audio information during encoding lowers the file’s frequency response and dynamic range, carefully choosing codecs minimizes the impact of compression.

A RealAudio 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 Chapter 3 beginning on page 19 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

RealAudio uses different codecs for music and spoken voice. Voice codecs focus on the standard frequency range of the human voice. Music codecs have broader frequency response to capture more of the high and low frequencies.

The following table provides a reference for all RealAudio codecs. Note that your encoding tool may not include all codecs listed. The table provides the following information:

- G2, 5, 4, 3, 2, 1

  An “X” in these columns indicates that a file encoded with this codec can be played by RealPlayer G2, 5.0, 4.0, and so on. You can encode a single file for multiple bandwidths only with the SmartStream codecs, which are playable only by RealPlayer G2.

- Sample Rate

  You can use any sampling rate in your audio source file. However, the suggested sampling rates in the table below 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.

RealAudio Codecs

RealAudio Codec	G2	5	4	3	2	1	Sample Rate	Resp.	Comments
Low Bandwidth Codecs									
5 Kbps Voice	X	X	–	–	–	–	8, 16, or 32 kHz	4 kHz	Lowest bit rate codec for speech or speech with background music.
6.5 Kbps Voice	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	Low bit rate codec for speech or speech with background music.
8 Kbps Voice	X	X	X	X	X	X	8, 16, or 32 kHz	4 kHz	Original voice codec. Superseded by 8.5Kbps Voice.
8 Kbps Music–G2 SmartStream	X	–	–	–	–	–	8, 16, or 32 kHz	4 kHz	SmartStream 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	High quality voice codec for voice or voice with background music.

(Table Page 1 of 3)

RealAudio Codecs (continued)

RealAudio Codec	G2	5	4	3	2	1	Sample Rate	Resp.	Comments
11 Kbps Music–G2 SmartStream	X	–	–	–	–	–	11.025, 22.05, or 44.1 kHz	5 kHz	SmartStream codec. Use with multiply encoded files.
12 Kbps Music	X	X	X	–	–	–	8, 16, or 32 kHz	4 kHz	DolbyNet codec.
Medium Bandwidth 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	High quality, high bit rate codec for voice or voice with background music.
16 Kbps Music–G2 SmartStream Mono	X	–	–	–	–	–	22.05 or 44.1 kHz	8 kHz	SmartStream 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 SmartStream Mono	X	–	–	–	–	–	22.05 or 44.1 kHz	10 kHz	SmartStream 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 Codecs									
32 Kbps Music–G2 SmartStream Mono	X	–	–	–	–	–	44.1 kHz	16 kHz	SmartStream 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.

(Table Page 2 of 3)

RealAudio Codecs (continued)

RealAudio Codec	G2	5	4	3	2	1	Sample Rate	Resp.	Comments
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 3 of 3)

Encoding RealAudio with RealSystem Tools

When you encode RealAudio files with a RealSystem G2 encoding tool, you simply set parameters such as audio type (voice or music) and compatibility with earlier versions of RealPlayer. You can also specify multiple bandwidth targets for the file, such as both 28.8 Kbps modems and ISDN connections. The tool then chooses the best codec or codecs to use. The following sections give tips on using RealSystem tools.

Additional Information

See the tool's manual or online help for step-by-step instructions on encoding RealAudio. RealSystem encoding tools are available for purchase or free download at **http://www.real.com**.

Using RealAudio in a MultiFile Presentation

When you encode a RealAudio file, consider whether it will play in parallel with another clip such as a RealPix. If you target 28.8 Kbps modems when encoding, for example, the tool may select a 20 Kbps codec, leaving no bandwidth for the second clip. Make sure you specify that the RealAudio file is just one part of the presentation. The tool then lets you choose a lower bandwidth codec, such as 8 or 12 Kbps.

Multiple Encoding in a Single File

You can create a single file encoded for multiple bandwidths only with the SmartStream codecs introduced in RealSystem G2. In the "RealAudio Codecs" table starting on page 32, these codecs are marked as playable only by RealPlayer G2. To support multiple bandwidths with other codecs, you must encode a separate file with 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 62.

Backwards Compatibility with Earlier Versions of RealPlayer

When you use a RealSystem 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 SmartStream 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 SmartStream codecs. In the RealSystem encoding tool, you choose backwards compatibility to create an additional 8 Kbps stream with an older codec. Depending on its connection speed, RealPlayer G2 receives the SmartStream 8, 16, or 32 Kbps 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 5: Producing Video

This chapter covers the types of video that RealSystem G2 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 Audio" on page 28.

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 31.

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 12 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 "Choosing RealAudio Codecs" on page 31 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 6: Producing Animation

RealFlash makes it easy to put animation on the Web. Combining the power of Macromedia Flash with the clarity of RealNetwork's RealAudio, RealFlash produces visually arresting animations with superb sound. This chapter tells how to create RealFlash content for different bandwidths. It also provides tips for streaming RealFlash presentations.

Additional Information

For exciting examples of streaming animation, visit the RealFlash showcase at **http://www.real.com/showcase/animation/index.html**.

Introduction to RealFlash

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,
- and Karaoke.

Tools for Creating RealFlash

You need the following tools to create and stream RealFlash:

- Macromedia Flash

  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.

Additional Information

For information on Flash, visit **http://www.macromedia.com**.

Warning

RealFlash supports Flash 2.0 but does not fully support the features introduced in Flash 3.0, such as morphing and transparency. If you are using Flash 3.0, limit your RealFlash presentation to the 2.0 feature set. Support for Flash 3.0 features will be added at a later date.

- Sound capture and editing tools

  You should use professional quality hardware and software to capture and process the sound file you will encode as RealAudio.

  Additional Information

  For more on audio production, see Chapter 4 beginning on page 27.

- 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.

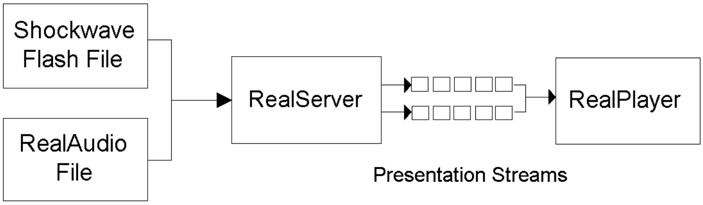
- RealServer and RealPlayer

  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 clip 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 Chapter 3 beginning on page 19.

The good news is that designing Realflash content for a low bandwidth does not diminish the quality of your animation. RealFlash transmits vector information that the viewer's machine then renders. So unlike bitmap animations, 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.

Recommended Bandwidth Divisions between RealAudio and Flash at 20Kbps

Presentation Type	RealAudio	Flash
Emphasis on animation with good quality spoken soundtrack	5 Kbps Voice	15 Kbps
Emphasis on animation with high quality spoken soundtrack	6.5 Kbps Voice	13.5 Kbps
Emphasis on animation with very high quality spoken soundtrack	8.5 Kbps Voice	11.5 Kbps
Emphasis on animation with good quality music soundtrack	8 Kbps Music–G2 SmartStream	12 Kbps
Emphasis on high quality music soundtrack with animation	11 Kbps Music–G2 SmartStream	9 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.

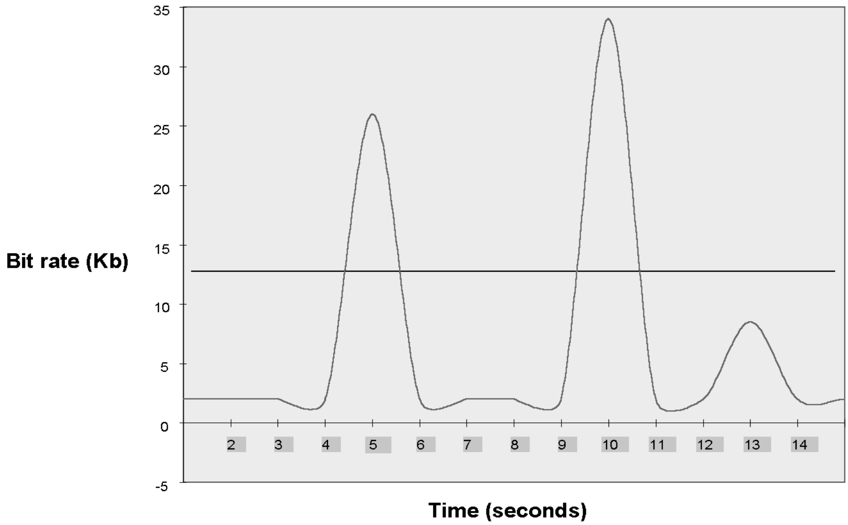
Additional Information

See “Choosing RealAudio Codecs” on page 31 for a full 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 streamed. 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 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, Flash 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 RealFlash CPU requirements:

- Set a frame rate of seven (7) fps for 28.8 Kbps connections. This 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 the Shockwave commands listed below. 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 an internal RealPlayer command that displays the URL in the browser window.

Shockwave and RealPlayer Interactive Commands

Shockwave Command	RealPlayer Mapping	Action
`Play`	`Play`	Playback begins or resumes.
`Stop`	`Pause`	Presentation pauses until action is performed or Play button pressed.
`Goto`	`Seek and Pause`	RealPlayer seeks to the designated frame, buffers the presentation preroll, and pauses.
`Goto and Play`	`Seek and Play`	RealPlayer seeks to the designated frame, buffers the presentation preroll, and begins playback.
`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.

Tip

Because seeking requires buffering, do not use Flash `Goto` commands to advance from one scene to the next. When you export your Flash animation to Shockwave Flash, scenes are concatenated so that the animation flows automatically from one scene to the next.

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 G2 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 RealAudio Codecs” on page 31 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 7 beginning on page 51 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 10 beginning on page 97.

CHAPTER 7: 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. After writing the SMIL file, you put it on RealServer and link your Web page to it as described in Chapter 10 beginning on page 97. A Web user then clicks the SMIL file link to view your presentation.

Tip

> If you have just one file in your presentation, such as a single RealVideo file, you don't need to create a SMIL file. Just link your Web page to the media file. See Chapter 10 for more information.

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. 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 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`.
- You need to use codes to add quotation marks, apostrophes, ampersands, or angle brackets to a SMIL header. See "Using Coded Characters" on page 53.
- 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 SMIL functions, though.

Adding a Header

The SMIL file can have a header section that defines aspects of the entire presentation:

```
<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 (for more on source tags, see "Specifying File Locations" on page 54):

```
<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. For more on these tags, see "Organizing a Presentation" on page 57.

Using Coded Characters

In a header, SMIL interprets quotation marks, apostrophes, ampersands, and angle brackets as syntax markers. To have these characters show up as text in RealPlayer, you use codes in the header. As shown in the following table, codes begin with an ampersand (&) and end with a semicolon (;). SMIL interprets these characters the same way as popular Web browsers.

SMIL Coded Characters

Code	Character	Example
`&quot;`	quotation mark	"
`&amp;`	ampersand	&
`&apos;`	apostrophe	’
`&lt;`	left angle bracket (”less than” sign)	<
`&gt;`	right angle bracket (”greater than” sign)	>

For example, to add the following as a title:

```
 ”Multimedia’s <smil> & you”
```

You enter this in the SMIL file header:

```
<meta name=”title” content=
 ”&quot;Multimedia&apos;s &lt;smil&gt; &amp; you&quot;”/>
```

Specifying File Locations

To specify a presentation file, you add to the SMIL body section a tag that describes the file type and location:

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

The tag begins with one of the file type attributes listed in the following table.

Source File Attributes

File Attribute	Used For
`animation`	Animation files, such as Shockwave Flash files used in a RealFlash presentation.
`audio`	Audio files.
`img`	Still images in JPEG or STiNG format. RealPlayer does not currently support the rendering of GIFs. For more on STiNG, visit **http://www.iterated.com/**.
`ref`	Any file type not covered by other attributes.
`text`	Static text files.
`textstream`	Streaming RealText files.
`video`	Video or other files that display continuous motion.

Although a source file tag must start with a file type attribute, 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.

After the file type attribute, the src attribute lists the file location. How you specify this location depends on whether you will stream the presentation over a network with RealServer or place the presentation files on a local computer.

Linking to Files on RealServer

When a RealSystem G2 presentation streams over a network, the media files reside on RealServer. Each source file's src attribute gives the file's URL:

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

The following table explains the URL components in the example above. Contact your RealServer administrator to get the RealServer address, RTSP port, and directory structure.

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. It typically uses an identifier such as realserver instead of www. Or it may use a TCP/IP address such as 172.2.16.230 instead of a name.
: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/	This is the directory that holds the file. The directory structure may be more than one level deep as shown in this example.
first.ra	This is the file name.

Creating 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, so you can use directory notation such as "../". 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 machines.

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.

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

Use the <seq> tag to play clips in sequence. 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 most commonly combined with <par> to create combinations of sequential and parallel clips.

Playing Files in Parallel

You can play two or more clips at the same time through the <par> (“parallel”) tag. For example, the following combines a RealVideo clip with a RealText clip:

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

RealSystem ensures that clips stay synchronized. If some video frames don’t arrive, for example, RealSystem either drops those frames or halts the presentation playback until the frames arrive.

Additional Information

When multiple display clips such as RealVideo and RealText play in parallel, you also need to define each clip's playback region. For more information, see "Laying Out Multiple Clips" on page 64.

Combining <seq> and <par> Tags

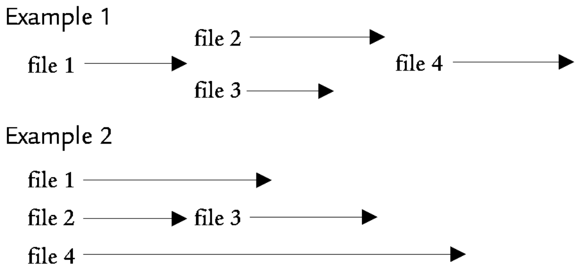
You can combine and nest `<seq>` and `<par>` tags as needed. Note that the organization of these tags greatly affects the presentation playback.

```
<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.



Specifying Timing

Timing elements let you specify when a clip 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.

Timing Shorthand Examples

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 a "normal play time" 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 and End Times

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

```
<video src="videos/newsong.rm" begin="20.5s"/>
```

If this clip is in a `<par>` group, this `begin` attribute starts the clip 20.5 seconds after the group becomes active. If the clip is in a `<seq>` group, the attribute inserts 20.5 seconds of blank time before the clip starts to play back. The `begin` attribute works for images, audio, video, or any other file type.

Additionally, you can set an end time that specifies when the clip ends within the presentation timeline. You can use the `end` attribute alone or combined with a `begin` attribute as shown here:

```
<video src="videos/newsong.rm" begin="20.5s" end="62.7s"/>
```

Setting a Fill

The `fill` attribute determines what happens when the end 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 end 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. This has no effect on audio.

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

```
<video src="videos/newsong.rm" end="30s" fill="freeze"/>
```

After the video plays, its last frame displays for 10 seconds. The video disappears when the end time elapses.

Setting Internal Clip Begin and End Times

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

```
<video src="videos/newsong.rm" 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 playback when it reaches its 50.7-second mark, having played for 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.

Switching Between Alternate Choices

With the `<switch>` tag, you can specify multiple options that RealPlayer can choose between. The `<switch>` group specifies any number of choices:

```
<switch>
  <choice1 test-attribute=value1/>
  <choice2 test-attribute=value2/>
  ...
</switch>
```

RealPlayer looks at choices in order, evaluating the test attribute and its value to determine which file to choose.

Setting Language Choices

When the `<switch>` group test attribute is `system-language`, each source file is for a different language. The following example shows a video slide show with separate audio narrations in English, French, and German. Based on its language preference setting and the system-language code in the SMIL file, RealPlayer selects a file to play:

```
<par>
  <video src="slides/seattle.rm"/>
  <!-- select audio based on RealPlayer language preference setting -->
  <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>
```

Additional Information

Appendix B starting on page 115 lists the system-language codes such as "en" you use to designate content in different languages.

Setting Bandwidth Choices

To take advantage of high bandwidth connections, 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 can 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 bandwidth 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 ensure that the last option satisfies the lowest bandwidth 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 Files

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

- When the presentation consists solely of a SmartStream clip, simply link to that clip within the SMIL file. The clip then streams at the rate appropriate for RealPlayer's connection speed. You do not need to specify bandwidth choices with a <switch> group.
- Use the <switch> tag when combining a SmartStream clip with other clips encoded for single bandwidths. The SmartStream clip is always used, but the <switch> group gives RealPlayer options for other files. The following example illustrates a RealAudio SmartStream clip and a choice between two RealPix presentations built for different bandwidths:

```
<par>
  <audio src="audio/newsong2.ra"/>
  <switch>
    <ref src="image/slideshow1.rp" system-bitrate="47000"/>
    <ref src="image/slideshow2.rp" system-bitrate="20000"/>
  </switch>
</par>
```

Additional Information

For more on SmartStream codecs, see "Choosing RealAudio Codecs" on page 31. Refer to "Supporting Multiple Bandwidth Connections" on page 24 for more on developing presentations for multiple bandwidths.

Laying Out Multiple Clips

If your presentation plays only one clip at a time, you do not need to create 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 called “regions” within the main RealPlayer window:

1. In the SMIL file header, you create a `<layout>` group and use `<region>` tags to name each playback region and define its size and location within RealPlayer. See “Defining the Layout” below.
2. In the SMIL file body, you use `region` attributes to specify which source files play in which regions. See “Assigning Clips to Regions” on page 69.

Additional Information

See “Working with SMIL Layouts” on page 90 for instructions on using SMIL and RealPlayer’s Netscape plug-in or ActiveX control to lay out your presentation in a Web page instead of in RealPlayer.

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. Measurements are in pixels or percentages, with zero pixels as the default. The following example defines a root-layout region that sets the overall window size. It also defines two regions for displaying video and text:

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

Setting the Background Region

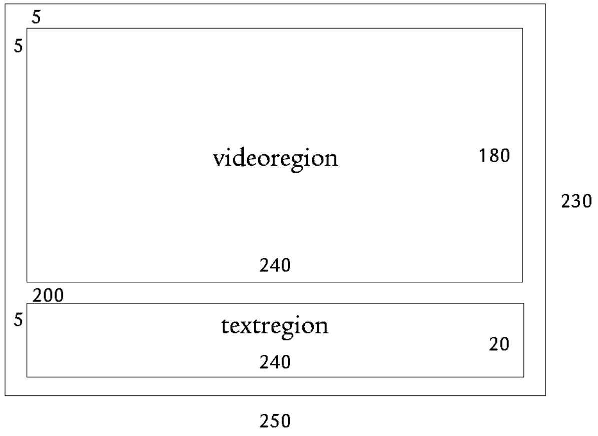
With `root-layout`, you specify the size of the entire playback area in pixels. You cannot play media files in the root-layout region, however. The example shown above creates a root-layout region 230 pixels high by 250 pixels wide. When the presentation begins, the RealPlayer window expands to this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

Tip

You can omit root-layout to have RealPlayer calculate the playback area size based on the pixel sizes and offsets of the other defined regions. But to avoid unexpected results, always define root-layout.

Defining Playback Regions

You create playback regions for media files with <region> tags. These regions must lay within the root-layout region. Any part of a region that lays outside the root-layout region is cut off. The example tags above define two regions named "videoregion." and "textregion." Both regions are offset 5 pixels to the right of the root-layout region's left edge. The video region displays 5 pixels down from the top of the root-layout region, and the text region displays 200 pixels down. The following figure illustrates this placement.



Each region must specify a height and width. For a region's height, width, and offset measurements, you can also percentages that reflect a fraction of the root-layout region's size The following example uses percentages to define playback areas similar to the those shown in the sample above:

```
<head>
  <layout>
    <root-layout background-color="maroon" width="250" height="230"/>
    <region id="videoregion" top="2%" left="2%" width="96%" height="78%"/>
    <region id="textregion" top="80%" left="2%" width="96%" height="18%"/>
  </layout>
</head>
```

Tips for Defining Regions with Percentage Values

Note the following when using percentage values to define regions:

- You must include a root-layout region defined in pixels when specifying region measurements in percentages.
- You can mix pixels and percentages. You could define the `top` and `left` offset measurements in percentages, for example, while specifying the `width` and `height` measurements in pixels.
- You can use whole and decimal values for percentages. For example, the values "4%" and "4.5%" are both valid.
- Because a region is clipped at the boundary of the root-layout region, no percentage value can effectively be more than 100%.

Specifying How a Clip Fits a Region

Note

For this release, only default values for the fit attribute are supported.

When a media clip is encoded at a size different from the playback region's defined size, the fit attribute determines how the clip fits the region:

```
<region id="videoregion" width="128" height="64" fit="meet"/>
```

The fit attribute uses one of the values described in the following table. If you do not specify a fit attribute, the clip uses its default value. Different types of clips may have different default values, however, depending on how well the clip can scale. Default values are typically fill or hidden. In no case will media display outside the playback region's boundaries.

Region Fit Attributes

Attribute	Action
fill	Scale the clip's height and width so that it fills the region exactly.
hidden	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.

Tip

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

Note

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

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

```
<layout>
  <root-layout 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 RGB 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)

Ordering Overlapping Clips

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

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

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

Tips for Defining Z-Index Values

The following are points to observe when using z-index:

- The root-layout region is always behind all other regions and does not use z-index.
- The z-index values start at 0 (zero) and proceed through the positive integers without limit. You cannot use a negative value such as -4.
- The default value of 0 (zero) applies if you don’t specify 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 values. Side-by-side videos can both have z-index="3", for example.
- When overlapping clips have the same value, the clip that starts later in the presentation appears in front.

Assigning Clips to Regions

After you define the layout in the header section, you use region attributes within 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>
```

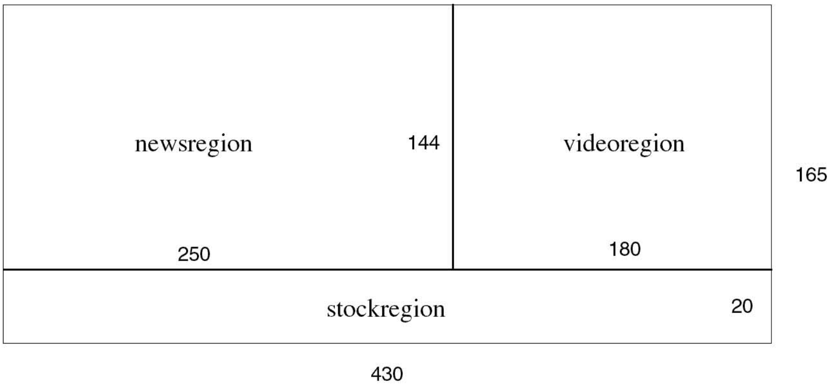
In this example, the video and text files are assigned to the video and text regions defined in the header. You can reuse regions by assigning sequential clips to them. For example, you can play a video clip in a region, then display another clip in that region after the first clip finishes. Don't assign 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 region.

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 RealPlayer display 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 width="430" height="165"/>
      <region id="newsregion" top="0" left="0" width="250" height="144"/>
      <region id="videoregion" top="0" left="250" width="180" height="144"/>
      <region id="stockregion" top="145" left="0" width="430" height="20"/>
    </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 illustrates the design of these regions.



Linking to Other Media

A SMIL file can define links to other media. A video might link to a second video, for example. When the viewer clicks the link, the second video replaces the first. Or the video could link to an HTML page that opens in the viewer's browser. You can even define areas as hot spots with links that vary over time. The bottom corner of a video can link to a different URL every ten seconds, for instance.

Note

Some media files can also define hyperlinks. A RealText file, for example, can define hyperlinks for portions of text. When a viewer clicks an area where a media file link and a SMIL file link overlap, the SMIL link is used.

Making a Source File a Link

The simplest type of link connects a whole media source file to another file. As in HTML, you define the link with <a> and </a> tags. But whereas you enclose text between <a> and </a> in HTML, you enclose a media source tag between <a> and </a> in SMIL:

```
<a href="rtsp://realserver.company.com/video2.rm">
  <video src="video.rm" region="videoregion"/>
</a>
```

The example above links the source file video.rm to the target file video2.rm. When a viewer moves the cursor over the source file as it plays, the cursor turns to a hand icon to indicate that the file is a link. When the viewer clicks video.rm as it plays, video2.rm replaces it. The URL begins with rtsp:// if the linked file streams to RealPlayer from RealServer, or http:// if the file downloads to the browser from a Web server.

Additional Information

For information on RTSP URLs, see "Linking to Files on RealServer" on page 55. For more on URLs using HTTP, see "Linking your Web Page to your Presentation" on page 97.

Targeting RealPlayer or a Browser

An <a> tag or <anchor> tag (see "Defining Hot Spot Links" on page 72) can include a show attribute that determines where a linked file displays:

```
<a href="rtsp://realserver.company.com/video2.rm" show="pause">
  <video src="video.rm" region="videoregion"/>
</a>
```

replace (default)

The attribute show="replace" causes the linked file to replace the source file in RealPlayer. This default behavior also occurs if you do not include the show attribute in the link. The following are important differences between RealPlayer and Web browsers to keep in mind when creating links:

- RealPlayer does not include a **Back** button that allows the viewer to return to the link source file after clicking the link.
- Only one instance of RealPlayer can run at a time. You therefore cannot open a SMIL link in a new RealPlayer window the way you can open an HTML link in a new browser window.

- Clicking the link removes any existing regions. If you have three regions defined and the viewer clicks a link in one region, for example, the target file replaces all media files in all regions. You can preserve regions by targeting a SMIL file that defines the same set of regions. You cannot preserve the timeline positions of files playing in those regions when the viewer clicks the link, however.

Additional Information

"Linking to a SMIL File" on page 75.

new, pause

The values new and pause both open the linked file in the viewer's default browser. The source file continues to play in RealPlayer if you use show="new". With show="pause", the source file pauses in RealPlayer. The viewer can restart playback at any time, though, by clicking RealPlayer's **Play** button.

Use either show="new" or show="pause" to open a Web page or another file viewable within a browser. You can use these attributes to link a RealSystem presentation to your home page, for example. Do not use them to link to another media file played in RealPlayer, however, such as a SMIL file or a RealVideo file.

Defining Hot Spot Links

Within a SMIL file you can define hot spots using an <anchor> tag. Whereas the <a> tag turns the entire media source file into a link, the <anchor> tag turns only a defined area into a link. With <anchor> tags you can create links similar to those in HTML image maps. But SMIL links can be temporal as well as spatial. A link might be valid for just ten seconds during a source file's timeline, for instance.

Setting an Anchor

The <anchor> tag differs from the <a> tag in that you place it within the media source tag rather than before it:

```
<video src="video.rm" region="videoregion">
  <anchor href="rtsp://realserver.company.com/video2.rm" .../>
</video>
```

An <anchor> tag ends with a closing slash. But the media source tag does not end with a closing slash as it normally would. Instead, the source tag and its subsequent <anchor> tags are followed by a closing source tag, such as </video>.

The `<anchor>` tag includes an `href` attribute that typically uses `rtsp://` if the linked file streams to RealPlayer from RealServer, or `http://` if the file downloads to the browser from a Web server

Additional Information

For information on RTSP URLs, see "Linking to Files on RealServer" on page 55. For more on URLs using HTTP, see "Linking your Web Page to your Presentation" on page 97. To target a browser with a link, see "Targeting RealPlayer or a Browser" on page 71.

Defining Spatial Coordinates

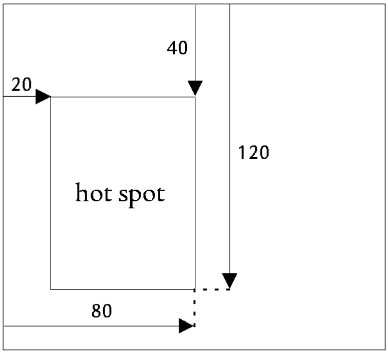
The `<anchor>` tag's `coords` attribute defines spatial coordinates for the hot spot rectangle. Coordinate values in pixels or percentages define the rectangle's offset from the upper, left-hand corner of the media source file as shown in this example:

```
<video src="video.rm" region="videoregion">
  <anchor href="..." coords="20,40,80,120"/>
</video>
```

The coordinate values for the hot spot rectangle follow this order:

1. left side pixel or percentage value
2. top pixel or percentage value
3. right side pixel or percentage value
4. bottom pixel or percentage value

The sample above uses pixel values to define a hot spot 60 pixels wide (80 pixels minus 20 pixels) 80 pixels high (120 pixels minus 40 pixels). It produces a hot spot like that shown in the following figure:



coords="20,40,80,120"

Percentage Values

The `coords` attribute can also use percentage values to create hot spots relative to the source file's size. The following sample places in the center of the source file a hot spot that is a quarter the size of the source file:

```
<video src="video.rm" region="videoregion">
  <anchor href="..." coords="25%,25%,75%,75%"/>
</video>
```

The following table lists sample percentage coordinates that define hot spots for a media source file. Each hot spot is a quarter the size of the source file.

Sample Hot Spot Percentage Coordinates

Hot Spot Rectangle Position	Coordinate Attribute
Upper, left-hand quadrant	`coords="0,0,50%,50%"`
Upper, right-hand quadrant	`coords="50%,0,100%,50%"`
Lower, left-hand quadrant	`coords="0,50%,50%,100%"`
Lower, right-hand quadrant	`coords="50%,50%,100%,100%"`
Center	`coords="25%,25%,75%,75%"`

Tips for Defining Anchor Coordinates

Note the following when defining hot spots:

- You can mix pixels and percentages. For example, the coordinates "50,50,100%,100%" place the hot spot's left and top boundaries in and down 50 pixels from the source file's upper, left-hand corner, respectively. But the hot spot's right and bottom boundaries extend to the source file's right and bottom edges, respectively, no matter the source file's size.

- Values such as “30,30,10,10” are ignored. Here, the hot spot’s left side is defined as farther to the right than its right side. As well, the top is defined to be below the bottom.
- You can use whole and decimal values for percentages. For example, the values “4%” and “4.5%” are both valid.
- A hot spot defined to extend beyond the media source file is clipped at the file’s edge. For example, if the hot spot has coordinates “50,50,300,300” but the source file is 200 by 200 pixels, the hot spot’s effective coordinates are “50,50,200,200”. For this reason, no percentage value can effectively be more than 100%.

Setting Temporal Coordinates

In addition to defining spatial coordinates, the `<anchor>` tag can set temporal attributes that specify when the link is active. If you do not include temporal attributes, the link stays active as long as the source file appears on screen. To add timing attributes, use the SMIL `begin` and `end` values. (You cannot use `clip-begin` or `clip-end`, however.)

The following example creates two temporal links for the file `video.rm`. The first link is active for the first five seconds of playback. The second link is active for the next five seconds. Because no spatial coordinates are given, the entire video is a link:

```
<video src="video.rm" region="videoregion">
  <anchor href="rtsp://.../video2.rm" begin="0s" end="5s"/>
  <anchor href="rtsp://.../video3.rm" begin="5s" end="10s"/>
</video>
```

Additional Information

See “Setting Begin and End Times” on page 60. The attributes use the SMIL timing values described in “Specifying Timing” on page 59.

Linking to a SMIL File

A SMIL file can define a link to another SMIL file. For example, a video played through a SMIL file may link to another SMIL file so that when a viewer clicks the video, a new presentation starts up in RealPlayer. To do this, you simply set the `href` attribute for the `<a>` or `<anchor>` tag to the new SMIL file’s URL.

You can also link to portions of a SMIL file. The following example from a target SMIL file uses id attributes (such as those used in regions to create region names) to define a target name for a <par> tag that groups a video and a text file. This id attribute functions like a name attribute in an HTML <a> tag:

```
<par id="text_and_video">
  <video src="video2.rm" region="newsregion"/>
  <textstream src="text.rt" region="textregion"/>
</par>
```

You then link the source SMIL file to the named target by including a pound sign ("#") and the target name within the link URL. Assuming the target SMIL file is named newmedia.smi, the source file's link to the <par> group looks like this:

```
<a href="rtsp://realserver.company.com/newmedia.smi#text_and_video">
  <video src="video.rm" region="videoregion"/>
</a>
```

Note that the target SMIL file defines two regions, newsregion and textregion. When RealPlayer receives the new SMIL file, it creates those regions as specified in the file's header.

Tips for Linking to a SMIL File

Note the following when linking to another SMIL file:

- You can link to any media source file or <par> or <seq> group by defining an id attribute for the file or group. Do not link to an element in a SMIL file header, however, or to a <switch> group.
- You cannot link to a media file in a <par> group and exclude the other files in that group. All files in the group will play in their designated regions.
- If additional files follow the target file in the SMIL file, those files play when the target file finishes playback.

  Tip

  To link to the single media file, link to it directly instead of through a SMIL file. Or create a new SMIL file that lists only the target media file.

- To link to a target within the same SMIL file, simply set the href attribute value to the target id, such as <a href="#text_and_video">. Be sure to include the pound sign before the id value.

Linking with a Timeline Offset

You can use the <anchor> tag's time coordinates to create a timeline offset in a linked file. Suppose you want to link a video to another video at 30 seconds into the second video's timeline. In the source SMIL file you define an <a> or <anchor> link from the first video to a SMIL file that contains the second video. In the second SMIL file, the video's <anchor> tag defines the timeline offset using SMIL timing parameters.

Here is a sample of the link in the first SMIL file:

```
<a href="rtsp://realserver.company.com/newmedia.smi#vid2">
  <video src="video.rm" region="videoregion"/>
</a>
```

The following is the linked video file in the second SMIL file, newmedia.smi:

```
<video src="video2.rm" region="newsregion">
  <anchor id="vid2" begin="30s"/>
</video>
```

Additional Information

"Specifying Timing" on page 59 describes the SMIL timing values.

CHAPTER 8: Playing a Presentation in a Web Page

As Chapter 7 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 RealPlayer 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 97.

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>
```

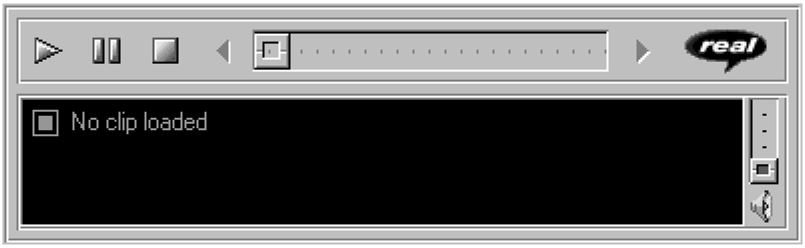
The following sections describe each RealPlayer control you can use. A tag’s WIDTH and HEIGHT parameters set the control’s size. The standard pixel widths and heights given below produce embedded controls approximately the same sizes as the RealPlayer controls. Specifying different pixel sizes scales 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 87.

Full Controls

All


No clip loaded

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

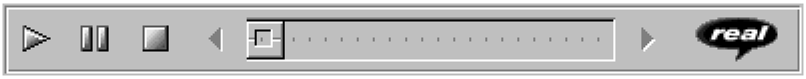
ImageWindow

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

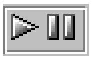
ControlPanel



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

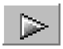
PlayButton



Displays the Play/Pause button.

Standard pixel width:	44
Standard pixel height:	26

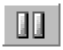
PlayOnlyButton



Displays the Play button without Pause.

Standard pixel width:	26
Standard pixel height:	26

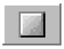
PauseButton



Displays the Pause button without Play.

Standard pixel width:	26
Standard pixel height:	26

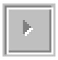
StopButton



Displays the Stop button.

Standard pixel width:	26
Standard pixel height:	26

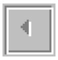
FFCtrl



Displays the Fast Forward control.

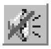
Standard pixel width:	26
Standard pixel height:	26

RWCtrl



Displays the Rewind control.

Standard pixel width:	26
Standard pixel height:	26

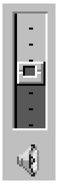


MuteCtrl

Displays the Mute control button.

Standard pixel width: 26

Standard pixel height: 26

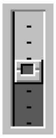


MuteVolume

Displays the Mute control and volume slider.

Standard pixel width: 26

Standard pixel height: 88



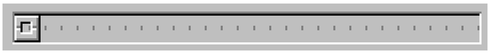
VolumeSlider

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

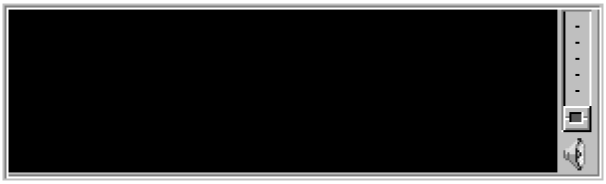
TACCtrl

Displays the Title, Author, and Copyright control. See also “Suppressing Labels” on page 87.

Standard pixel width: tbd

Standard pixel height: tbd

InfoVolumePanel

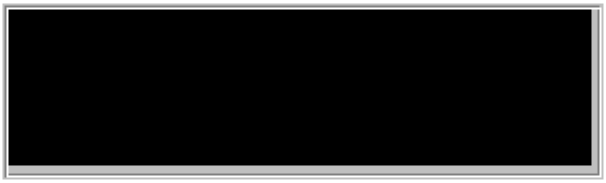


Displays the Title, Author, and Copyright information panel, as well as the volume slider. See also “Suppressing Labels” on page 87.

Standard pixel width: 300

Standard pixel height: 88

InfoPanel



Displays the Title, Author, and Copyright information panel. See also “Suppressing Labels” on page 87.

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=44 HEIGHT=26 CONTROLS=PlayButton CONSOLE=video1>
<EMBED SRC="http://realserver.company.com/sample1.rm"
WIDTH=26 HEIGHT=26 CONTROLS=StopButton CONSOLE=video1>

<H4>Video 2</H4>
<EMBED SRC="http://realserver.company.com/sample2.rm"
WIDTH=44 HEIGHT=26 CONTROLS=PlayButton CONSOLE=video2>
<EMBED SRC="http://realserver.company.com/sample2.rm"
WIDTH=26 HEIGHT=26 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=44 HEIGHT=26>
<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=26 HEIGHT=26>
<PARAM NAME=”CONTROLS” VALUE=”StopButton”>
<PARAM NAME=”CONSOLE” VALUE=”video1”>
</OBJECT>

<H4>Video 2</H4>
<OBJECT ID=RVOCX CLASSID=”…” WIDTH=44 HEIGHT=26>
<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=26 HEIGHT=26>
<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 the LOOP 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 64 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 69 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 69 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 a similar <EMBED> or <OBJECT> tag 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. In this case the results may not be what you expect.

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.

<EMBED> and <OBJECT> Tag Parameters

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 92
`AUTOSTART`	yes	yes	Starts playback automatically.	page 90
`CENTERED`	yes	yes	Centers clip in playback area.	page 82
`REGION`	yes	yes	Associates clip with SMIL regions.	page 91
`CONSOLE`	yes	yes	Links multiple instances of a tag.	page 87
`CONTROLS`	yes	yes	Adds RealPlayer controls to presentation.	page 82

(Table Page 1 of 2)

<EMBED> and <OBJECT> Tag Parameters (continued)

Parameter	Works with <EMBED>	Works with <OBJECT>	Function	Refer To
LOOP	yes	yes	Makes presentation loop continuously.	page 90
NAME	yes	no	Provides reference for JavaScript.	page 92
NOLABELS	yes	yes	Suppresses title, author, and copyright fields in controls.	page 87
SRC	yes	yes	Specifies source file.	page 81
WIDTH and HEIGHT	yes	yes	Sets size of playback area.	page 81
WINDOWED	yes	no	Makes plug-in windowless when set to false.	page 93

(Table Page 2 of 2)

CHAPTER 9: 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 10: 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 RealSystem G2 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 52.
- URLs for media source files begin with `rtsp://` when streaming from RealServer and `http://` when streaming or downloaded from a Web server.
- URLs list the correct locations of media source files.

Tip

RealNetworks' publishing tools can transfer 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">...</a>
```

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">..</a>
```

The following table explains the components of these URLs. Contact your RealServer administrator to get the RealServer address, HTTP port, and RAMGEN directory structure.

URL Components in a Web Page Hypertext Link

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 100.
`?embed`	This makes the presentation play back in the Web page. See Chapter 8 starting on page 79 for more information.

Using RAMGEN

In your Web page hyperlink, `/ramgen/` causes RealServer to generate a RAM file and download it to the user's Web browser. This RAMGEN parameter designates a virtual directory on RealServer. It may be followed in the URL by actual directory listings, as in this example:

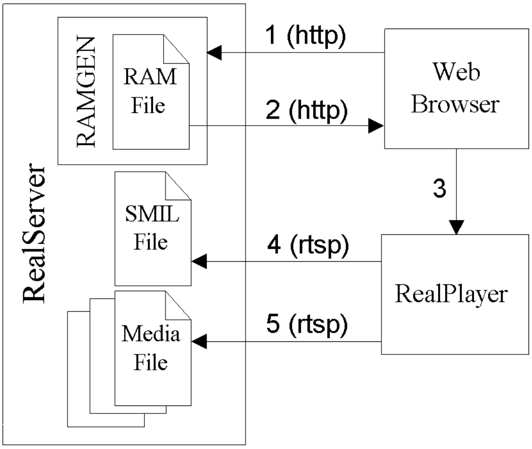
```
<a href="http://.../ramgen/media/samples/sample.smi">..</a>
```

This RAMGEN parameter 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 clips from RealServer. Because the RAM and SMIL files are small, this interaction occurs quickly. You should use RAMGEN even when linking to a single file type (such as a .rm file) that automatically launches RealPlayer.

Note

If you cannot use RAMGEN, you can write the RAM file as described in "Creating a RAM File Manually" on page 102.

The following figure illustrates the process of requesting a presentation through RAMGEN:



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 requests 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.

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 users to link to either the older presentation file or the new SMIL file depending on the RealPlayer version they use:

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

This link instructs RealServer to generate a RAM file for `sample.smi` if the user 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 URL after `altplay`. The following shows an end time set for `old_sample.rm`:

```
<a href="http://.../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`, users 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 RealSystem G2 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.

➤ **Follow these steps to stream files from a Web server:**

1. Note that restrictions on Web server streaming discussed in “Limitations on HTTP Streaming” below.
2. Configure your server’s MIME types as described in “Configuring Web Server MIME Types” on page 102.
3. Transfer your files to the Web server.
4. Write a RAM file as described in “Creating a RAM File Manually” on page 102.

Limitations on HTTP Streaming

There are several limitations on presentations streamed by Web servers:

- RealSystem File Types Only

  Except for RealFlash, the RealSystem 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 user 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 users 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 98. 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 54.

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.

Bundling RealPlayer Presets for Download

The **Presets** menu for RealPlayer G2 lets you save URLs, much like a Web browser lets you set bookmarks or favorites. If you want to share Presets with your users, you can post on your Web page a Presets Pack that users can download and import into their RealPlayers.

➤ **To create, export, and publish a Presets Pack:**

1. With RealPlayer, create a Presets Pack folder using **Presets>Organize Presets**. Click **New Folder** and supply a name that describes the Presets the folder will hold. This name appears in RealPlayer's **Presets** menu when a user imports the Presets Pack.
2. Play each URL in RealPlayer and use **Presets>Add to Presets** to add it to your Presets Pack folder. Be sure to give each Preset an informative name. Repeat this until you have added all the Presets you want to export.

   Tip

   Use **Presets>Organize Presets** to move existing Presets into your Presets Pack folder. You can move them back after you have exported the Presets Pack.

3. Choose **Presets>Organize** and select your Presets Pack folder.
4. Click **Export** and use the navigation dialog box to name and save your Presets Pack. The extension `.prx` is added to the file automatically.
5. To make your Presets Pack available for import, simply add a link to your Web page so that users can download the file. Here's an example:

   ```
   <a href="http://www.company.com/presets.prx">My Favorite Presets</a>
   ```

Depending on the browser used, the download may import the Presets Pack into RealPlayer automatically. Otherwise users can import it with one of these methods:

- Choosing **Presets>Import** and opening the Presets Pack.
- Double-clicking the Presets Pack.
- Dragging and dropping the Presets Pack onto RealPlayer.

Note

Imported Presets will not overwrite existing Presets that have the same Title or Category. See the **Sharing Presets** topic in RealPlayer's online help for details.

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 users 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.

Advertising on Timecast

Every day, thousands of people visit Timecast (**http://www.timecast.com**), RealNetworks’ online guide for streaming media sites and live events. If you regularly host RealAudio or RealVideo content of interest to the public, or you have a live event you want to advertise, you can place a listing on Timecast. Simply complete an online form to list your site:

- **http://www.timecast.com/help/addsite.html**

or live event:

- **http://www.timecast.com/help/addevent.html**

In the form, you give the site or event name, URL, short description, and contact person. A staff member then verifies your site or event before including it on Timecast. If you have questions or need to change a listing, please e-mail **timecast@timecast.com**.

For live events, please submit your request at least one business day in advance. If you have several live events, you can insert “live tags” within comment tags in your site’s HTML. A software robot then reads the tags and enters the events in Timecast. For details, see **http://www.real.com/help/content/livetags.html**.

Note

Timecast reserves the right to refuse or edit submissions.

CHAPTER 11: Broadcasting a Presentation

The Internet is swiftly becoming the next great broadcast medium. RealSystem G2 lets you broadcast live or prerecorded presentations over the Internet or your intranet. This chapter explains how you use encoding tools and RealServer to set up broadcasts.

> **Note**
>
> RealServer G2 currently supports live broadcasting, but RealNetworks G2 encoding tools do not yet support broadcasting with SmartStream codecs. An upcoming tools release will include SmartStream broadcast capability.

Introduction to Broadcasting

When a streaming presentation is delivered on demand, its starts from its beginning when the viewer clicks the presentation link in a Web page. Each viewer can receive the presentation at any time and use RealPlayer's controls to fast-forward or rewind through the presentation.

In a streaming broadcast, however, the user hosting the broadcast starts the presentation at a certain time. Viewers who click the presentation link join the broadcast in progress. Before the broadcast begins and after it completes, the presentation URL is not valid. During the broadcast the RealPlayer fast-forward and rewind controls do not function.

To make an analogy, on-demand content is like a movie on videotape. The viewer can see it at any time, skip forward, rewind, and pause. A streaming broadcast, though, is like a movie broadcast on a television channel. As with a TV broadcast, there are two types of streaming media broadcasts:

- Live content

  Live content is broadcast as it occurs. For example, you can broadcast the output of a video camera across the Internet or an intranet. RealSystem encodes the content as RealVideo in real-time without writing the content to a RealVideo file first.

- Prerecorded content

  Prerecorded content consists of video or audio you record and write to a digitized file. You can then edit the file before converting it to a streaming format and broadcasting it across a network. To the viewer, the presentation looks just like a live broadcast.

Broadcasting Tools

To broadcast a presentation, you need the following tools:

- Source capture equipment

  This equipment captures the broadcast content. It is typically a microphone or video camera connected to a audio or video capture card. For text it could be a live text feed coming in over a network.

- Editing equipment

  When you broadcast prerecorded content, you first write the source to a digitized file. You can then use editing software to optimize the file for broadcast. When broadcasting live content, though, content goes directly from the capture equipment to the encoder.

- Encoding tool

  The encoding tool takes the broadcast source and encodes it in the appropriate streaming format. When the broadcast is live, the tool sends the encoded output directly to RealServer. A RealVideo encoder, for example, encodes a camera's video output as RealVideo in real-time. A live encoder typically runs on a separate machine that has a network connection to the RealServer machine.

  RealNetworks encoding tools have live encoding capability for audio and video. Their broadcast wizards guide you through the encoding process and let you connect to RealServer easily. RealSystem's open architecture also lets you build an encoding tool to send RealServer any type of data for broadcast. For example, you could build an application that adds RealText mark-up to a stock ticker feed and broadcasts the output in real-time.

Additional Information

Visit **http://www.real.com** to download RealNetworks tools or the RealSystem G2 Software Development Kit (SDK), which you use to build a broadcast application. You can also get *RealText Authoring Guide*, which explains RealText mark-up.

- RealServer

RealServer streams the broadcast to your audience. You will need to work with the RealServer administrator to get the broadcast URL and the parameters for connecting an encoding tool to RealServer. Because each RealServer has a limit on the number of streams it can produce, verify that the RealServer you intend to use has broadcast capabilities appropriate for your anticipated audience size.

Tip

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

Additional Information

RealServer Administration Guide, available at **http://www.real.com**, explains how to configure RealServer for a broadcast.

Preparing a Broadcast

The following are tips for preparing a live or prerecorded broadcast. See the manual or online help for your encoding tool for instructions on how to encode and broadcast content.

Use SmartStream Codecs

RealAudio SmartStream codecs introduced in RealSystem G2 let you encode a broadcast for multiple bandwidths. Each viewer’s RealPlayer G2 then selects an encoding appropriate for its connection. Older versions of RealPlayer will not be able to receive a SmartStream broadcast, however. RealNetworks encoding tools let you specify SmartStream when you begin the broadcast.

To broadcast with non-SmartStream codecs, you need a separate encoder application running on a separate computer for each bandwidth connection you intend to support. This is required because non-SmartStream codecs require more CPU for each encoding than do SmartStream codecs. You then connect each encoder computer to RealServer and broadcast the different encodings under different URLs. Each viewer decides which URL to choose based on their network connection speed.

Additional Information

For more on codecs, see "Choosing RealAudio Codecs" on page 31.

Verify that the Encoding Machine has Enough CPU Power

Refer to your encoding tool's manual or online help for guidelines on machine requirements for broadcasting. The efficient SmartStream technology can broadcast multiply encoded bandwidth streams in real-time using just a moderately powerful PC. In general, encoding a 176x128 pixel RealVideo window using a frame rate of 4 to 15 frames per second requires a machine with the power of a 200 MHz Pentium computer. For higher frame rates and larger screen sizes, use a machine approximating a Dual Pentium 2.

Decide Whether to Archive a Live Broadcast

RealNetworks encoding tools let you write a live broadcast to file. RealServer can also archive the broadcast to file. The latter optional may be the better solution if your broadcast is long and your encoding machine has limited disk space. The RealServer administrator can set up RealServer to archive the live broadcast.

Get the Broadcast URL and RealServer Parameters

The RealServer administrator can give you the parameters you need to connect an encoding tool to RealServer. The administrator will also provide the broadcast URL or URLs.

Perform a Trial Run of a Live Broadcast

When you broadcast live content, you don't get a second chance. So it's good practice to perform a trial run to ensure that the equipment works properly and the broadcast results are what you expect. Because you can't edit a live broadcast the way you can edit a prerecorded file, it's important to set your audio levels and plan your video shots carefully in advance.

During both the trial run and the live broadcast, view the broadcast ourput with RealPlayer. When RealPlayer connects, check that the buffering time for receiving a live RealVideo stream does not exceed 5-10 seconds. Throughout the presentation, keep an eye on the broadcast quality.

If you experience problems with the encoded output, use the encoding tool to lower the video frame size and frame rate (fps), or select lower-bandwidth codecs. If these adjustments do not help, you may need to run your encoder on a more powerful machine.

Additional Information

For pointers on producing audio, see "Recording Audio" on page 28. For video, see "Producing High Quality Video" on page 37.

Create a SMIL File for a Multipart Presentation

Using SMIL, you can easily embed your broadcast in a multifile presentation. You might use a SMIL file, for example, to create a video region for your live broadcast and a RealPix region that features rotating ads. The SMIL file then uses the broadcast URL for the video region and a standard URL to the RealPix presentation file.

Keep in mind, however, that SMIL does not synchronize on-demand files with the broadcast. When the SMIL presentation starts, the viewer begins to receive the on-demand files in the order they are defined by the SMIL grouping and timing tags. The viewer joints into the broadcast in progress, however. So if you set up ad rotation through SMIL, viewers receive ads relative to the time they click the presentation link. Thus all viewers will not see the same ads at the same points during the broadcast.

Additional Information

See Chapter 7 beginning on page 51 for information on SMIL. Visit **http://www.real.com** to get *RealPix Authoring Guide*, which explains RealPix presentations.

Advertise a Public Broadcast Event

If you are broadcasting content that has wide public appeal, advertise your event on Timecast (**http://www.timecast.com**), the online guide to RealAudio and RealVideo.

Additional Information

"Advertising on Timecast" on page 105.

APPENDIX A: File Type Reference

The following table provides a quick reference to file types commonly used in RealSystem G2 production. This is not a definitive list of all file types used in RealSystem.

RealSystem File Types

Extension	File Type	Reference
RealSystem G2 Streaming File Types		
.ra	RealAudio	“Producing RealAudio” on page 31
.rm	RealMedia file consisting of RealVideo and RealAudio	“Producing RealVideo” on page 39
.rp	RealPix mark-up	“Images” on page 11
.rt	RealText mark-up	“Text” on page 11
Files Types Also Streamed by RealSystem		
.au	Audio file	“AU” on page 35
.avi	Windows standard video file	“AVI” on page 40
.jpg	JPEG file used in RealPix	“Images” on page 11
.stg	STiNG file used in RealPix	“Images” on page 11
.swf	Shockwave Flash file for RealFlash	“Producing Animation” on page 41
.viv	Vivo video file	“Vivo” on page 40
.wav	Waveform audio file	“WAV” on page 35
RealSystem Information Files		
.prx	RealPlayer Presets export file	“Bundling RealPlayer Presets for Download” on page 103
.ram	RAM file to launch RealPlayer	“Linking your Web Page to your Presentation” on page 97
.rpm	RAM file to launch RealPlayer for an embedded presentation	“Linking your Web Page to your Presentation” on page 97
.smi	SMIL file for putting presentations with multiple file types together	“Assembling a Presentation with SMIL” on page 51

(Table Page 1 of 2)

RealSystem File Types (continued)

Extension	File Type	Reference
Common Source Files Converted to Streaming File Formats		
.aif	Audio Interchange Format source for RealAudio	"Producing RealAudio" on page 31
.fla	Flash source file for Shockwave Flash	"Producing Animation" on page 41
.mov	QuickTime movie source for RealVideo	"Producing RealVideo" on page 39
.snd	Sound source file for RealAudio	"Producing RealAudio" on page 31

(Table Page 2 of 2)

APPENDIX B: SMIL Language Codes

As “Setting Language Choices” on page 61 explains, a SMIL file can list different language choices that RealPlayer chooses from based on its language preference. The following table lists the codes you can use in a SMIL file to indicate media clips created for a specific language.

Code	Language
af	Afrikaans
sq	Albanian
ar-iq	Arabic (Iraq)
ar-dz	Arabic (Algeria)
ar-bh	Arabic (Bahrain)
ar-eg	Arabic (Egypt)
ar-jo	Arabic (Jordan)
ar-kw	Arabic (Kuwait)
ar-lb	Arabic (Lebanon)
ar-ly	Arabic (Libya)
ar-ma	Arabic (Morocco)
ar-om	Arabic (Oman)
ar-qa	Arabic (Qatar)
ar-sa	Arabic (Saudi Arabia)
ar-sy	Arabic (Syria)
ar-tn	Arabic (Tunisia)
ar-ae	Arabic (U.A.E.)
ar-ye	Arabic (Yemen)
eu	Basque
bg	Bulgarian
ca	Catalan

Code	Language
zh-hk	Chinese (Hong Kong)
zh-cn	Chinese (People’s Republic)
zh-sg	Chinese (Singapore)
zh-tw	Chinese (Taiwan)
hr	Croatian
cs	Czech
da	Danish
nl	Dutch (Standard)
nl-be	Dutch (Belgian)
en	English
en-au	English (Australian)
en-bz	English (Belize)
en-gb	English (British)
en-ca	English (Canadian)
en	English (Caribbean)
en-ie	English (Ireland)
en-jm	English (Jamaica)
en-nz	English (New Zealand)
en-za	English (South Africa)
en-tt	English (Trinidad)
en-us	English (United States)

Code	Language
et	Estonian
fo	Faeroese
fi	Finnish
fr-be	French (Belgian)
fr-ca	French (Canadian)
fr-lu	French (Luxembourg)
fr	French (Standard)
fr-ch	French (Swiss)
de-at	German (Austrian)
de-li	German (Liechtenstein)
de-lu	German (Luxembourg)
de	German (Standard)
de-ch	German (Swiss)
el	Greek
he	Hebrew
hu	Hungarian
is	Icelandic
in	Indonesian
it	Italian (Standard)
it-ch	Italian (Swiss)
ja	Japanese
ko	Korean
ko	Korean (Johab)
lv	Latvian
lt	Lithuanian
no	Norwegian
pl	Polish
pt-br	Portuguese (Brazilian)
pt	Portuguese (Standard)
ro	Romanian
sr	Serbian
sk	Slovak

Code	Language
sl	Slovenian
es-ar	Spanish (Argentina)
es-bo	Spanish (Bolivia)
es-cl	Spanish (Chile)
es-co	Spanish (Colombia)
es-cr	Spanish (Costa Rica)
es-do	Spanish (Dominican Republic)
es-ec	Spanish (Ecuador)
es-sv	Spanish (El Salvador)
es-gt	Spanish (Guatemala)
es-hn	Spanish (Honduras)
es-mx	Spanish (Mexican)
es-ni	Spanish (Nicaragua)
es-pa	Spanish (Panama)
es-py	Spanish (Paraguay)
es-pe	Spanish (Peru)
es-pr	Spanish (Puerto Rico)
es	Spanish (Spain)
es-uy	Spanish (Uruguay)
es-ve	Spanish (Venezuela)
sv	Swedish
sv-fi	Swedish (Finland)
th	Thai
tr	Turkish
uk	Ukrainian
vi	Vietnamese

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.

broadcast

To deliver a presentation, whether live or prerecorded, in which all viewers join the presentation in progress.

buffering

The process of receiving and storing presentation data before playing it back. RealSystem G2 presentations have an initial buffering time called “preroll”. Once the presentation starts, buffering should be minimal or the presentation may pause.

byte

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

C

client

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

Part of 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.

F **Flash**

A Macromedia tool for creating animations that can be streamed as RealFlash. See also “Shockwave Flash”.

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 G2 supports for backwards compatibility with RealSystem versions 3.0 through 5.0.

preroll

Buffering that occurs at the start of a presentation. Preroll should be 15 seconds or less.

presentation

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

R **RDP**

The proprietary data package protocol RealServer G2 uses (along with RTSP) when communicating with RealPlayer G2. Contrast to “RTP”.

RealAudio

A RealSystem data type for streaming audio over a network. RealAudio files use the `.ra` extension.

RealFlash

A RealSystem data type for streaming animation and audio over a network. A RealFlash presentation consists of Shockwave Flash and RealAudio.

RealPix

A RealSystem data type (file extension `.rp`) for streaming still images over a network. It uses a markup language for creating special effects such as fades and zooms.

RealPlayer

RealNetworks client designed to play multimedia presentations streamed by RealServer.

RealServer
RealNetworks software used to stream multimedia presentations to RealPlayer.

RealSystem
The system for streaming data types such as RealAudio and RealVideo over a network. It consists of RealServer, RealPlayer, and production tools.

RealText
A RealSystem data type (file extension `.rt`) for streaming text over a network. It uses a mark-up language for formatting the text stream.

real-time
Delivered as it occurs. For example, a live broadcast is streamed across a network in real-time.

RealVideo
A RealSystem data type for streaming video over a network. RealVideo files use the `.rm` extension.

RTP
The open, standards-based data package protocol RealServer G2 uses (along with RTSP) to communicate with RTP-based clients. Contrast to “RDP”.

RTSP
The open, standards-based control protocol RealServer G2 uses to stream files to RealPlayer G2 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.

Shockwave Flash
The Flash file format (file extension `.swf`) used to stream the animation component of a RealFlash presentation.

SmartStream
A codec technology that provides superior RealAudio sound and the ability to encode a single file for multiple bandwidths. Available only in RealSystem G2.

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.

Index