REALSYSTEM G2 PRODUCTION GUIDE

BETA 2 Release

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, images, and text.

Note

The HTML version of this manual, available at **http://service.real.com/help/library/index.html**, contains multimedia examples you can view with RealPlayer.

How this Manual is Organized

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 RealSystem Presentations

If you are new to streaming media, this chapter walks you through the steps you take to put together a RealSystem G2 presentation.

Chapter 3: Targeting Bandwidth

This chapter explains how to target bandwidth connections, 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 explains how to use 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 into your Web page, follow the instructions in this chapter.

Chapter 9: Inserting Ads in a Presentation
This chapter explains how to use SMIL to insert ads into your streaming presentation.

Chapter 10: Delivering a Presentation
This chapter provides step-by-step instructions for moving your presentation files to RealServer and linking your Web page to them. It also explains how to use a Web server to deliver simple presentations.

Chapter 11: Broadcasting a Presentation
Refer to this chapter if you plan to broadcast an audio or video event.

Appendix A: Quick Steps for Streaming Clips
If you are new to RealSystem streaming, this appendix helps you get started quickly by listing the basic steps you take to create simple presentations.

Appendix B: Advanced Production Tips
For advanced users, this appendix conveys useful production techniques that help you get the most out of RealSystem G2.

Appendix C: File Type Reference
This appendix provides a quick reference for common file types used in RealSystem G2 streaming.

Appendix D: SMIL Language Codes
If you create clips in different languages, you can use these codes in the SMIL file to indicate language choices.

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.
emphasis	Bolded text is used for emphasis.
[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://service.real.com/help/library/index.html**:

- *RealText Authoring Guide*

  This manual explains how to create streaming text. You can use RealText, for example, to create a stock ticker 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 or create new plug-ins for RealServer or RealPlayer. Knowledge of programming is required to use the SDK. Register for and download the SDK from **http://www.real.com/devzone/sdks/rmsdk/**.

Technical Support

For technical support with RealSystem G2, please fill out the form at:

- **http://service.real.com/contact/email.htm**

The information you provide in this form will help technical support personnel to give you a prompt response. For general information about RealNetworks' technical support, visit:

- **http://service.real.com/help/call.html**

CHAPTER 1: What's New in RealSystem G2?

The open, end-to-end architecture of RealSystem G2 changes the nature of streaming multimedia, giving you more possibilities for creating Web-based multimedia than before. If you're familiar with past versions of RealSystem, this chapter gives you a quick look at the many changes in RealSystem G2.

New 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. In RealSystem G2, they join RealAudio, RealVideo, and RealFlash as standard RealSystem clip types.

Additional Information

See "Choosing Media to Stream" on page 11.

Popular File Formats Stream Natively

With RealSystem, you can now stream many popular audio and video formats. Stream WAV files 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 11.

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/devzone/realdevelopers/benefits.html** for information on joining the RealNetworks developer program.

Advancements in RealAudio and RealVideo

SureStream Technology Encodes Single Clips for Multiple Bandwidths

The new SureStream technology available exclusively in RealSystem G2 lets you encode a single RealAudio or RealVideo clip for up to six separate bandwidths. All Web page visitors click the same link to play the clip, but their RealPlayers receive different encodings appropriate for their various connection speeds.

Additional Information

See "Encoding Clips for Multiple Bandwidths" on page 35 for an overview of SureStream.

New RealAudio Codecs Provide Superior Sound Quality

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

Additional Information

"Choosing RealAudio Codecs" on page 45 lists the new 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. RealPlayer G2 can now play multiple streams that use the same codec.

New RealVideo Codec Provides Fast Encoding for Multiple Bandwidth Clips

RealSystem G2 introduces a new RealVideo codec that provides faster encoding and lets you use SureStream technology to encode a single RealVideo clip for up to six different bandwidths.

Additional Information

See "Choosing RealVideo Codecs" on page 57 for more information.

Easier Presentation Assembly

SMIL Files Coordinate Presentations

For presentations that include more than one clip, you create a SMIL file to specify how and when each clip 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 18. Refer to Chapter 7 beginning on page 75 for instructions on using SMIL.

Bandwidth Negotiation through Multiply Encoded Clips or SMIL

RealSystem G2 introduces simpler methods for supporting multiple bandwidth connections. As described previously, the new SureStream technology allows you to encode a single RealAudio or RealVideo clip for multiple connection speeds. Or you can let RealPlayer choose between different versions of a presentation based on bandwidth parameters in the SMIL file. Either way, you need just one link on your Web page, and your encoded clips do not need to conform to any naming conventions.

Additional Information

See "Supporting Multiple Bandwidth Connections" on page 35 for an overview.

RealPlayer Launched Automatically

RealServer's RAMGEN feature can launch RealPlayer automatically. This means you do not need to create a RAM file (extension `.ram` or `.rpm`) manually. In your Web page, you link to a SMIL file or media clip, including in the URL a RAMGEN parameter that causes the Web browser to launch RealPlayer and give it the SMIL file or clip.

Additional Information

See "Linking your Web Page to RealServer" on page 141.

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 clips 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. RealPlayer G2 can still play back clips streamed by an earlier version of RealServer.
- 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. See "Listing Alternate Presentations" on page 143.

- 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 RealServer G2.

CHAPTER 2: Creating RealSystem Presentations

RealSystem G2 gives you the power to create compelling, complex multimedia presentations streamed over a network. It 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. This chapter explains the basics of how you use RealSystem G2 to put together stunning multimedia presentations.

Tip

To experience the many possibilities of streaming media, download RealPlayer G2 from **http://www.real.com** and view presentations at **http://www.timecast.com** and **http://www.real.com/showcase**.

Streaming Multimedia Clips

A RealSystem presentation can consist of one clip, such as a single RealAudio clip, or multiple clips played in sequence or in parallel. RealSystem *streams* these presentations over a network. If you have ever *downloaded* a video or audio clip to your Web browser, you know that it can take several minutes to receive a clip that plays for a few seconds. In contrast, a clip streamed by RealServer G2 begins to play back almost immediately.

Choosing Media to Stream

RealSystem G2 gives you many possibilities for combining media clips into presentations. There are two basic types of file formats you can stream with RealSystem G2:

- Standard or "open" formats

  Editing programs typically let you save or export files to a standard, "open" format. Video editing programs usually let you export files to the AVI format or QuickTime. RealSystem can stream several standard formats, but these formats may not be optimized for network 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 export files directly to streaming formats.

Audio

Chapter 4 beginning on page 39 discusses the audio formats you can stream:

- RealAudio (.ra)
- AIFF (.aif)
- AU (.au)
- Wave (.wav)

Video

Chapter 5 beginning on page 51 describes the video formats you can stream:

- RealVideo (.rm)
- ASF (.asf)
- AVI (.avi)
- QuickTime (.mov)
- Vivo (.viv)

Animation

RealFlash, which pairs Macromedia Flash animation with a RealAudio soundtrack, lets you stream animated presentations. See Chapter 6 beginning on page 63 for details.

Images

RealSystem presentations can include still images in these formats:

- JPEG (.jpg)

  RealPlayer G2 can display RGB baseline JPEGs. Progressive and grayscale JPEGs are not supported.

- GIF87, GIF89a, and animated GIF (.gif)

  Both interlaced and noninterlaced GIFs will work, but noninterlaced GIFs are recommended.

You can also assemble GIF, JPEG, BMP, or STiNG images in a RealPix presentation to create eye-catching slide shows with special effects such as dissolves and zooms. For more information, download *RealPix Authoring Guide* from **http://service.real.com/help/library/index.html**. For more on the STiNG image format, visit **http://www.iterated.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://service.real.com/help/library/index.html**.

Additional Streaming Formats

RealSystem's plug-in technology lets it stream nearly any type of clip 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.

Compatibility with Earlier RealPlayer Releases

Most RealPlayer users upgrade when a new version of RealPlayer becomes available. If you need to create presentations playable by older versions of RealPlayer, though, take into account RealPlayer and clip type compatibility. The table below indicates which RealPlayer versions, such as RealPlayer G2 or RealPlayer 5.0, can play which types of clips. RealPlayer 4.0, for example, plays only RealAudio and RealVideo.

RealPlayer Compatibility with RealSystem G2 Clips

Clip Type	G2	5	4	3	2	1
RealAudio	X	X	X	X	X	X
RealVideo	X	X	X	–	–	–
RealFlash	X	X	–	–	–	–

(Table Page 1 of 2)

RealPlayer Compatibility with RealSystem G2 Clips (continued)

Clip Type	G2	5	4	3	2	1
RealPix	X	–	–	–	–	–
RealText	X	–	–	–	–	–
Open formats such as WAV and AVI	X	–	–	–	–	–
SMIL	X	–	–	–	–	–

(Table Page 2 of 2)

Note that this table covers general clip compatibility, not codec compatibility. Later versions of RealPlayer typically introduce new RealAudio and RealVideo codecs. Earlier versions of RealPlayer cannot play the new RealAudio G2 codecs, for example. So when planning for backwards compatibility, make sure you encode RealAudio or RealVideo clips with a codec available in your targeted versions of RealPlayer.

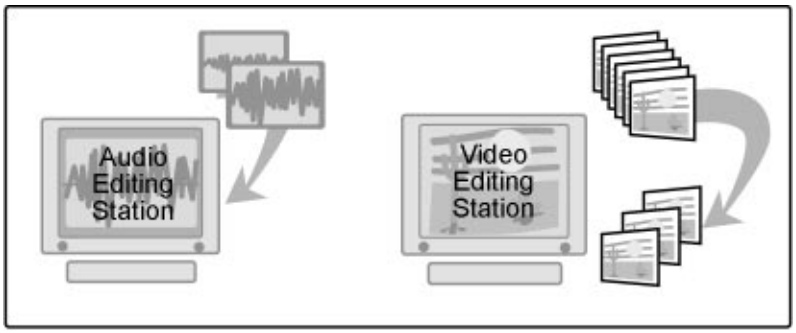
Additional Information

For more on codecs, see "Choosing RealAudio Codecs" on page 45 and "Choosing RealVideo Codecs" on page 57.

Using Editing and Encoding Tools

After choosing your streaming media formats, gather content and use your preferred editing tools to prepare your source files. Sound editing software, for example, lets you optimize an audio clip's dynamic range. With video editing software you can set the video's window size. RealSystem does not require you to use specific editing tools. Just ensure that your editing tools can save files in streaming formats, or in open formats you can easily convert to streaming formats.

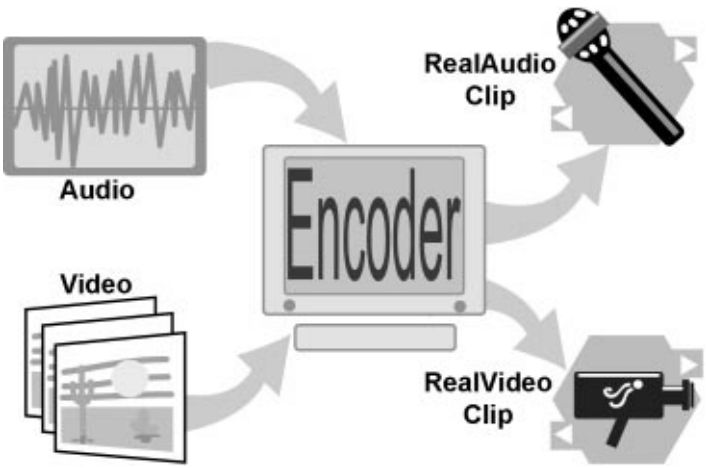
Use Your Favorite Editing Tools to Prepare Media Files



RealNetworks Encoding Tools

If your editing program does not export files to the streaming format you want, you can use an encoding tool to convert the file. RealNetworks provides free, basic tools and sells enhanced tools for converting popular sound and video formats to RealAudio and RealVideo. In addition, plug-ins for popular programs such as Adobe Premiere and Microsoft PowerPoint let you save presentations directly as RealVideo.

RealNetworks Encoding Tools Create Streaming Clips



Additional Information

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

Additional Tools

RealSystem G2 provides virtually unlimited possibilities for streaming media. It does not lock you into a small set of streaming formats or a specific set of tools. In addition to new types of streaming media, RealNetworks' partners and other parties are continually producing new tools that help you put together sophisticated presentations.

Tip

Visit **http://www.realstore.com** to find out about the many tools available to help you create streaming media. Also check **http://www.real.com/devzone** frequently for information about new media types and tools.

Working with Timelines and Bandwidths

As you create streaming media clips, you need to consider how to keep the clips synchronized to a single presentation timeline. You also need to create presentations for specific bandwidths. These are two important, interrelated steps for producing streaming multimedia presentations.

Synchronizing Clips to a Timeline

Because a static Web page has no timeline, images and text download without a preset order. You may notice when browsing a Web page that one image may download partially, then another image begins to appear, then the first image completes, and so on. A Web page does not have an internal timeline, so exactly when a certain image gets to the browser doesn't matter. What matters is that the entire page downloads as soon as possible.

When you stream multimedia, though, clips have timelines and must flow smoothly once they've started to play back. Imagine how jarring it would be for the visual track of a video to play silently for a minute, then pause as its audio track catches up to it a minute later. When you stream multimedia, therefore, it's important that your presentation keeps clips synchronized. There are three aspects of timelines you may work with:

- Clips with Internal Timelines

  Audio, video, and animation have internal timelines. In a two-minute video, for instance, each frame corresponds to a specific point in a two-minute timeline. Each second of audio meshes with each second of the

visual image throughout the clip's overall timeline. Your video, audio, or animation editing program is your main tool for manipulating the clip's timeline, which is part of the fabric of the clip.

- Clips with Variable Timelines

  With RealPix or RealText, you define timing tags that set when each image or text block appears. Unlike a video clip in which the image should always be moving, images and text can remain stationary for indefinite periods until new images and text replace them. When combining clips, it's typically easier to produce audio, video, or animation first. Then set the RealPix and RealText timelines to coordinate with those clips.

  Additional Information

  Get the RealPix and RealText authoring guides from **http://service.real.com/help/library/index.html**.

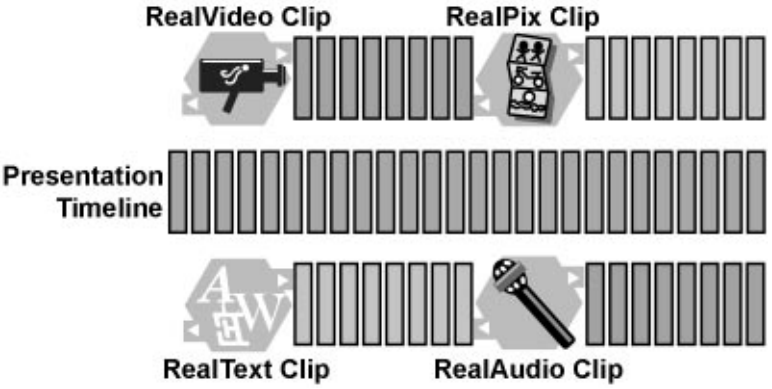
- SMIL Timing Commands

  A SMIL file assembles your presentation and can include its own timing elements. Timing a presentation can be as simple as starting one clip as soon as another one stops. But you can also use explicit timing commands to delay playback for ten seconds, for example, or start a clip playing at 30 seconds into its internal timeline. SMIL's timing commands are optional, but they give you an extra level of flexibility you may need when putting multimedia clips together.

  Additional Information

  "Writing a SMIL File" on page 18 provides an overview of SMIL. For specifics on SMIL timing, see "Specifying Timing" on page 86.

Clip Timelines Coordinate with an Presentation Timeline



Keeping a Bandwidth Budget

Any presentation streamed over a network has a *bandwidth* budget because each person viewing streaming clips has a network connection with a top speed, such as 28.8 Kilobits per second. Even when your clips are perfectly synchronized, the presentation may stall if at some point it requires more bandwidth than the viewer has available. This happens because RealServer needs to transmit at a certain point in the timeline more data than can get through the network connection to RealPlayer.

When you develop a streaming media presentation, you need to consider the bandwidth constraints your audience will have. This is crucial for creating presentations that start to play back quickly and flow smoothly. Web users don't like to wait more than a few seconds for something to happen after they click a link. And if your clips stop and restart frequently, viewers are not likely to stay watching. Fortunately, RealSystem offers sophisticated features that let you create a single presentation that looks good over a slow connection and great over a fast one.

Additional Information

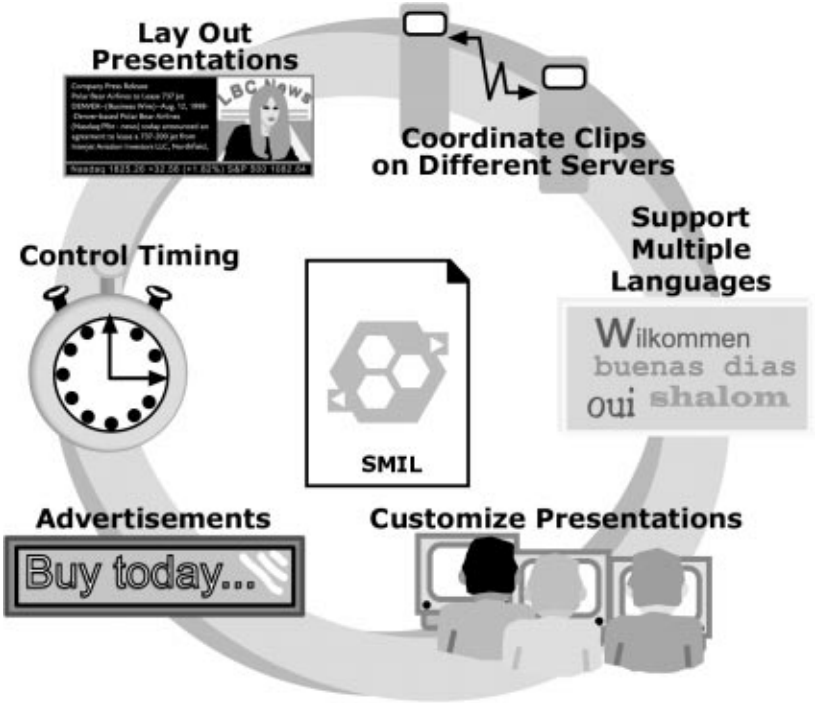
Chapter 3 beginning on page 27 provides the details for considering bandwidth in streaming presentations.

Writing a SMIL File

With your clips 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 clip. But when you have multiple clips, SMIL's simple mark-up language specifies how and when the clips play.

Advantages of Using SMIL



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 clips into container formats such as these:

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

  Although RealSystem G2 can stream either of these container formats, using RealSystem streaming formats and putting the presentation together with a SMIL file gives you greater flexibility. To alter your presentation, for example, you simply edit the SMIL file rather than merge the clips again into a different container file.

- Use Clips in Different Locations

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

- Support Multiple Languages

  A SMIL file can list different language options for clips. To create a video with sound tracks in different languages, for example, you produce one video clip with no soundtrack, then create audio clips 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 clips 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.

  Note

  RealSystem G2's SureStream technology also lets you support multiple bandwidth connections within a single clip, for more information, see "Supporting Multiple Bandwidth Connections" on page 35.

- 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 clip.

- Lay Out the Presentation

  When your presentation includes multiple clips, such as a RealVideo clip playing simultaneously with a RealPix slideshow, you use SMIL to define the layout.

- Include Ads

  For commercial sites, the SMIL file can insert ads into your presentation, either in a separate ad banner or between clips playing in sequence.

Additional Information

Chapter 7 beginning on page 75 explains the SMIL file syntax. Chapter 9 beginning on page 133 provides examples of ad insertion.

Hosting a Presentation on a Server

When your presentation is complete, you move the streaming media clips and SMIL file to RealServer G2 or a Web server for delivery. You can play back your presentation in RealPlayer or directly in your Web page. The latter option, which uses RealPlayer's Netscape plug-in or ActiveX Control, requires mark-up tags in your Web page that specify how the presentation displays.

Additional Information

Beginning on page 139, Chapter 10 explains how to link your Web page to a RealSystem presentation hosted on a server. For information on embedding a presentation in your Web page, see Chapter 8 on page 115.

Using RealServer G2 or a Web Server

RealServer G2 is the preferred host for RealSystem presentations. Designed specifically to stream multimedia over networks. RealServer keeps multiple clips synchronized and uses many advanced features to ensure that clips stream smoothly under adverse network conditions. While you can use a standard Web server to host some streaming presentations, you will not get the results you'll experience when using RealServer.

Tip

If you don't have RealServer G2 available, check out the hosting services of Real Broadcast Network (RBN) at **http://www.real.com/rbn**. RBN provides full services

for encoding, hosting, and broadcasting events to small or large audiences.

HTTP vs. RTSP

A Web server uses the HTTP protocol, as you can see in URLs that begin with `http://`. The HTTP protocol downloads files without regard to timelines, making clips with timelines more likely to stall. Although RealPlayer can play back clips as they download, the HTTP protocol does not give RealPlayer the ability to adjust the download to compensate for changing network conditions.

In contrast, URLs for media clips streamed by RealServer begin with `rtsp://`, because RealServer uses the RTSP protocol, which is designed specifically to stream clips that have timelines. RTSP lets RealPlayer G2 and RealServer G2 exchange information about a presentation in progress and adjust the streaming data to keep the clips playing smoothly.

When two clips play side-by-side, for example, RealPlayer uses RTSP to communicate with RealServer about each clip's progress, indicating how much data it needs to keep the presentation synchronized. RealServer can then adjust the data flow, reducing low priority data if necessary to ensure that crucial data gets through. Communication like this is not possible when a Web server sends clips to RealPlayer.

You don't need to know the specifics of RTSP to create great presentations. You just need to ensure that RealServer G2 is available to stream your clips. If only a Web server is available, you can still create multimedia presentations, but you won't be able to use all RealSystem features. In either case, make sure you have a good understanding of RealSystem G2 production as described in this manual before you start creating your clips. This helps ensure that your presentation harmonizes with the server that hosts it.

Additional Information

For information on RealSystem features that do not work with Web server hosting, read "Limitations on Web Server Playback" on page 145.

Launching a Presentation

Because some browsers may not be configured to launch RealPlayer when they receive a SMIL file or a media clip, you need to ensure that RealPlayer launches when a visitor to your Web page clicks the link to your presentation. When

RealServer G2 hosts your presentation, you simply include a RAMGEN parameter in the Web page URL to launch RealPlayer.

Additional Information

For more on RAMGEN, see "Streaming Clips from RealServer G2" on page 140.

When a Web server hosts your presentation, you launch RealPlayer by linking your Web page to a RAM file instead of a SMIL file or a media clip. When the browser receives the RAM file, it launches RealPlayer and gives it the RAM file, which contains the URLs to your media clips or SMIL file. RealPlayer uses this information to request the presentation from the Web server. Because the RAM file is a small text file, this interaction takes little time.

Additional Information

For more on Web server playback and RAM files, see "Playing Clips from a Web Server" on page 143.

Working with the RealServer Administrator

Because earlier versions of RealServer do not handle SMIL files and many of the RealSystem G2 media types, make sure you have RealServer G2 available to host your presentation. When you host a presentation on RealServer G2, the RealServer administrator will give you the basic URL parameters, such as the server address and its RTSP and HTTP port parameters. The RealServer administrator can also set up many content delivery and security features, such as:

- live broadcasts
- pay-per-view content
- password authentication

Additional Information

RealServer Administration Guide, available at **http://service.real.com/help/library/index.html**, explains RealServer features.

Playing Clips Back Locally

Although the main function of RealSystem is to stream media clips over a network, you can also create presentations that play back from a user's local machine. An example is a multimedia-enhanced book written with HTML and

containing links to RealSystem clips. Users then download all the book files to their local machines and play back the media clips in RealPlayer.

In this case, you produce clips as described in this manual, except that you don't have to target specific network bandwidths. When you put the presentation together, you simply use URLs that point to clips on the user's machine instead of on RealServer or a Web server.

Additional Information

For more on local URLs, see "Linking to Local Clips" on page 81.

Viewing a Presentation with RealPlayer

With RealPlayer G2 installed, you simply click the presentation link in your Web page. RealPlayer buffers presentation clips for a few seconds, then begins to play the presentation back in its own window or your browser. RealPlayer downloads are available from RealNetworks at **http://www.real.com**.

Protecting Copyrighted Content

Unlike a Web browser, RealPlayer does not store clips in a disk cache or allow users to copy or download still images. This helps you keep copyrighted material secure. Optionally, RealNetworks tools let you encode video and audio clips so that viewers can record the streamed clips to their local machines.

Downloading RealPlayer Plug-Ins

RealPlayer can play virtually any streaming clip because of its plug-in technology. RealPlayer plug-ins function like Web browser plug-ins. When RealPlayer receives a streaming RealVideo clip, 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 clip, 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 information about developing RealPlayer plug-ins or building RealPlayer capabilities into an application, visit **http://www.real.com/devzone/realdevelopers/**.

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 clips in your presentation reside on users' local computers rather than on RealServer or a Web server.

What is Bandwidth?

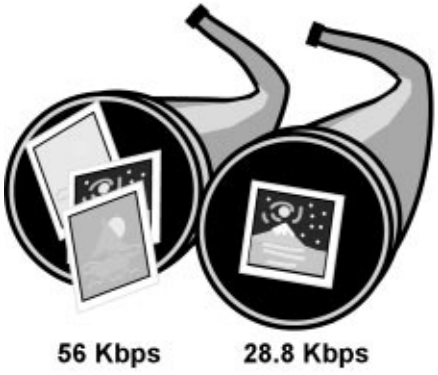
Total bandwidth is the upper limit on how much data can pass through a network connection per second. Internet bandwidth is described in Kilobits per second (Kbps). A 28.8 Kbps modem, for example, can receive data at any speed up to 28.8 Kbps. Bandwidth is analogous to a speed limit, such as 60 m.p.h. 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 bandwidths is crucial to delivering compelling multimedia presentations. 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. Additionally, you can target multiple bandwidths with a single clip that looks good at low bandwidths and great at high bandwidths.

Faster Connections Receive More Data



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 before 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 network 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 clips. For a multiclip 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 clips. For a 28.8 Kbps connection, you have approximately 20 Kbps total for your presentation. The following table lists the recommended maximum bit rates for streaming presentations over various network connections.

Bit Rates Available for Streaming Clips

Target Connection Speed	Recommended Maximum Bit Rate for Streaming Clips
14.4 Kbps modem	10 Kbps
28.8 Kbps modem	20 Kbps
56.0 Kbps modem	32 Kbps
56.0 Kbps ISDN	45 Kbps
112 Kbps dual ISDN	80 Kbps

Developing a Bandwidth Strategy

Once you know the bit rate or rates available for your streaming clips, you can begin to develop your bandwidth strategy. If you want to stream just one clip, 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 RealSystem media types. If you stream other types of clips, make sure you understand each clip's bandwidth characteristics.

RealAudio

RealAudio consumes bandwidth at a flat rate determined by the codec used to encode the clip. If you have a RealAudio clip encoded with a 8 Kbps codec, for example, that clip will steadily consume 8 Kbps of bandwidth as long as it plays. And with SureStream technology described in "Encoding Clips for Multiple Bandwidths" on page 35, you can encode a single RealAudio clip for up to six bandwidths.

Additional Information

See "Choosing RealAudio Codecs" on page 45.

RealVideo

RealVideo consumes bandwidth at a flat rate for a given bandwidth target. If you target 20 Kbps and use an 8 Kbps RealAudio codec for the soundtrack, the image track steadily consumes 12 Kbps. RealVideo encoders vary a clip's frame rate and image quality to produce the best possible quality for the bandwidth target. SureStream technology described in "Encoding Clips for Multiple Bandwidths" on page 35 lets you encode a single RealVideo clip for up to six bandwidths.

Additional Information

See "Choosing RealVideo Codecs" on page 57. For tips on shooting video, see "Recording Video" on page 53.

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 Clip" on page 65.

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**.

Images

By default, JPEG and GIF images rendered directly in RealPlayer (that is, images that are not part of a RealPix presentation) stream at 12 Kbps. You can modify this bit rate through a SMIL file.

Additional Information

See "Defining Image Options" on page 109 for more on using SMIL to change image clip bandwidth use.

Delivering a Single Media Clip

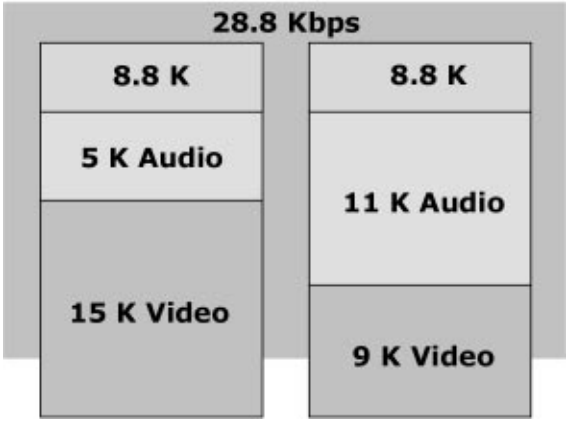
Suppose you want to create an audio clip that Web users with 14.4 Kbps modems can play. You can simply create a RealAudio clip that consumes 8 Kbps of bandwidth. Anyone with a 14.4 Kbps or higher connection can then listen to your presentation. However, when a clip has multiple streams, such as

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 clip will be playable, you may not like the results. Motion might appear too jerky, for example, or fast-moving images might not resolve visually.

Possible Audio and Video Tracks in a 20 Kbps RealVideo Clip



The point here is that even when you stream just a single clip, 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

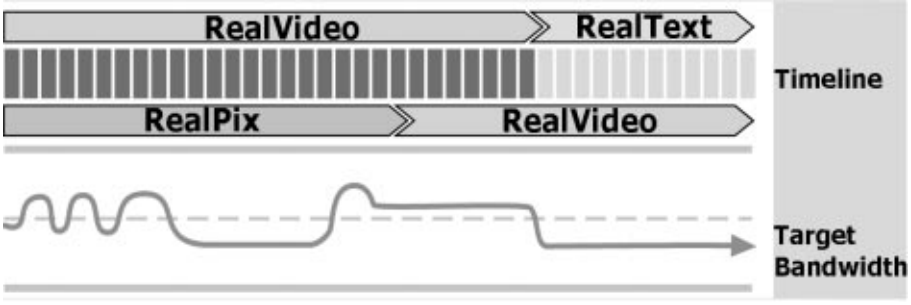
See "Recording Video" on page 53.

Developing Multiclip 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 clip. 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.

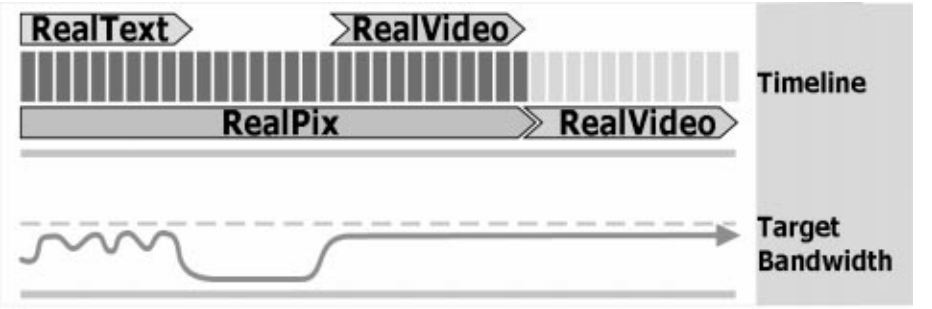
The following figure illustrates a poor presentation for a low-bandwidth connection. The multiclip presentation starts with RealVideo and RealPix clips that require more than the target bandwidth at first. Bandwidth use, illustrated by the solid line, peaks again when the second RealVideo clip begins to play concurrently with the first video clip. To even out the bandwidth spikes, RealSystem streams a high preroll for the RealPix and RealVideo clips, making the presentation slow to start and likely to stall during transitions between clips.

Poor Bandwidth Use in a Multiclip Presentation



The next figure shows better bandwidth management. The presentation starts with a low-bandwidth RealText clip that does not interfere with streaming the images in the RealPix clip. A RealVideo clip starts after the RealPix clip has streamed all its images and does not need bandwidth. The second RealVideo clip starts after the first RealVideo clip has ended, so the two video clips do not compete for bandwidth.

Improved Bandwidth Use in a Multiclip Presentation



Tips for Creating Multiclip Presentations

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 can take advantage of the extra bandwidth to begin streaming higher bandwidth data to RealPlayer "behind the scenes."

  Additional Information

  See "Smoothing Transitions between Clips" on page 163.

- Test presentations in "real world" circumstances, replicating your audience's bandwidth conditions. Clips may play back fine when the files

reside on your desktop computer. The presentation may bog down, though, when you stream the clips over a modem.

Supporting Multiple Bandwidth Connections

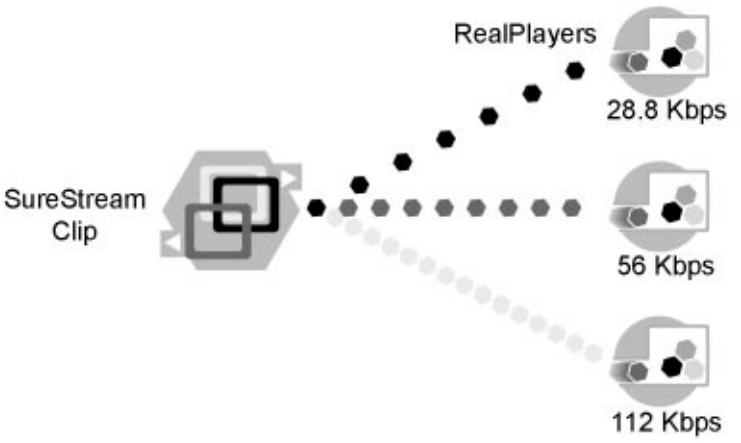
To reach a wide audience on the Internet, you need to provide content that can play over slow connections. You can encode a RealAudio clip at 8 Kbps, for example, so that anyone with a 14.4 Kbps or higher connection can play it. This clip will have good quality sound. But the same clip encoded as a 16 Kbps clip will have better sound. Encoded for 32 Kbps delivery, the clip 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 clip that targets different bandwidths. In the second method, you create separate clips for each bandwidth target and let RealPlayer choose which set of clips to play. Either way, you add to your Web page just one link for all visitors. You don't need separate links for modems and ISDN connections, for example.

Encoding Clips for Multiple Bandwidths

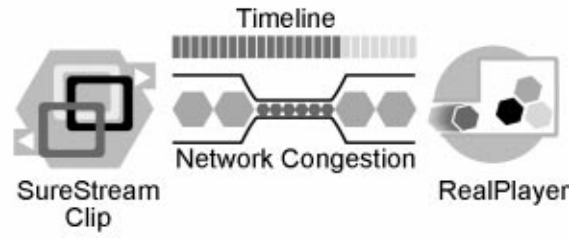
With RealSystem's SureStream technology, you can encode a RealAudio or RealVideo clip for up to six different bandwidths using new RealSystem G2 codecs. For example, you can encode a single RealAudio music clip for 28.8 Kbps modems, 56 Kbps modems, 112 Kbps dual ISDN, and T1 connections. Your Web page links to this single clip and when a visitor clicks the link, RealPlayer and RealServer determine which encoding to use based on the available bandwidth. The following figure illustrates this.

SureStream Clip Encoded for Multiple Bandwidths



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

Switching Bandwidths During Network Congestion



Additional Information

For more on RealAudio and SureStream, see "Choosing RealAudio Codecs" on page 45. See also "Choosing RealVideo Codecs" on page 57.

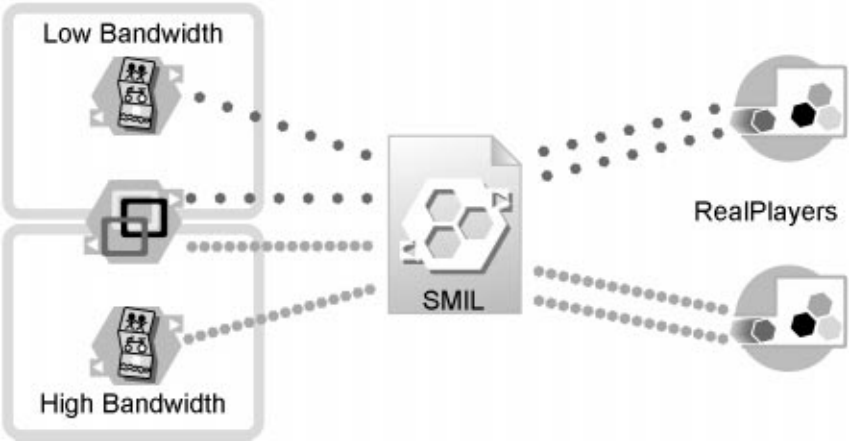
Letting RealPlayer Choose between Multiple Clips

If your presentation uses clips other than RealAudio or RealVideo, you can create multiple versions of the clips for different bandwidths. When you assemble your presentation, you use a SMIL file to designate a bandwidth connection for each of the different groups. When a user clicks your Web page

link, RealPlayer receives the SMIL file and chooses which clip group to play based on its own connection speed.

The following figure illustrates a SMIL file that lists two choices between RealPix clips, a high-bandwidth choice and a low-bandwidth choice. RealPlayer chooses which clip to receive based on its connection speed and the SMIL file's bandwidth parameters. Both presentations use the same RealAudio file, which has been encoded for multiple bandwidths.

Bandwidth Choices through SureStream Clip and SMIL



Because each connection speed uses a different set of clips, RealServer cannot switch between the different encodings as it can with a single, multiply encoded clip. 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 92 explains how to use a SMIL file to designate different bandwidth groups. See also "Switching with SureStream Clips" on page 166.

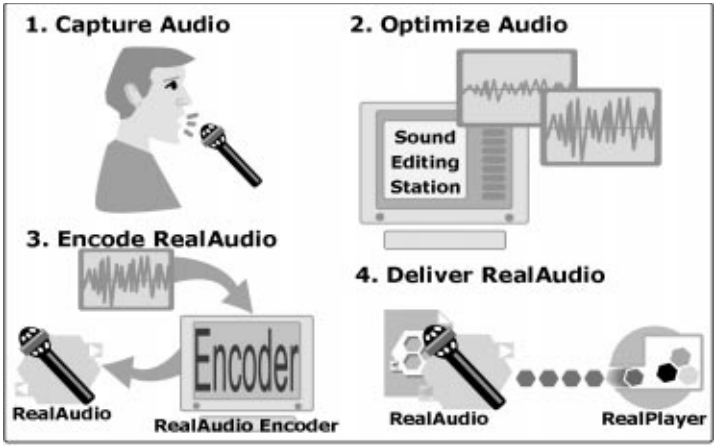
CHAPTER 4: Producing Audio

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, approaching 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 for streaming.

Steps for Streaming RealAudio

To produce a great RealAudio clip, you need to use great source material, high-quality equipment, and good production practices. This section provides a quick overview of the steps involved in streaming a RealAudio clip.

Creating a RealAudio Clip



➤ **To create a streaming RealAudio clip, follow these basic steps:**

1. Capture Audio.

You start audio production by capturing audio from a source, such as a person speaking into a microphone. You might also start with a digitized audio source file from a compact disc, for example.

Additional Information

"Capturing Audio" on page 41 provides guidelines for capturing audio.

2. Optimize audio.

   With the audio file digitized in a common file format, such as WAV or AIFF, you next use a sound editor to optimize the audio for streaming. When broadcasting live, however, you encode the streaming audio directly from the source, optimizing the audio source during the capture.

   Additional Information

   See "Optimizing Audio" on page 43 for tips on sound editing.

3. Encode RealAudio clip.

   With your 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

   "Creating RealAudio Clips" on page 44 explains RealAudio and its codecs. As described in "Streaming Other Audio Formats" on page 49, RealSystem G2 can stream other audio formats as well.

4. Deliver RealAudio clip.

   With your presentation is ready to go, you make your RealAudio clip or broadcast available through your Website. To combine sound with another streaming clip, such as RealPix, you write a SMIL file.

   Additional Information

   Chapter 7 starting on page 75 explains how to create a SMIL file. See Chapter 10 beginning on page 139 for instructions on linking your Web page to a RealAudio or SMIL file. For more on live broadcasting, read Chapter 11 starting on page 153.

Capturing Audio

A streaming audio clip reflects the quality of the audio source. Degradations in sound quality within the audio source affect the final streaming audio clip. The following sections will help you capture high-quality audio source files or broadcasts.

Broadcasting live audio introduces several issues beyond those you need to consider when creating a standard clip. This is because you cannot edit a broadcast the way you can edit a digitized audio file. When broadcasting, though, you can set up your sound equipment to capture high-quality sound before encoding it.

Additional Information

For pointers on producing video, see "Recording Video" on page 53.

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.

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 in the signal chain so that it utilizes its full range of available amplitude without distortion during the program's loudest sections. The signal chain typically includes a microphone, a mixing desk, a compressor, and a sound card. For each piece of equipment, set levels as close as possible to 0 dB without going over.

Check at each point in the signal chain for signal distortion. Perform several test runs and make sure there are no peaks above maximum amplitude. Adjust levels on 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, it is useful to have a dynamics compressor (gain compression, not data compression), which is a piece of audio equipment that automatically adjusts the volume level. By providing a consistent level, the compressor allows you to "set and forget" the input levels to the RealAudio encoder.

Use Optimum RealAudio Sampling Rates

Try to capture sound with a sampling width of 16 bits. RealAudio codecs have different sampling rates that produce the best sound, however. If your sound card allows it, capture audio at the optimum sampling rate for the codec you intend to use. The RealAudio encoder will convert the file to the optimum rate if necessary, but this is recommended only for static files. For live broadcasts, use a sound card that supports the optimum rate. This avoids the overhead of converting the rate while encoding in real-time.

Additional Information

The tables beginning on page 46 list the optimum sampling rates for each codec.

Tip

You do not need to capture stereo sound if you plan to use a mono codec. However, many sound cards simply discard the right input channel in mono mode. If you

have a mixing desk, pan all inputs to the center so the conversion to mono loses nothing.

Optimizing Audio

If you are not broadcasting audio live, you work with a digitized audio source file in a supported format such as WAV, QuickTime, or AIFF. You then edit the audio file to optimize it. To do this, you need to be familiar with the editing functions your audio editing program offers. The following sections give some optimization tips you can carry out with your audio editing software.

DC Offset

DC Offset is 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.

Normalization

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 clips. 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. Because some systems have trouble with files normalized to 100%, normalize to 95% of maximum or to -0.5dB.

Dynamics Compression

Normalization maximizes the input level of the audio file's loudest sections. Consequently, quiet sections may not encode as well. Dynamics compression evens out input levels by attenuating (turning down) the input when it rises above a threshold. Check your audio software for a **Compression** or **Dynamics**

feature. You can control attenuation by specifying a compression ratio. This turns down the loudest sections, and you can readjust input levels accordingly.

Tip

For multipurpose dynamics compression, set the threshold to -10dB, the ratio to 4:1, and the attack and release times to 100ms. Adjust the input level to get around 3dB of compression and an output level around 0dB.

Equalization

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 clip sound as close as possible to the initial recording.

Tip

For voice-only content, you can make the file more intelligible by cutting frequencies below 100 Hz and carefully boosting frequencies in the 1-4 kHz range.

Creating RealAudio Clips

RealAudio is a compressed format suitable for streaming over the Internet or intranets. A RealAudio clip can use either `.ra` or `.rm` as its file extension. 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 clip 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 Format (`.aif`)
- Audio (`.au`)
- MPEG-1 (`.mpg`)
- QuickTime (`.mov`)
- Sound (`.snd`)
- WAV (`.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 clip 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 clip'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 clip. RealPlayer uses the same codec to decompress the streamed RealAudio clip for playback. When you encode a RealAudio clip, you choose a codec (or series of codecs) based on two criteria:

1. Bandwidth

   As Chapter 3 beginning on page 27 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 tables provide a reference for all RealAudio codecs. Note that your encoding tool may not include all codecs listed. The tables give the following information:

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

  An "X" in these columns indicates that a clip encoded with this codec can be played by RealPlayer G2, 5.0, 4.0, and so on. Because of RealSystem's SureStream technology, you can encode a single clip for up to six bandwidths using the RealAudio G2 codecs. Only RealServer G2 can stream SureStream clips, though, as they require a large download time if played back from a Web server.

- Rate

  Using a codec's optimum sampling rate in your audio source file ensures that the audio stays synchronized with other media and prevents pitch shifting in audio resampling. Audio quality degrades if you use lower than

the optimum sampling rate. If you use a higher sampling rate, it is best to use a multiple of the optimum rate. If the optimum rate is 8 kHz, for example, use a higher rate of 16 kHz or 32 kHz.

- Resp.

  This column lists the codec's frequency response in kHz. A codec with a higher frequency response reproduces a wider range of sound than a codec with a lower response. A measure of codec quality, the frequency response does not affect how you produce audio.

RealAudio Low Bandwidth Codecs

RealAudio Codec	G2	5	4	3	2	1	Rate	Resp.	Comments
5 Kbps Voice	X	X	–	–	–	–	8 kHz	4 kHz	Lowest bit rate codec for speech or speech with background music.
6.5 Kbps Voice	X	X	X	–	–	–	8 kHz	4 kHz	Low bit rate codec for speech or speech with background music.
6 Kbps Music–G2 Mono	X	–	–	–	–	–	8 kHz	3 kHz	Use with SureStream clips.
8 Kbps Voice	X	X	X	X	X	X	8 kHz	4 kHz	Original voice codec. Superseded by 8.5Kbps Voice.
8 Kbps Music–G2 Mono	X	–	–	–	–	–	8 kHz	4 kHz	Use with SureStream clips.
8 Kbps Music	X	X	X	–	–	–	8 kHz	4 kHz	DolbyNet codec.
8.5 Kbps Voice	X	X	X	–	–	–	8 kHz	4 kHz	High-quality voice codec for voice or voice with background music.
11 Kbps Music–G2 Mono	X	–	–	–	–	–	11.025 kHz	5 kHz	Use with SureStream clips.
12 Kbps Music	X	X	X	–	–	–	8 kHz	4 kHz	DolbyNet codec.

RealAudio Medium Bandwidth Codecs

RealAudio Codec	G2	5	4	3	2	1	Rate	Resp.	Comments
15.2 Kbps Voice	X	X	X	X	X	–	8 kHz	4 kHz	Superseded by 16 Kbps Voice codec.
16 Kbps Voice–Mono	X	X	–	–	–	–	16 kHz	7 kHz	High-quality wideband for voice or voice with background music.
16 Kbps Music–G2 Mono	X	–	–	–	–	–	22.05 kHz	8 kHz	Use with SureStream clips.

(Table Page 1 of 2)

RealAudio Medium Bandwidth Codecs (continued)

RealAudio Codec	G2	5	4	3	2	1	Rate	Resp.	Comments
16 Kbps Music–Mono Low	X	X	X	X	–	–	8 kHz	4 kHz	DolbyNet codec. Low response.
16 Kbps Music–Mono Medium	X	X	X	X	–	–	11.025 kHz	4.7 kHz	DolbyNet codec for pop/rock music. Medium response.
16 Kbps Music–Mono High	X	X	X	X	–	–	11.025 kHz	5.5 kHz	DolbyNet codec for classical music. High response.
20 Kbps Music–G2 Mono	X	–	–	–	–	–	22.05 kHz	10 kHz	Use with SureStream clips.
20 Kbps Music–G2 Stereo	X	–	–	–	–	–	11.025 kHz	5 kHz	Use with SureStream clips.
20 Kbps Music–Stereo	X	X	X	X	–	–	8 kHz	4 kHz	DolbyNet codec.

(Table Page 2 of 2)

RealAudio High Bandwidth Codecs

RealAudio Codec	G2	5	4	3	2	1	Rate	Resp.	Comments
32 Kbps Voice–G2 Mono	X	–	–	–	–	–	22.05 kHz	11 kHz	Use with SureStream clips.
32 Kbps Music–G2 Mono	X	–	–	–	–	–	44.1 kHz	16 kHz	Use with SureStream clips.
32 Kbps Music–G2 Stereo	X	–	–	–	–	–	44.1 kHz	8 kHz	Use with SureStream clips.
32 Kbps Music–Mono	X	X	X	–	–	–	16 kHz	8 kHz	DolbyNet codec.
32 Kbps Music–Stereo	X	X	X	–	–	–	11.025 kHz	5.5 kHz	DolbyNet codec.
40 Kbps Music–Mono	X	X	X	X	–	–	22.05 kHz	11 kHz	DolbyNet codec.
40 Kbps Music–Stereo	X	X	X	X	–	–	16 kHz	8 kHz	DolbyNet codec.
44 Kbps Music–G2 Mono	X	–	–	–	–	–	44.1 kHz	20 kHz	Use with SureStream clips.
44 Kbps Music–G2 Stereo	X	–	–	–	–	–	44.1 kHz	11 kHz	Use with SureStream clips.
64 Kbps Voice–G2 Mono	X	–	–	–	–	–	44.1 kHz	20 kHz	Use with SureStream clips.

(Table Page 1 of 2)

RealAudio High Bandwidth Codecs (continued)

RealAudio Codec	G2	5	4	3	2	1	Rate	Resp.	Comments
64 Kbps Music–G2 Mono	X	–	–	–	–	–	44.1 kHz	20 kHz	Use with SureStream clips.
64 Kbps Music–G2 Stereo	X	–	–	–	–	–	44.1 kHz	16 kHz	Use with SureStream clips.
80 Kbps Music–Mono	X	X	X	X	–	–	44.1 kHz	20 kHz	DolbyNet codec.
80 Kbps Music–Stereo	X	X	X	X	–	–	32 kHz	16 kHz	DolbyNet codec.
96 Kbps Music–G2 Stereo	X	–	–	–	–	–	44.1 kHz	24 kHz	Use with SureStream clips.

(Table Page 2 of 2)

Encoding RealAudio with RealSystem Tools

When you encode RealAudio clips 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 clip, 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/products/tools/**.

Retain Source Files

Always keep a copy of the original audio source file. To edit the RealAudio clip or encode it with a different codec, modify the source file as necessary, then encode the file again as RealAudio. You cannot convert RealAudio clips to other audio formats.

Using RealAudio in a MultiClip Presentation

When you encode a RealAudio clip, 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 clip

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 Clip

You can create a single RealAudio clip encoded for up to six bandwidths with SureStream technology introduced in RealSystem G2. You can also specify backwards compatibility with earlier versions of RealPlayer. The encoding tool then encodes the clip for your selected bandwidths with the new RealAudio G2 codecs. It also includes in the clip an encoding that uses an older codec and targets the lowest bandwidth choice.

For example, you can encode a single clip at 8, 16, and 32 Kbps using RealAudio G2 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 8, 16, or 32 Kbps RealAudio G2 stream. Earlier versions of RealPlayer receive the 8 Kbps stream encoded with the older codec regardless of their connection speeds.

Note

To support multiple bandwidths with codecs other than RealAudio G2 codecs, you must encode a separate clip with each codec. You then use SMIL to specify bandwidth choices. For more on bandwidth selection through SMIL, see "Setting Bandwidth Choices" on page 92.

Batch Encoding

Your encoding tool may have a batch mode that lets you encode several clips at once. The batch encoder may run through a command-line interface rather than a graphical user interface.

Streaming Other Audio Formats

RealSystem can stream several audio formats in addition to RealAudio. The following table lists the streamable formats and shows whether RealPlayer G2 for a different operating systems (Windows 95 or NT, Macintosh, and UNIX) can play the audio. RealSystem typically does not stream audio formats that have been compressed with a codec. Where codec compression is supported,

codecs are not included with RealPlayer G2 and must reside on the playback machine already.

Streamable Audio Formats

Format	Codec compression	Win32	Mac	UNIX
AIFF (.aif)	uncompressed	**yes**	**yes**	**yes**
AU (.au)	uncompressed	**yes**	**yes**	**yes**
WAV (.wav)	compressed	**yes**	no	no
	uncompressed	**yes**	**yes**	**yes**

Note

RealSystem plug-ins may exist for additional audio formats. Check **http://www.real.com** for information about other audio formats you can stream.

Tips for Streaming other Audio Formats

Observe the following points when streaming audio formats other than RealAudio:

- The audio formats listed above may not work for low bandwidth connections. Test bandwidth consumption by opening the file in RealPlayer G2 and watching the status panel, which will list the streaming bit rate. If the rate is too high, convert the file to RealAudio.
- A-law compression is not supported for the AU format. It is supported for WAV, however.
- Most uncompressed WAV files work on Macintosh and Unix. Some may not, however. Before streaming a WAV file, test playback locally by opening the file with RealPlayer G2 on these platforms.

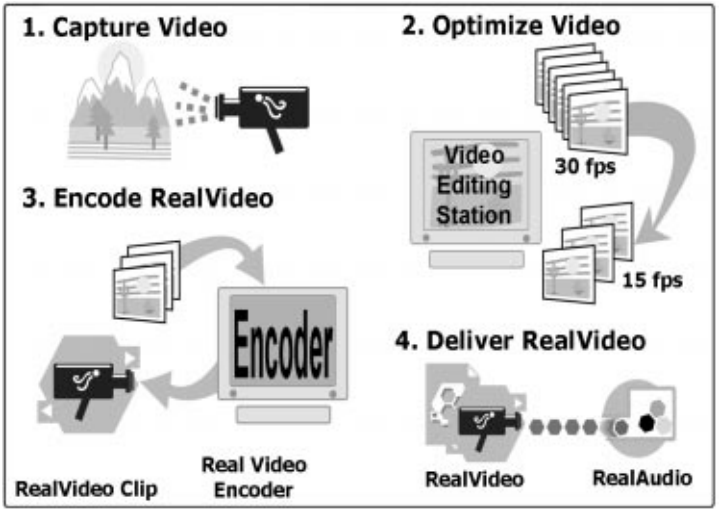
CHAPTER 5: Producing Video

RealNetworks introduced RealVideo with RealSystem 4.0, making streaming video available over the Internet. This chapter covers RealVideo production and explains other video formats that RealSystem G2 can stream. It also provides tips for capturing high-quality video and working with digitized video source files.

Steps for Streaming RealVideo

When producing a RealVideo clip, you should choose the best source material and best equipment possible. The goal throughout the video production process is to get the best video quality with the smallest streaming file size.

Creating RealVideo Clips



➤ **The following steps summarize how to create a RealVideo clip:**

1. Capture video.

To start video production, you capture video by, for example, shooting a tape with a camera, or gathering recorded content from tape, satellite, laserdisc, or so on.

Additional Information

"Recording Video" on page 53 provides guidelines for capturing video.

2. Digitize and edit video.

   You next digitize the video to a standard file format, such as AVI or QuickTime. With your preferred video editing software, you can then edit the video as necessary. If you are broadcasting live, however, you encode the streaming video directly from the source.

   Additional Information

   See "Digitizing Video" on page 54 for tips on video editing.

3. Encode RealVideo clip.

   With your digitized file optimized or your live broadcast ready to go, you encode your source as RealVideo. When you do this, you target a network bandwidth or set of bandwidths and choose a RealAudio codec or set of codecs for the audio track.

   Additional Information

   See "Creating RealVideo Clips" on page 57 for more on RealVideo production. "Choosing RealAudio Codecs" on page 45 explains RealAudio codecs.

4. Deliver RealVideo clip.

   With your presentation ready to go, you make your RealVideo clip or broadcast available through your Website. If you are combining video with another streaming clip, such as RealText, you write a SMIL file that assembles the pieces.

   Additional Information

   Chapter 7 starting on page 75 explains how to create a SMIL file. See Chapter 10 beginning on page 139 for instructions on linking your Web page to a RealVideo clip or a SMIL file. For more on live broadcasting, read Chapter 11, starting on page 153.

Recording Video

Observe the following points if you intend to shoot a new video rather than use existing video content. Because video loses image quality when compressed for streaming, always start with the best video source available.

Additional Information

For pointers on producing audio, see "Capturing Audio" on page 41.

Use a High-Quality Source Format

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

1. Betacam SP, also known simply as Beta. This format is common among video production professionals.
2. DV, miniDV, DVCam, or DVCPro.
3. Super-VHS (S-VHS) or HI-8mm.
4. VHS, 8mm.

Stage According to the Video's Final Size

It's important to consider the video's final frame size before you shoot the first frame. To reach Web users with 28.8 Kbps modems, you should produce a video approximately 176x132 pixels. With this small size, you need to frame important visual elements well.

Additional Information

For more on window size, see "Digitizing Video" on page 54.

Minimize Scene Changes and Movement

The less that changes from frame to frame, the sharper the image will appear in the encoded video. RealVideo compression technology reuses existing data when frames are similar, so a video with relatively stationary subjects ("talking heads") will look better 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 a rapidly moving object 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 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 Uniform Colors and Good Lighting

Bright lighting at a constant exposure keeps the foreground detail crisp. Use uniform, dark colors for backgrounds, and uniform, light colors (but not whites) for clothing. Complex textures such as paisley and stripes reduce the final image quality. They can also degrade the video with unwanted visual effects.

Digitizing Video

If you are not broadcasting RealVideo live, you digitize the source video (tape, disc, satellite feed, or so on) on your computer or video editing station. You can then edit the file with your preferred video editing software before encoding it as RealVideo. The following sections provide tips for capturing video.

Capture AVI on Windows or QuickTime on Macintosh

It is better to work with uncompressed formats. Otherwise, you compress the source once when you digitize it and again when you encode it as RealVideo. This double compression can decrease the image quality. Use a compressed source format only if your RealVideo encoding tool supports the file as input. You can use compressed AVI as long as the RealVideo encoding machine has the same Video for Windows (VFW) driver used to compress the AVI.

Additional Information

"Creating RealVideo Clips" on page 57.

Capture at a 320x240 Pixel Window Size, then Reduce

Unless you are short on disk space or your video capture card recommends a different window size, capture video in a window 320 pixels wide by 240 pixels

high. With your video editing software, you can then reduce the window size before encoding the RealVideo clip.

The following table, which you should use as a general guideline only, shows three common RealVideo clip sizes that maintain the 4:3 aspect ratio used in television. You can produce RealVideo at any width-to-height ratio, however.

RealVideo Size and Bandwidth Guidelines at 10-15 fps

Width x Height Pixel Size	RealVideo Bandwidth	Image Quality
176x132	20-500 Kbps	Good to High
240 x 180	100-500 Kbps	High
320 x 240	200-500 Kbps	High

For each window size, the table lists bandwidth guidelines that assume a desired frame rate of 10 to 15 fps. The table shows that for 28.8 Kbps modems (20 Kbps usable bandwidth), you should use a window no larger than 176x132. Although its overall quality will be good, the image may encode at less than 10 fps. You'll get better quality when encoding for higher bandwidths.

Keep in mind that you *can* create a 240x180 window for bandwidths lower than 100 Kbps, or a 320x240 video for bandwidths lower than 200Kbps. But you're not likely to get high image resolution and 10-15 fps. The video might encode at 5-7 fps, for example, which may be acceptable for your audience. Experiment with different sizes, playing the encoded clip back in RealPlayer G2 to test its quality.

Tip

With your RealVideo encoding tool, you can choose to encode a large video for low bandwidths as a "slide show," which streams approximately one frame per second.

Additional Information

See Chapter 3 beginning on page 27 for details on targeting bandwidth.

Capture at 15 Frames per Second (fps)

Although RealVideo can encode up to 30 fps, the maximum recommended frame rate for streaming video over a network is 15 fps. Higher frame rates may cause playback problems on slower machines. RealVideo encoding tools

use a frame rate that varies with the bandwidth target and the emphasis on smooth motion or image quality. With video source at 15 fps, your encoding tool can vary the RealVideo clip frame rate between 1 and 15 fps as needed.

Use 24-bit Color Depth

Always use 24-bit color. Lower color resolution produces poor quality videos.

Ensure Enough Disk Space for Digitized Video

Use this formula to calculate the approximate size in Megabytes of a digitized video file:

$$\frac{\text{(pixel width) x (pixel height) x (color bit depth) x (fps) x (duration in seconds)}}{\text{8,000,000}}$$

Suppose you want to capture a three minute video at 15 frames per second with 24-bit color in a window that is 320x240 pixels. Your digitized source file would be approximately 622 Megabytes:

```
(320) x (240) x (24) x (15) x (180) / 8,000,000 = 622 Megabytes
```

If necessary, you can conserve disk space by decreasing the window size or lowering the frame rate.

> **Tip**
>
> Windows 95 and Macintosh operating systems limit a single video file to 2 Gigabytes (2048 Megabytes). At a 320x240 window size and 15 fps, this translates to about 9.5 minutes of video. To work around this, create separate source files, encode them as separate RealVideo clips, and merge the RealVideo clips with RealVideo editing tools available from **http://www.real.com**.

Use a Fast Machine

Video capture places a large burden on a computer's CPU and hard drive. Be sure to have a fast machine. To avoid dropping frames during video capture, use a hard drive specially made for audio and video work. On Windows machines, you can use any video capture card that supports Video for Windows.

Use S-video Output

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

Creating RealVideo Clips

RealVideo's high compression rate makes it well-suited for streaming video over the Internet or intranets. A RealVideo clip uses the file extension .rm (or .rv) and typically includes an embedded soundtrack encoded as RealAudio. You start with a video file in a digitized format, then encode a RealVideo clip from this source file using a RealVideo encoding tool. Your encoding tool should be able to accept at least one of these input formats:

- AVI (.avi), uncompressed (recommended) or compressed
- MPEG-1 (.mpg)
- QuickTime 2.0 (.mov), uncompressed

Additional Information

See "RealNetworks Encoding Tools" on page 15 for more on RealVideo encoding tools available from RealNetworks at **http://www.real.com**.

Choosing RealVideo Codecs

Like RealAudio, RealVideo uses a "lossy" compression scheme that discards parts of the source file during encoding. The following table provides a reference for RealVideo codecs. An "X" in the G2, 5, or 4 column indicates that a clip encoded with this codec can be played by RealPlayer G2, 5.0, or 4.0. Earlier versions of RealPlayer do not play RealVideo.

RealVideo Codecs

RealVideo Codec	G2	5	4	Comments
G2 Standard (new default)	X	–	–	G2 codec for fast encoding and encoding a single clip for multiple bandwidths using SureStream.
Standard (old default)	X	X	X	Standard codec used in past releases. No multiple encoding in a single clip.
Fractal (discontinued)	X	X	X	Discontinued in RealSystem G2 encoders. RealPlayer G2 can play existing fractal encoded clips, though.

Encoding RealVideo with RealSystem Tools

When you encode RealVideo, you choose an overall bandwidth target or set of targets, then set parameters such as audio type (voice or music), compatibility with earlier versions of RealPlayer, and an emphasis on smooth motion or

sharp images. After you make your choices, the encoding tool chooses the audio codec or codecs to use and encodes the clip using a variable frame rate. The following sections give tips on using RealSystem encoding tools.

Additional Information

See the tool's manual or online help for step-by-step instructions on encoding RealVideo, as well as more information about encoding tool options. RealVideo encoding tools are available for purchase or free download at **http://www.real.com/products/tools/**.

Retain Source Files

Always keep a copy of the original video source file. To edit the RealVideo clip, modify the source file as necessary, then encode the file again as RealVideo. You cannot convert RealVideo clips to other video formats.

Using RealVideo in a MultiClip Presentation

When you encode a RealVideo clip, consider whether it will play in parallel with another clip. If so, you can set an option in your encoding tool so that the RealVideo clip does not consume all available bandwidth. For a 28.8 Kbps modem connection, for example, the encoder standardly encodes the clip to use 20 Kbps. You can alter this default value so that the video uses 12 Kbps, for example, leaving bandwidth free for another clip.

Additional Information

Chapter 3 beginning on page 27 explains more about target bandwidths.

Multiple Encoding in a Single Clip

You can create a single RealVideo clip encoded for up to six bandwidths with RealSystem's SureStream technology, which uses the new RealSystem G2 video and audio codecs. You can also specify backwards compatibility with RealPlayer 5.0. The encoding tool then encodes the clip for your selected bandwidths with the RealVideo and RealAudio G2 codecs. It also includes in the clip an encoding that uses a codec compatible with RealPlayer 5.0 and targets the lowest bandwidth choice.

For example, you can encode a single clip for the target connections shown in the table below. Based on its connection speed, RealPlayer G2 receives the SureStream 20, 32, 45, or 80 Kbps stream. Earlier versions of RealPlayer receive the 20 Kbps backwards-compatible stream regardless of their connection

speeds. Note that the RealAudio codec choices shown below are examples only. The encoding tool may choose other codecs depending on a voice or music soundtrack, and which features of the video you want to emphasize.

RealVideo Multiply Encoded Clip Example

Target Connection	Clip Bit Rate	RealAudio Codec	RealVideo Bandwidth
28.8 Kbps modem (backwards compatible)	20 Kbps	8 Kbps Music	12 Kbps
28.8 Kbps modem	20 Kbps	8 Kbps Music–G2 Mono	12 Kbps
56.0 Kbps modem	32 Kbps	11 Kbps Music–G2 Mono	21 Kbps
56.0 Kbps ISDN	45 Kbps	20 Kbps Music–G2 Stereo	25 Kbps
112 Kbps dual ISDN	80 Kbps	44 Kbps Music–G2 Stereo	36 Kbps

Additional Information

For information on RealAudio codecs, see "Choosing RealAudio Codecs" on page 45.

Note

To support multiple bandwidths without using SureStream and the new RealSystem G2 codecs, you must encode a separate clip with each codec. You then use a SMIL file to specify bandwidth choices. For more on bandwidth selection through SMIL, see "Setting Bandwidth Choices" on page 92.

Video Window Cropping

When you encode with the new RealVideo standard codec, the encoding tool crops the video window to multiples of 4 pixels, so a 176x132 video stays its original size. With the older standard codec, the tool crops to multiples of 16 pixels, however, so a 176x132 video encodes at 176x128 pixels. Encoding tools also let you manually crop the image window to leave areas of the source video out of the RealVideo clip.

High-Bandwidth Clips for Fast Machines

RealVideo clips encoded for 200 Kbps or higher bandwidths may have image quality and frame rates that overburden processors slower than 120 Mhz. You should therefore use high-bandwidth RealVideo only for fast machines with fast connections.

Batch Encoding

Your encoding tool may have a batch mode that lets you encode several clips at once. The batch encoder may run through a command-line interface rather than a graphical user interface.

Streaming Other Video Formats

RealSystem can stream several video formats in addition to RealVideo. The following table lists the streamable formats and shows whether RealPlayer G2 for different operating systems (Windows 95 or NT, Macintosh, and UNIX) can play back the video. RealSystem supports all Vivo codecs. For other video formats, however, it typically does not stream codec-compressed files. When codec compression other than Vivo is supported, codecs are not included with RealPlayer G2 and must reside on the playback machine already.

Streamable Video Formats

Format	Codec compression	Win32	Mac	UNIX
ASF 1.0 (.asf)	compressed	**yes**	no	no
	uncompressed	**yes**	**yes**	**yes**
AVI 1.0 (.avi)	compressed	**yes**	no	no
	uncompressed	**yes**	**yes**	**yes**
QuickTime 2.0 (.mov)	compressed	no	no	no
	uncompressed	**yes**	**yes**	**yes**
Vivo (.viv)	compressed	**yes**	**yes**	**yes**

Note

RealSystem plug-ins may exist for other video formats. Check **http://www.real.com** for information about additional video formats you can stream.

Tips for Streaming other Video Formats

Observe the following points when streaming video formats other than RealVideo:

- Because other video formats may not be as highly compressed as RealVideo, they may not be good for low bandwidth connections. Test bandwidth needs by opening the file in RealPlayer G2 and watching the

RealPlayer status panel, which will list the streaming bit rate. If the rate is too high for your target audience, use RealVideo instead.

- For the ASF and QuickTime formats, RealSystem ignores any tracks other than video or audio. It disregards scripting commands, for example.
- For high-bandwidth connections or videos that play back from a user's local disk, CPU power is a consideration. Play the file in RealPlayer G2 and monitor CPU use with the tool of your choice. Decide whether an "average" machine for your target audience can handle the load. If CPU use is too high, convert the video to a high-bandwidth RealVideo clip. This creates a high-quality video with a lower CPU requirement.

CHAPTER 6: Producing Animation

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

Introducing 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.

Additional Information

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

Tools for Creating RealFlash

You need the following tools to create and stream RealFlash:

- Macromedia Flash 2.0 or 3.0

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

  Additional Information

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

  Note

  RealFlash supports Flash 2.0 but does not support Flash 3.0 transparency.

- Sound capture and editing tools

  You should use professional 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 39.

- RealNetworks Production Tools

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

- RealFlash Tuner and Spreadsheet

  The RealFlash Bit Rate Calculation Spreadsheet and the RealFlash Bandwidth Tuner help you create and optimize your clip. They are included in the utils directory of the HTML version of this manual. To get these tools, download the bundled HTML version of this manual from **http://www.real.com/**. You will need Microsoft Word and Microsoft Excel to use the spreadsheet.

- RealServer G2 and RealPlayer G2

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

Preparing a RealFlash Clip

A RealFlash clip 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 clip to RealPlayer, ensuring that animation and sound stay synchronized.

RealFlash Clips Consist of Flash and RealAudio

Choosing a Target Bandwidth

When you begin to develop your RealFlash clip, 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 27.

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 possible 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 for RealPlayer G2 and 5.0	5 Kbps Voice	15 Kbps
Emphasis on animation with better quality spoken soundtrack for RealPlayer G2 and 5.0	6.5 Kbps Voice	13.5 Kbps
Emphasis on animation with high quality spoken soundtrack for RealPlayer G2 and 5.0	8.5 Kbps Voice	11.5 Kbps
Emphasis on animation with good quality music soundtrack for RealPlayer G2 and 5.0	8 Kbps Music	12 Kbps
Emphasis on animation with good quality music soundtrack for RealPlayer G2 only	6 Kbps Music–G2 Mono	14 Kbps
Emphasis on animation with higher quality music soundtrack for RealPlayer G2 only	8 Kbps Music–G2 Mono	12 Kbps
Emphasis on high quality music soundtrack with animation for RealPlayer G2 only	11 Kbps Music–G2 Mono	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

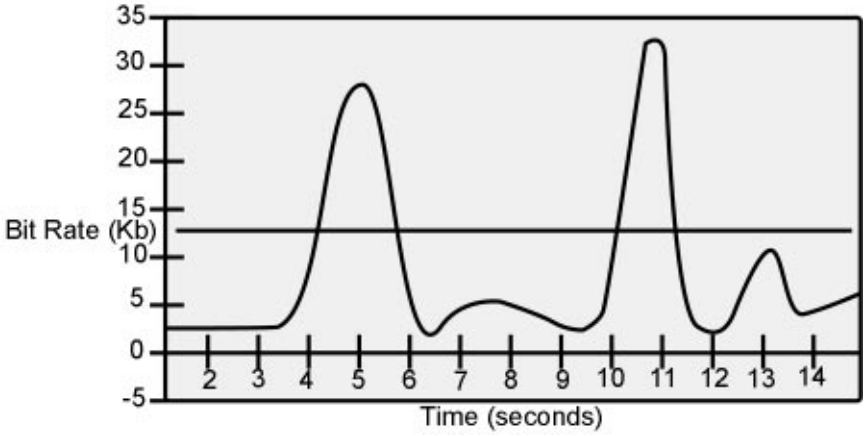
See “Choosing RealAudio Codecs” on page 45 for a full list of codecs. You can also use SureStream technology

and the RealSystem G2 codecs to encode a single RealAudio file for multiple bandwidths.

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 clip that targets a bandwidth of 12 Kbps. At five and ten seconds into the clip, the bandwidth requirement spikes because the clip requires 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 clip 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 Flash tuning utility to change the streaming file's bandwidth consumption. You can also use the RealFlash Bit Rate Calculation Spreadsheet to view frame-by-frame bandwidth needs.

Note

These tools are included in the utils directory of the HTML version of this manual. To get these tools, download the bundled HTML version of this manual from **http://www.real.com/**.

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.

- Minimize morphing

  The Flash 3.0 morphing feature creates additional key frames and hence increases the file size and streaming bit rate.

- 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 clip, 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`	Clip pauses until action is performed or Play button pressed.
`Goto`	`Seek and Pause`	RealPlayer seeks to the designated frame, buffers the clip preroll, and pauses.
`Goto and Play`	`Seek and Play`	RealPlayer seeks to the designated frame, buffers the clip 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 clip.

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 Clip

The following sections explain the steps for creating a RealFlash clip. 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 clips, 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 clip for bandwidth efficiency.

The ratio of Shockwave file size to clip 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 clip 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 must use the RealFlash Bandwidth Tuner to view bandwidth statistics and adjust the file's streaming bit rate. You can also use the RealFlash Bit Rate Calculation Spreadsheet to examine the file's bandwidth consumption frame-by-frame.

Note

These tools are included in the `utils` directory of the HTML version of this manual. To get these tools, download the bundled HTML version of this manual from **http://www.real.com/**. See the tuner online help for instructions on using the tuner and spreadsheet.

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 clip'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 G2 music codec is 6 Kbps. The lowest bit rate for a music codec playable by RealPlayer G2 and RealPlayer 5.0 is 8 Kbps.
- To ensure a high-quality visual clip, 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 45 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 75 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>
```

Moving Files to RealServer

When the clip is ready, 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 139.

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 clips 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 139.

Tip

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

Creating a SMIL File

You can create a SMIL 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 clips 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>
```

- A SMIL file can include an optional header section defined by <head> and </head> tags. It requires a body section defined by <body> and </body> tags:

```
<smil>
  <head>
    ...optional section with all header mark-up...
  </head>
  <body>
    ...required section with all body mark-up...
  </body>
</smil>
```

- 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. File names in SMIL must reflect the file name exactly as it appears on the server. They can use upper, lower, or mixed case.
- Save your file with the 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 78.
- 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. Although not required, including a header is highly recommended:

```
<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 Clip Locations" on page 78):

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

When the second clip 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 clips by including the author, title, and copyright attributes in <seq> and <par> tags. For more on these tags, see "Grouping Clips" on page 82.

Tip

Title, author, and copyright information you set in a SMIL file overrides any such information encoded in a clip. It's good practice to add the information to the clip as well, however, in case you want to serve the clip without using a SMIL file.

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 codes 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 Clip Locations

To add a clip to the presentation, you include in the SMIL body section a clip source tag that describes the clip type and location:

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

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

Source Clip Attributes

Clip Attribute	Used For
`animation`	Animation clips, such as Shockwave Flash (`.swf`) used in RealFlash.
`audio`	Audio clips such as RealAudio (`.ra` or `.rm`).
`img`	JPEG (`.jpg`) or GIF images (`.gif`). See also "Defining Image Options" on page 109.
`ref`	Any clip type not covered by other attributes, such as a RealPix file (`.rp`).
`text`	Static text clips (`.txt`).
`textstream`	Streaming RealText clips (`.rt`).
`video`	Video or other clips that display continuous motion, such as RealVideo (`.rm`).

Additional Information

For a list of supported streaming formats, see "Choosing Media to Stream" on page 11.

Although a source clip tag must start with a clip type attribute, attributes do not affect playback because RealPlayer determines the clip type through other means. Specifying a text clip as `audio`, for example, does not adversely affect playback. Although using the different clip attributes helps you keep track of the media types, you could specify all clips with `ref`, for example.

After the clip type attribute, the `src` attribute lists the clip location. How you specify this location depends on whether you will stream the presentation with RealServer, download the clip from a Web server, or play clips back from a local computer.

Linking to Clips on RealServer

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

```
<audio src="rtsp://realserver.company.com:554/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.

RealServer URL Components

Component	Meaning
rtsp://	RTSP streaming protocol. In contrast, URLs in Web pages start with http://. Although RealServer also supports HTTP, RealServer clips should always use RTSP.
realserver.company.com	RealServer address. This 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.
:554	RealServer port for RTSP connections. Port 554 is the default, so you can leave this out of URLs unless the RealServer administrator changed the RTSP port. If the port number is required, separate it from the address with a colon.
/audio/	RealServer directory that holds the clip. The directory structure may be several levels deep.
first.ra	Clip file name.

Creating Relative URLs

If your presentation includes many clips that are on the same server, 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 clip has a full URL specified for it, the base target is ignored. For the first two clips, however, the src values are appended to the base target, effectively giving the clips these URLs:

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

If no base target is given, RealPlayer assumes that the clip paths are relative to the location of the SMIL file. In the example above, therefore, you could leave the base target out if the SMIL file itself resides in a directory that contains the audio directory that in turn holds the RealAudio clips.

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 guide.

Linking to Clips on a Web Server

To use a clip hosted on a Web server, use a standard HTTP URL such as the following in a clip source tag:

```
<img src="http://www.company.com/images/logo.gif"/>
```

Although a Web server can host any clip, a Web server cannot perform all the functions of RealServer.

Additional Information

See "Hosting a Presentation on a Server" on page 21 and "Limitations on Web Server Playback" on page 145.

Linking to Local Clips

If your presentation clips 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 clips in this format:

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

This example is a local, relative link to a clip 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 clips 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 or launch the presentation through a RAM file as described in "Creating a RAM File Manually" on page 147. This problem does not occur with Netscape Navigator or Internet Explorer 4.0. Nor does it occur when you stream clips from a server to Internet Explorer 3.0.

Grouping Clips

With the SMIL <seq> and <par> tags, you can create groups that structure your presentation. The following sections explain how to play clips in sequence or parallel, as well as how to repeat groups or clips within groups.

Playing Clips 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 clips above, you wouldn't need to use the <seq> tag. You could simply list the clips 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 Clips 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>
```

When you play clips in parallel, be sure that they do not exceed the presentation bandwidth as described in "Choosing a Target Bandwidth" on page 29. When RealServer G2 streams parallel groups, it ensures that the clips stay synchronized. If some video frames don't arrive, for example, RealServer either drops those frames or halts playback until the frames arrive.

Additional Information

When multiple clips play in parallel, you need to define each clip's playback region. For more information, see "Laying Out Multiple Clips" on page 93.

Ending a Parallel Group on a Specific Clip

By default, a <par> group ends once all clips finish playing back. You can modify this with an end timing value as described in "Setting Begin and End Times" on page 87. Or you can use the endsync attribute to stop the group when a specific clip finishes playback:

```
<par endsync="first">
  ...
</par>
```

first

Use the attribute endsync="first" to stop the <par> group the first time a clip in the group ends playback. All other clips in the group stop playing at that point regardless of their playback status and any timing parameters specified for them.

last

The attribute endsync="last" causes the <par> group to conclude when all clips have finished playing. Because this is the default value, you can omit the endsync attribute from the <par> tag to achieve this effect.

id(clip id)

The attribute endsync="id(*clip id*)" causes the <par> group to conclude when the specified clip reaches its end. All other clips in the group stop playing at that point regardless of their playback status and any timing parameters specified for them. The specified clip must have a corresponding id value in its source tag:

```
<par endsync="id(vid)">
  <video id="vid" src="videos/newsong.rm"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

id(clip id) (time)

The attribute endsync="id(*clip id*) (*time*)" makes the <par> group conclude at a given time after the specified clip reaches its end. The clip must have a corresponding id value in its source tag. The following example ends the <par> group 12 seconds after the text stream finishes:

```
<par endsync="id(text) (12s)">
  <video id="vid" src="videos/newsong.rm"/>
  <textstream id="text" src="lyrics/newsong.rt"/>
</par>
```

Additional Information

"Specifying Timing" on page 86 gives the timing shorthand values you can use.

Repeating a Clip or Group

The repeat="n" attribute makes a clip or group play a specified number of times. You can add it to a clip source tag, for example:

```
<video src="videos/newsong.rm" repeat="4"/>
```

You can also add it to a <seq> or <par> group to make the entire group repeat:

```
<par repeat="3">
  <video src="videos/newsong.rm"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

In the following example, a video clip within a <par> group plays twice. Assuming the video lasts longer than the text stream, the <par> group ends when the video concludes its second playback:

```
<par>
  <video src="videos/newsong.rm" repeat="2"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

In the following <seq> group, the second video plays only after the first video plays twice:

```
<seq>
  <video src="videos/newsong.rm" repeat="2"/>
  <video src="videos/newsong2.rm"/>
</seq>
```

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:

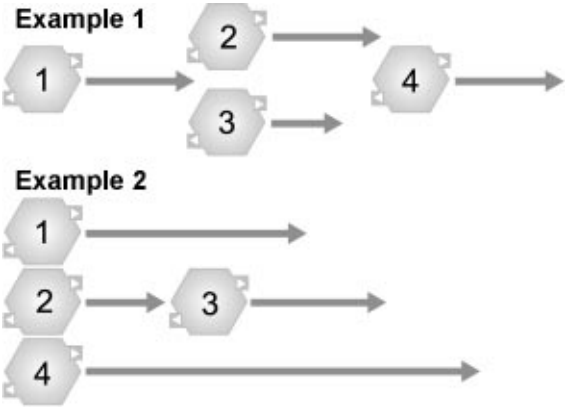
```
<seq>
   clip 1
   <par>
     clip 2
     clip 3
   </par>
   clip 4
</seq>
```

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

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

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

Different Playback Results with <seq> and <par> Groups



Specifying Timing

Timing elements let you specify when a clip or group 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

Decimal values are not required. You can express two seconds as `"2s"` or `"2.0s"`, for example. You can also express complex time elements in a "normal play time" format that includes an `npt=` prefix, as shown in this example:

```
begin="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:

```
begin="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:

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

Setting Begin and End Times

The begin attribute works for any clip or group. You can use it to start a clip at a specific point within the timeline:

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

Were this clip in a <par> group, the begin attribute would start the clip playing back 20.5 seconds after the group becomes active. Timing is relative to the start of the <seq> or <par> group, therefore, not the start of the overall presentation. If the clip were in a <seq> group, the attribute would delay the clip's normal playback by inserting 20.5 seconds of blank time before the clip starts.

Additionally, you can set an end attribute alone or combined with a begin attribute as shown here:

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

In this example, the clip ends at 62.7 seconds into its part of the presentation timeline, playing a total of 42.2 seconds regardless of the length of its internal timeline. If the video's internal timeline is shorter (30 seconds, for example) the fill attribute determines what happens onscreen after the video stops playing but before the end time is reached.

Additional Information

"Setting a Fill" on page 89.

Using Begin and End Times with Groups

You can use begin or end with a <par> or <seq> group:

```
<par begin= "5s" end= "3.5min">
  ...
</par>
```

This begin value delays group playback until 5 seconds after the preceding group or clip finishes. The end time means all clips in the group stop playing after 3.5 minutes regardless of their states. If all clips reach their normal conclusion by 3 minutes and 20 seconds after the group starts, for example, the next group or clip starts after 10 seconds of blank time.

If you use an end time and a repeat attribute as described in "Repeating a Clip or Group" on page 84, the group repeats only after the end time elapses. You should not set an end time along with an endsync attribute in a <par> group, as this sets up conflicting end times.

Additional Information

For more on endsync, see "Ending a Parallel Group on a Specific Clip" on page 83.

Setting Internal Clip Begin and End Times

The clip-begin and clip-end attributes specify a clip's internal timing marks where playback begins and ends. You can use them with clips that have internal timelines, such as audio, video, and animation. Do not use them with groups or static clips such as still images. Here is an example:

```
<video src="videos/newsong.rm" clip-begin="10.5s" clip-end="50.7s"/>
```

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

Combining clip-begin and clip-end with begin and end

You can combine clip-begin and clip-end attributes with begin and end attributes. Here, a begin time has been added to the clip example shown above:

```
<video src="videos/newsong.rm" clip-begin="10.5s" clip-end="50.7s" begin="5s"/>
```

The begin time delays the clip's normal starting point by five seconds. When this time elapses, the clip starts at its 10.5 second internal timeline marker, then plays for 40.2 seconds. In this case, the clip-end attribute determines how long the video is active. But you could also add an end attribute as shown here to modify this behavior:

```
<video src="videos/newsong.rm" clip-begin="10.5s" clip-end="50.7s" begin="5s"
  end="50s"/>
```

Combined with begin, the end value of 50 means the clip's "window" within the presentation is 45 seconds. Because the clip stops playing after 40.2 seconds, there is an extra 4.8 seconds during which the clip does not play but remains active. How the video window appears during these final seconds depends on the fill attribute.

Additional Information

"Setting a Fill" on page 89.

Setting Durations

The dur attribute controls how long a clip or group appears after it starts to play back. It is useful with graphic images, as shown in this example:

```
<img src="graphics/poster.jpg" dur="14.5s"/>
```

This dur time makes the graphic disappears 14.5 seconds after it appears. You can also use dur in place of end. For example, in the following clip:

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

you can substitute a dur attribute for the end attribute to achieve the same result:

```
<video src="videos/newsong.rm" begin="20.5s" dur="42.2s"/>
```

In both examples, the clips stop playback 42.2 seconds after starting. With the end attribute, the total playing time is the end value minus the begin value. The dur attribute ignores the begin value, stopping the clip 42.2 seconds after it starts. Use either end or dur, therefore, depending on how you want to measure time. Do not use both attributes in the same tag, however.

Additional Information

The dur attribute can function like end in a group, too. For more information, see "Using Begin and End Times with Groups" on page 87.

Setting a Fill

The fill attribute determines what happens to the clip immediately after it plays to its normal end point, or its specified end time or duration elapses. The fill value can be remove or freeze.

remove

The default value fill="remove" removes the clip. When this attribute is used with a still image, the image disappears once the end time has elapsed.

freeze

Use fill="freeze" to freeze the clip on its last frame. When used with a video, the video's last frame stays on the screen. Suppose that you have a 20-second video and specify a 30-second duration with freeze:

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

After the video plays, its last frame displays for 10 seconds. The video disappears when the end time elapses. The fill="freeze" attribute has no effect on audio. Do not use fill="freeze" for a graphic image that also uses a dur attribute.

Clip Timing Example

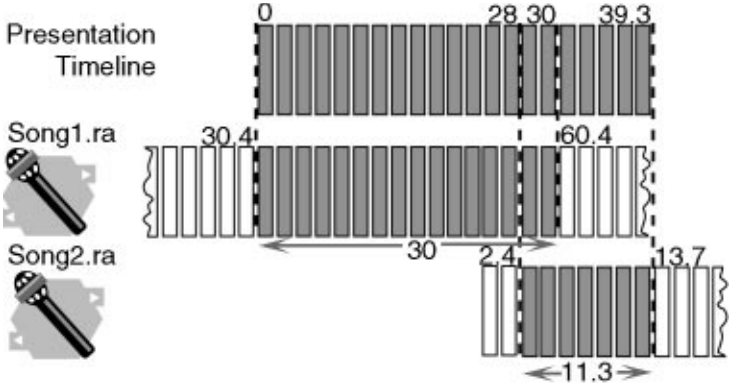
The following example shows two audio clips with different timing options:

```
<par>
  <audio src="song1.ra" clip-begin="30.4s" dur="30s"/>
  <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, playing 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. The following figure illustrates the relationships of the clip timelines to the overall presentation timeline.

Clip Timing Example



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 in this form:

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

RealPlayer looks at choices in order, evaluating each test attribute and its value to determine which clip to choose. The choices are typically clip source tags such as <video *test-attribute="value"*/>, but RealPlayer can also choose between groups when test attributes appear in <par> or <seq> tags.

Note

Every RealPlayer must have a viable choice within a <switch> statement. RealPlayer will not play any clip in the <switch> group if it finds no satisfactory test attribute. The following sections explain how to guarantee that every RealPlayer has a viable option.

Setting Language Choices

When the <switch> group test attribute is system-language, each source clip is for a different language. The following example shows a video slide show with separate audio narrations in French, German, Spanish, and English. Based on the language preference set through its **Options** menu, as well as the system-language code defined in the SMIL file, RealPlayer chooses a clip to play:

```
<par>
  <video src="slides/seattle.rm"/>
  <!-- select audio based on RealPlayer language preference setting -->
  <switch>
    <audio src="french/seattle.ra" system-language="fr"/>
    <audio src="german/seattle.ra" system-language="gr"/>
    <audio src="spanish/seattle.ra" system-language="es"/>
    <audio src="english/seattle.ra"/>
  </switch>
</par>
```

Because the last option does not have a test attribute, a RealPlayer that does not have French, German, or Spanish explicitly set as its preferred language chooses the English clip regardless of its actual language setting. RealPlayer evaluates options in order, so the last option should be, as shown in this example, a default language that applies if no other option is viable. If you list no default option and certain RealPlayers prefer languages other than the ones you list, those RealPlayers will not play any of the clips.

Additional Information

Appendix D starting on page 173 lists the system-language codes such as "en" you use to designate content in different languages.

Setting Bandwidth Choices

To serve different clips to viewers with different connection speeds, use the <switch> tag to define options each RealPlayer can choose from based on its available bandwidth. As shown below, you can group clips 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.

Additional Information

"Writing Complex SMIL Switch Statements" on page 166 explains how to use `<switch>` with SureStream clips, as well as how to test for both language and bandwidth.

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. If you want to keep the playback area stable in size as different clips play back, or if your presentation displays several clips at a time, you can define playback areas called "regions" within the main RealPlayer window:

1. In the SMIL file header, you create a `<layout>` group, using `<region>` tags to name playback regions and define their sizes and locations within the RealPlayer clip window. See "Defining the Layout" below.
2. In the SMIL file body, you use `region` attributes to specify which source clips play in which regions. See "Assigning Clips to Regions" on page 101.

Additional Information

See "Working with SMIL Layouts" on page 128 for instructions on using SMIL and RealPlayer's Netscape plug-in or ActiveX control to lay out the presentation in a Web page instead of in RealPlayer.

Defining the Layout

When you lay out regions, you create a root-layout region that defines the size of the RealPlayer main window. You then create other regions using a simple

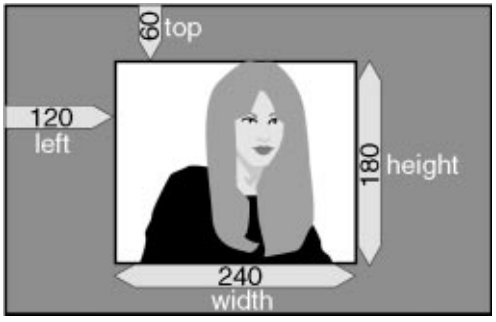
coordinate system measured across and down from the top, left-hand corner of the root-layout region. All measurements are in pixels or percentages, with zero pixels as the default. The following table lists the attributes that define region size and placement.

Region Size and Placement Attributes

Attribute	Sets	Pixel Example	Percentage Example
`top`	offset from top of window	`top="60"`	`top="10%"`
`left`	offset from left side of window	`left="120"`	`left="20%"`
`width`	region width	`width="240"`	`width="40%"`
`height`	region height	`height="180"`	`height="30%"`

The next figure illustrates the layout of a single playback region within a root-layout region.

Basic Region Layout



Setting the Root Layout Region

With the `root-layout` attribute, you specify the size of the entire playback area in pixels (percentages are not accepted for the root-layout region). You cannot display images or play clips in the root-layout region, as it is meant to set the overall playback area. The example shown below creates a root-layout region 250 pixels wide by 230 pixels high. 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:

```
<head>
  <layout>
    <root-layout width="250" height="230"/>
    ...other regions defined here...
  </layout>
</head>
```

Note

Although you can omit root-layout to have RealPlayer calculate the playback area based on the sizes of the other regions, it is better to define root-layout to avoid unexpected results.

Defining Playback Regions

You create playback regions for media clips 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 below defines two regions named "videoregion" and "textregion":

```
<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>
```

In this example, 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.

SMIL Sample Layout with Video and Text Windows

Tips for Defining Regions

Note the following about SMIL regions:

- All regions used in the presentation must be defined in the header. All regions are created at the beginning of the presentation, so regions containing background colors as described in "Adding a Background Color" on page 97 may obscure other regions.
- Using default region transparency and the `z-index` attribute, you can emulate dynamic creation and destruction of regions. For details, see "Hiding Regions with z-index" on page 164.
- To use a graphic image such as a GIF or a JPEG as a background, define a playback region the same size as the root-layout region. Display the image in this region and use the `z-index` attribute to make it display behind other regions.

Additional Information

See "Ordering Overlapping Regions with z-index" on page 99.

Using Percentage Values for Regions

For a region's height, width, and offset measurements, you can use percentages that reflect a fraction of the root-layout region's size. The following example uses percentages to define playback areas similar to 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 Using 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%.

Adding a Background Color

By default, the root-layout region is black. All other regions use transparency as their default. In the SMIL layout, you can specify another background color for any region:

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

For the color value, use any RGB hexadecimal value (#RRGGBB) supported by HTML, or one of the following predefined color names, listed here with their corresponding hexadecimal values:

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

Using Transparency

A region is transparent if you do not define its background color. You cannot explicitly specify transparency as a region color, however. Default transparency means that the region is not visible until a clip starts to play in it. This is not true transparency, though. If the media clip contains transparency too, you will not see through the clip and region to the root-layout region.

Fitting Clips to Regions

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"/>
```

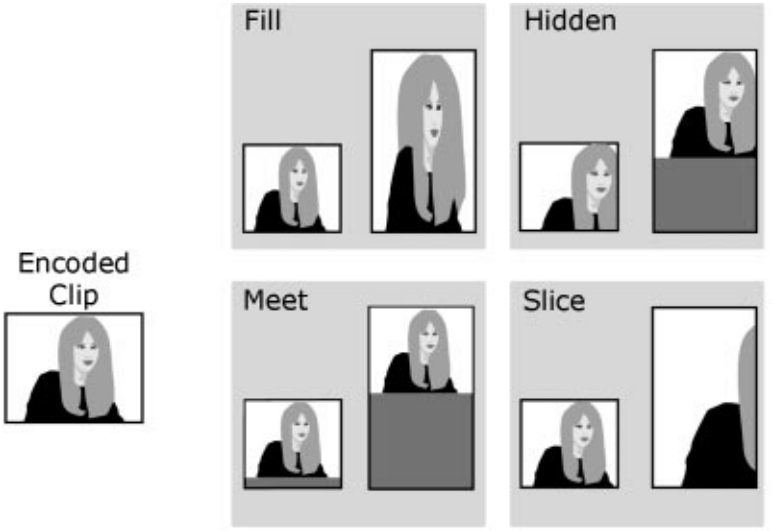
The fit attribute uses one of the values described in the following table. If you do not specify a fit attribute, the clip uses the default value `hidden`. In no case will media display outside the playback region's boundaries.

Region Fit Attributes

Attribute	Action
`fill`	Scale the clip so that it fills the region exactly. Image distortion occurs if the encoded clip and playback region have different height-to-width ratios.
`hidden` (default)	Keep the clip at its encoded size and place it 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.
`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.

The following figure illustrates the effect that fit attributes have on a source clip that plays in windows with different sizes and aspect ratios.

Different Clip Scaling Results Based on "fit" Attribute



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. Also, scaling a clip consumes CPU on the RealPlayer machine.

Ordering Overlapping Regions with z-index

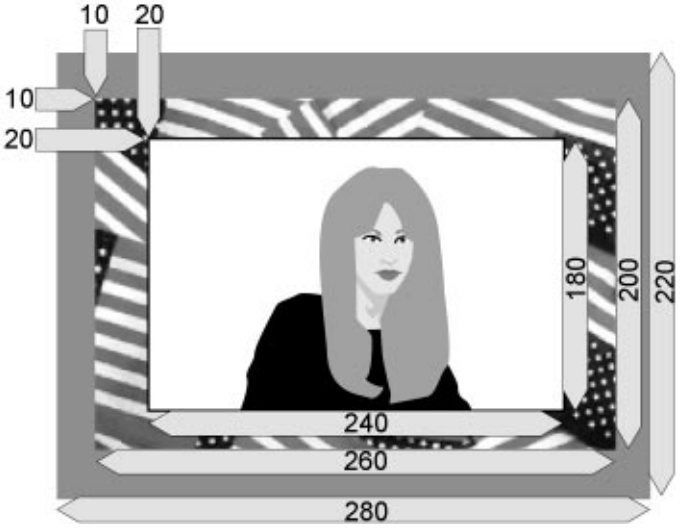
If regions overlap, you can use the `z-index` attribute to determine which regions appear in front. The following example creates a video region that overlaps an image region:

```
<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. The following figure illustrates these regions.

Regions Overlapping through z-index



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.

Additional Information

See "Hiding Regions with z-index" on page 164 for information on using `z-index` to make regions appear to come and go dynamically.

Assigning Clips to Regions

After you define the layout in the header section as described in"Defining the Layout" starting on page 93, you use `region` attributes within source tags to attach each source to a region. In the following example, the video and text clips are assigned to the video and text regions defined in the header:

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

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 clips to regions at all because audio does not require a display region.

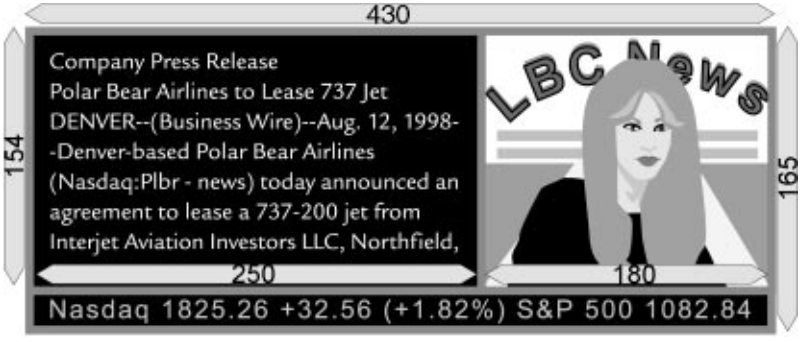
SMIL Layout Example

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 clips and 1 video clip-->
    <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.

SMIL Sample Layout with Text, Video, and Stock Ticker Windows



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

Making a Source Clip a Link

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

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

The example above links the source clip video.rm to the target clip video2.rm. When a viewer moves the cursor over the source clip as it plays, the cursor turns to a hand icon to indicate that the clip is a link. When the viewer clicks video.rm as it plays, video2.rm replaces it. The URL begins with rtsp:// if the linked clip 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 Clips on RealServer" on page 79.

Targeting RealPlayer or a Browser

An <a> tag or <anchor> tag (see "Defining Hot Spot Links" on page 104) can include a show attribute that determines where a linked clip displays:

```
<a href="http://www.company.com/index.htm" show="new">
  <video src="video.rm" region="videoregion"/>
</a>
```

replace

The default attribute `show="replace"` causes the linked clip to replace the source clip 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 clip 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 clip replaces all media clips 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 clips playing in those regions when the viewer clicks the link, however.

Additional Information

"Linking to a SMIL File" on page 107.

new, pause

The values `new` and `pause` both open the linked clip in the viewer's default browser. The source clip continues to play in RealPlayer if you use `show="new"`. With `show="pause"`, the source clip 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 clip 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 clip played in RealPlayer, however, such as a SMIL file or a RealVideo clip.

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 clip 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 clip'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 uses rtsp:// if the linked clip streams 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 Clips on RealServer" on page 79. To target a browser with a link, see "Targeting RealPlayer or a Browser" on page 103.

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 clip as shown in this example:

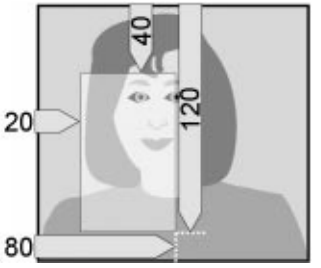
```
<video src="video.rm" region="videoregion">
  <anchor href="..." coords="20,40,80,120"/>
</video>
```

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.

Hot Spot (Hyperlink) Defined with SMIL



Percentage Values

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

```
<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 source clip. Each hot spot is a quarter the size of the source clip.

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 clip's upper, left-hand corner, respectively.

But the hot spot's right and bottom boundaries extend to the source clip's right and bottom edges, respectively, no matter the source clip'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 source clip is clipped at the clip's edge. For example, if the hot spot has coordinates "50,50,300,300" but the source clip 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 clip appears on screen. To add timing attributes, use the SMIL `begin` and `end` values. (You cannot use `dur`, `clip-begin`, or `clip-end`, however.)

The following example creates two temporal links for the clip `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 87. The attributes use the SMIL timing values described in "Specifying Timing" on page 86.

Linking to a SMIL File

A SMIL file can define a link to another SMIL file or another part of itself. 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 clip. 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 clip or <par> or <seq> group by defining an id attribute for the clip 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 clip in a <par> group and exclude the other clips in that group. All clips in the group will play in their designated regions.
- If additional clips follow the target clip in the SMIL file, those clips play when the target clip finishes playback.

> **Tip**
>
> If you want to link to a single clip but the SMIL file that refers to that clip lists other clips as well, link to the clip directly. Or create a new SMIL file that lists only the single target clip.

- 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 clip. 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 clip 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 86 describes the SMIL timing values.

Defining Image Options

For a still image, you can use the following options to modify the image's streaming characteristics, or to link it to RealPlayer commands. These options are not SMIL parameters, but extensions to the image's SMIL source tag,

which is described in "Specifying Clip Locations" on page 78. The following table summarizes the image options, which are describe in detail below.

Options for Image Source Tag

Option	Function	Format	Values	Example
`bitrate`	sets streaming bit rate	all	bits per second	`bitrate=1000`
`bgcolor`	substitutes color for transparency	GIF	color name or hex value	`bgcolor=black` `bgcolor=AA3344`
`url`	links to URL or a RealPlayer command	all	URL or command	`url=http://www.real.com` `url=command:stop()`
`target`	sets browser or RealPlayer as target	all	`_browser` `_player`	`target=_browser` `target=_player`

A question mark operator ("?") separates image options from the image URL in the SMIL source tag. To use multiple image options, put a question mark before the first option, then separate the remaining options with ampersands (&). The order of options does not matter. The following shows the general form for two image options added to the image source tag. Note that image options and values are not in quotation marks because they are part of the quoted src value:

```
<image src= "URL?option=value&option=value"/>
```

Here is an example that combines bit rate and background color options:

```
<image src="ad1.gif?bitrate=1000&bgcolor=blue"/>
```

Setting Image Bit Rate

By default, an image streams at 12 Kbps until RealPlayer receives it. You can set a higher bit rate to take advantage of bandwidth availability and stream an image quickly. Or you can set a lower bit rate when streaming images in parallel with another clip so that the other clip has enough bandwidth for uninterrupted playback. The following example shows three sequential image files set to stream at a low bit rate to ensure that a video playing in parallel does not stall:

```
<par>
  <video src="video.rm" region ="videoregion"/>
  <par>
    <seq>
      <image src="ad1.gif?bitrate=1000" begin="30s" dur="60s" region="adregion"/>
      <image src="ad2.gif?bitrate=1000" dur="120s" region="adregion"/>
```

```
    <image src="ad3.gif?bitrate=1000" dur="60s" region="adregion"/>
   </seq>
  </par>
</par>
```

In this example, each GIF image is set to stream at 1 Kbps. This ensures that each image consumes a small amount of bandwidth and does not interfere with video playback. The <par> tag just outside the <seq> tag makes RealServer balance the bit rates of the images as a single group rather than as a discrete sequence.

Additional Information

For more on using <par> to balance bandwidth, see "Smoothing Transitions between Clips" on page 163.

Keep in mind that the image size divided by the maximum bit rate equals the minimum time for streaming the image. Assuming images in this example are 30 Kilobits, each image takes about 30 seconds to stream. For this reason, the first image is set to display 30 seconds after the video begins to play. If no begin time were defined, the video would be delayed up to 30 seconds while the first image streams to RealPlayer. As well, each successive image must be able to stream during the preceding image's duration.

Overriding GIF Transparency

For GIF images that include transparency, you can use bgcolor to substitute a color for the transparency. Color values are any hexadecimal color value without the leading pound sign, "#", or a predefined color name as described in "Adding a Background Color" on page 97. If you do not set the color, the region's defined background color shows through transparent spots in the image. Here is an example:

```
<image src="ad1.gif?bgcolor=BB21AA"/>
```

Linking to a URL

The image url and target options let you link an image to a fully qualified URL so that when the viewer clicks the image, the URL opens in the viewer's browser or RealPlayer. The following example shows an image linked to a URL that opens in the viewer's default browser:

```
<image src="ad.gif?url=http://www.company.com&target=_browser"
 region="ads"/>
```

This next example shows an image linked to a URL that opens in RealPlayer:

```
<image src="go.gif?url=rtsp://realserver.company.com/media.smi&target=_player"
 region="next"/>
```

Keep in mind that opening a URL in RealPlayer replaces the current presentation with the new clip or SMIL file.

Note

The image hyperlink options duplicate the SMIL hyperlink features described in "Linking to Other Media", beginning on page 103. You can use either method depending on your preference.

Controlling RealPlayer

With `url=command:`*command* and `target=_player`, you can make the image a link that causes RealPlayer to stop, pause, play, or seek to a specific point in the presentation timeline. The following example shows three images set to appear in three separate SMIL regions. Each image issues a different RealPlayer command when clicked:

```
<par>
 <image src="play.gif?url=command:play()&target=_player" region="play"/>
 <image src="pause.gif?url=command:pause()&target=_player" region="pause"/>
 <image src="stop.gif?url=command:stop()&target=_player" region="stop"/>
</par>
```

An image can also issue a seek command that specifies a certain point in the presentation timeline:

```
<image src="seek.gif?url=command:seek(1:35.4)&target=_player"
 region="seek"/>
```

In the example above, clicking the image instructs RealPlayer to seek to 1:35.4 into the presentation. Because this option is not a SMIL parameter, it does not use SMIL timing values such as `s` for seconds or `min` for minutes. Instead, the time format is the following:

```
dd:hh:mm:ss.xyz
```

Here, `dd` is days, `hh` is hours, `mm` is minutes, `ss` is seconds, `x` is tenths of seconds, `y` is hundredths of seconds, and `z` is milliseconds. Only the `ss` field is required. When the time value does not include a decimal point, the last field is read as the seconds.

For example, `1:30` means 1 minute and 30 seconds, whereas `1:30:00` means 1 hour and 30 minutes. Thus all the following values seek to the point in the timeline 90 minutes after the presentation begins:

```
url=command:seek(1:30:00.0)
url=command:seek(90:00)
url=command:seek(5400)
```

CHAPTER 8: Playing a Presentation in a Web Page

Embedding a presentation in your Web page provides seamless playback without launching RealPlayer as a separate application. You can even include RealPlayer controls such as fast forward and pause. This chapter explains how you embed RealPlayer's Netscape plug-in or ActiveX Control in your Web page so that people view your presentation directly through their Web browsers.

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:

- Microsoft Internet Explorer 3.0 and 4.0
- Microsoft Visual Basic applications
- Microsoft Access

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, the tag syntax for the plug-in and the control differs, however. 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 can be used only when retrieving content from a RealServer that uses the RAMGEN feature. It causes RealPlayer to play the presentation through its Netscape plug-in instead of launching as a separate application. If your RealServer does not use RAMGEN or you are hosting content on a Web server, create a RAM file with a .rpm extension manually and link to it instead of a SMIL file.

> **Additional Information**
>
> For more on RAMGEN and presentation URLs, see "Linking your Web Page to RealServer" on page 141. For instructions on writing a .rpm file, see "Creating a RAM File Manually" on page 147.

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">

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.

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=26 HEIGHT=26 CONTROLS=PlayButton>
```

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

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=26 HEIGHT=26>
<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 suggested 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 123.

Full Controls

All

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

Suggested pixel width: 400
Suggested pixel height: 100

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** and **Stop**.

Suggested pixel width: 176 or higher
Suggested pixel height: 132 or higher

Additional Information

"Controlling Image Display" on page 125.

Individual Controls and Sliders

ControlPanel

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

Suggested pixel width: 350

Suggested pixel height: 36

PlayButton

Displays the Play/Pause button.

Suggested pixel width: 44

Suggested pixel height: 26

PlayOnlyButton

Displays the Play button without Pause.

Suggested pixel width: 26

Suggested pixel height: 26

PauseButton

Displays the Pause button without Play.

Suggested pixel width: 26

Suggested pixel height: 26

StopButton

Displays the Stop button.

Suggested pixel width: 26

Suggested pixel height: 26

FFCtrl

Displays the Fast Forward control.

Suggested pixel width: 26

Suggested pixel height: 26

RWCtrl

Displays the Rewind control.

Suggested pixel width: 26

Suggested pixel height: 26

MuteCtrl

Displays the Mute control button.

Suggested pixel width: 26

Suggested pixel height: 26

MuteVolume

Displays the Mute control and volume slider.

Suggested pixel width: 26

Suggested pixel height: 88

VolumeSlider

Displays the volume slider.

Suggested pixel width: 26

Suggested pixel height: 65

PositionSlider

Displays the clip position slider.

Suggested pixel width: 240

Suggested pixel height: 22

Information Panels

InfoVolumePanel

Displays the Title, Author, and Copyright information panel, as well as the volume slider. See also "NOLABELS" on page 127.

Suggested pixel width: 300

Suggested pixel height: 88

InfoPanel

Displays the Title, Author, and Copyright information panel. See also "NOLABELS" on page 127.

Suggested pixel width: 300

Suggested pixel height: 55

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.

Suggested pixel width: 300

Suggested pixel height: 30

PositionField

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

Suggested pixel width: 90

Suggested pixel height: 30

StatusField

Displays the message text area of the status bar.

Suggested pixel width: 300

Suggested pixel height: 30

StatusBar

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

Suggested pixel width: 300

Suggested pixel height: 30

Using Multiple Controls

The `CONSOLE` parameter defines a name that unifies `<EMBED>` or `<OBJECT>` tags so that multiple 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 tags, you can set the size of each control separately and define its layout with HTML tags. You could put each control in a different HTML table cell, for example.

To tie controls together, define the same `CONSOLE` name within each `<EMBED>` or `<OBJECT>` tag, or use 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 showing two video clips, for example, you can define console names of `video1` and `video2`. All controls linked by `video1` interoperate and all controls linked by `video2` interoperate. But a volume slider for `video1`, for example, will not affect the volume of the `video2` clip.

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. The predefined console name _master links the image window 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.

Netscape Plug-in Sample Mark-up

Because each <EMBED> tag must have a SRC value, the image window in the following example simply uses the same source as the first Play button. 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.

```
<EMBED SRC="http://realserver.company.com:8080/ramgen/sample1.rm?embed"
WIDTH=176 HEIGHT=128 CONTROLS=ImageWindow CONSOLE=_master>

<H4>Video 1</H4>
<EMBED SRC="http://realserver.company.com:8080/ramgen/sample1.rm?embed"
WIDTH=44 HEIGHT=26 CONTROLS=PlayButton CONSOLE=video1>
<EMBED SRC="http://realserver.company.com:8080/ramgen/sample1.rm?embed"
WIDTH=26 HEIGHT=26 CONTROLS=StopButton CONSOLE=video1>

<H4>Video 2</H4>
<EMBED SRC="http://realserver.company.com:8080/ramgen/sample2.rm?embed"
WIDTH=44 HEIGHT=26 CONTROLS=PlayButton CONSOLE=video2>
<EMBED SRC="http://realserver.company.com:8080/ramgen/sample2.rm?embed"
WIDTH=26 HEIGHT=26 CONTROLS=StopButton CONSOLE=video2>
```

ActiveX Control Sample Mark-up

In the following ActiveX example, only the controls for the two play buttons define the source URLs. For convenience the CLASSID value is omitted.

```
<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:8080/ramgen/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:8080/ramgen/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>
```

Controlling Image Display

The following <EMBED> and <OBJECT> parameters control aspects of how clips play back.

BACKGROUNDCOLOR

This parameter specifies a background color that appears in the image window if no clip is playing. The specified background color also shows through if a clip includes transparency. The background color is black by

default. You can use an RGB hexadecimal color value (#RRGGBB) or the following color names, shown here with their corresponding RGB values:

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

Note

SMIL region background colors override this background color. For more on setting SMIL region colors, see "Adding a Background Color" on page 97.

CENTER

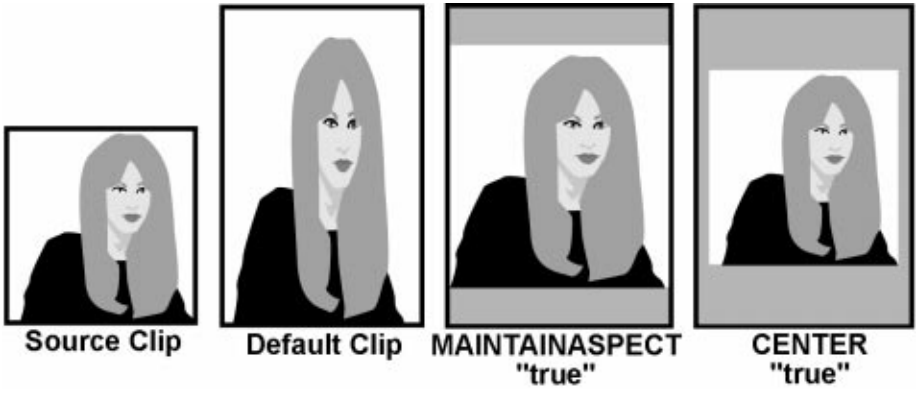
The default value for CENTER is false, which causes the clip to fill the entire playback area. If you set CENTER to true, the clip is centered within the playback area and displays at its encoded size. So by setting CENTER to true, you can create a large playback area with WIDTH and HEIGHT and still have the clip play at its normal size. You cannot use CENTER along with MAINTAINASPECT.

MAINTAINASPECT

This parameter, which you cannot combine with CENTER, determines whether the clip's height-to-width ratio stays constant when the clip scales to fit the image window. The default value of false causes this ratio to change as necessary to fill the image window fully. This may distort the source image.

If you set MAINTAINASPECT to true, a clip's height-to-width ratio stays constant. For example, a clip's height-to-width ratio of 1:1 stays constant even if the image window's height-to-width ratio is 2:3. In these cases, the clip is centered in the image window and scaled until one dimension reaches the window's boundaries and the other dimension is within the boundaries. The following figure shows how clips scale by default, with MAINTAINASPECT set to true, and with CENTER set to true.

Clip Scaling with MAINTAINASPECT and CENTER



NOLABELS

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>
```

NOLOGO

When set to true, NOLOGO prevents the RealLogo from displaying in the image window. When no clip is playing, only the specified background color shows in the window. The parameter is false by default.

Setting Automatic Playback

The AUTOSTART, LOOP, and NUMLOOP parameters let you set the content to start playing automatically, as well as loop continuously or a specified number of times. The following example shows two of these parameters used with the Netscape plug-in:

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

Here is 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 playback immediately. When you have multiple <EMBED> or <OBJECT> tags linked by a CONSOLE name, set AUTOSTART to true in just one tag. Leaving AUTOSTART out or setting its value to false means the presentation does not start until the user starts it by 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, set looping in just one tag. If you leave the LOOP parameter out, the default value of false applies and the presentation stops after the first playback. The user can play the presentation again by clicking the Play button.

> **Note**
>
> LOOP and NUMLOOP are not functional in the Beta 2 version of RealPlayer G2.

NUMLOOP

If you specify a NUMLOOP value such "2", the presentation loops the specified number of times and then stops. If both LOOP and NUMLOOP are used, the LOOP parameter is ignored.

Working with SMIL Layouts

As "Laying Out Multiple Clips" on page 93 explains, you can use a SMIL file to define separate playback regions for different parts of a presentation. This lets you lay out two videos side-by-side, for example. When playing a presentation in a Web page, you can define the layout in SMIL, or define it 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 102 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 clip with a SMIL region. For example, you would modify the sample layout shown in "SMIL Layout Example" on page 102 to look like the following:

```
<smil>
 <head>
  <!--presentation with 2 text clips and 1 video clip-->
  <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 clip 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://realserver.company.com:8080/ramgen/sample.smi?embed"
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://realserver.company.com:8080/ramgen/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 clip. 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, and 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.DoPlay()">
```

Note

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

Parameter Reference

<EMBED> and <OBJECT> Tag Parameters

Parameter	Function	Works with <EMBED>	Works with <OBJECT>	Values	Default	Refer To
AUTOGOTOURL	Sends URLS to applet or VisualBasic application when set to false.	**yes**	**yes**	true|false	true	page 130
AUTOSTART	Sets autoplayback.	**yes**	**yes**	true|false	false	page 128
BACKGROUNDCOLOR	Sets background color.	**yes**	**yes**	color name or RGB hex value	black	page 125
CENTER	Centers clip in window.	**yes**	**yes**	true|false	false	page 126
CONSOLE	Links multiple controls.	**yes**	**yes**	Console name or _master	(none)	page 123
CONTROLS	Adds RealPlayer controls.	**yes**	**yes**	Control name	All	page 118
HEIGHT	Sets window height.	**yes**	**yes**	percentage or pixel value	(none)	page 118
LOOP	Loops clips indefinitely.	**yes**	**yes**	true|false	false	page 128
MAINTAINASPECT	Keeps image aspect ratio.	**yes**	**yes**	true|false	false	page 126
NAME	JavaScript reference.	**yes**	no	Any name	(none)	page 131
NOLABELS	Suppresses title, author, and copyright fields.	**yes**	**yes**	true|false	false	page 127
NOLOGO	Suppresses RealLogo.	**yes**	**yes**	true|false	false	page 127
NUMLOOP	Loops clip a given number of times.	**yes**	**yes**	Any number	(none)	page 128
REGION	Ties clip to SMIL region.	**yes**	**yes**	SMIL region	(none)	page 129
SRC	Specifies source clip.	**yes**	**yes**	URL	(none)	page 117
WIDTH	Sets window width.	**yes**	**yes**	percentage or pixel value	(none)	page 118

CHAPTER 9: Inserting Ads in a Presentation

SMIL makes it easy to include advertisements in your streaming presentation. You can lay out ad banners, supply ad URLs, and even specify the times when ads appear. This chapter provides examples of how you can use SMIL to include ads in your presentations. Before you start, you should have a thorough understanding of SMIL as described in Chapter 7 beginning on page 75.

Placing Ads in Presentations

With RealSystem G2, you can create ads in any format, whether graphics, video, audio, or animation. Using SMIL, you simply treat your ads the same way you treat your media clips. You lay out your presentation and time when each media and ad clip appears. Because a SMIL file lists separate URLs to each clip, your ads can be on any server, whether RealServer, a Web server, or an ad server. And because a SMIL file is a plain text file, you can dynamically generate it by any means to present different ads for each page visitor.

Using an Ad Banner

The following example creates a video region with an ad banner above it. When a user plays the presentation, the video plays uninterrupted while ads appear in the ad banner twice a minute. The video and ads are defined to play at the same time within a <par> group. Within that group, ads are nested within a <seq> group and each defined to stay onscreen for 30 seconds. Each ad links to a different HTML page that displays in the browser when the user clicks the ad graphic.

```
<smil>
  <head>
    <!--presentation with video clip and ad banner-->
    <layout>
      <root-layout width="330" height="330" background-color="black"/>
```

```
    <region id="adbanner" top="5" left="5" width="320" height="72"/>
    <region id="videoregion" top="85" left="5" width="320" height="240"/>
   </layout>
  </head>
  <body>
   <par>
    <video src="newsvid.rm" region="videoregion"/>
    <par>
     <seq>
      <!--Display new linked ad banner every 30 seconds-->
      <a href="http://www.company.com/ads/offer1.htm" show="new">
       <img src="ad1.gif" region="adbanner" dur="30s"/></a>
      <a href="http://www.company.com/ads/offer2.htm" show="new">
       <img src="ad2.gif?bitrate=1000" region="adbanner" dur="30s"/></a>
      <a href="http://www.company.com/ads/offer3.htm" show="new">
       <img src="ad3.gif?bitrate=1000" region="adbanner" dur="30s"/></a>
      <a href="http://www.company.com/ads/offer4.htm" show="new">
       <img src="ad4.gif?bitrate=1000" region="adbanner" dur="30s"/></a>
     </seq>
    </par>
   </par>
  </body>
</smil>
```

Ad Banner Layout

Additional Information

"Laying Out Multiple Clips" on page 93 explains how to set up regions. SMIL hyperlinking is described in "Linking to Other Media" on page 103. For more on the image bitrate options, see "Defining Image Options" on page 109.

Inserting Ads in a Clip

The following example inserts three ad videos into a long video called news.rm. The effect is like a television program with commercial breaks. Each ad video links to an HTML page that displays in the browser when the user clicks the ad:

```
<smil>
 <head>
  <!--video clip with commercial breaks-->
  <meta name="author" content="Jane Morales"/>
  <meta name="title" content="Technology in the News"/>
  <meta name="copyright" content="(c)1998 Jane Morales"/>
 </head>
 <body>
  <seq>
   <video src="news.rm" clip-end="8.5min"/>
    <a href="http://www.company.com/ads/offer1.htm" show="new">
   <video src="commercial1.rm"/></a>
   <video src="news.rm" clip-begin="8.5min" clip-end="16.75min"/>
    <a href="http://www.company.com/ads/offer2.htm" show="new">
   <video src="commercial2.rm"/></a>
   <video src="news.rm" clip-begin="16.75min" clip-end="24.5min"/>
    <a href="http://www.company.com/ads/offer3.htm" show="new">
   <video src="commercial3.rm"/></a>
   <video src="news.rm" clip-begin="24.5min"/>
  </seq>
 </body>
</smil>
```

As shown above, the three ad clips and the main video clip, news.rm, are in a <seq> group. The news.rm video begins at its normal beginning and plays for 8.5 minutes. It then stops while the commercial1.rm ad plays. The news.rm clip then resumes where it left off, breaking at later intervals for commercial2.rm and commercial3.rm. Because no timing elements are specified for the ads, they play from their normal beginnings to normal ends and may be any length.

Additional Information

For more on clip timing, see "Setting Internal Clip Begin and End Times" on page 88. SMIL hyperlinking is described in "Linking to Other Media" on page 103.

Tips on Inserting Ads in Clips

The following tips will help you create smoother transitions between media and ad clips:

- If you do not define SMIL regions, encode all media clips at the same size so the window does not resize with each new clip. When using media at different sizes, define a region that all clips play in. This keeps the playback area stable.

  Additional Information

  "Laying Out Multiple Clips" on page 93 explains how to set up regions.

- If you experience unacceptable latency between clips, such as a long buffering time between a media clip and an ad clip, you can insert a small graphic between the clips to mask preroll. Or follow the instructions in "Smoothing Transitions between Clips" on page 163 to reduce rebuffering between clips.

Using RealPix for Ads

Instead of streaming ads to RealPlayer directly through SMIL, you can use RealPix to assemble JPEG and GIF (including animated GIF) images into ad presentations. Using RealPix gives you these advantages over streaming images directly through SMIL:

- Create transitions between ad clips.

  RealPix lets you create exciting transitions, such as fades, wipes, zooms, and pans. You can create these effects through the RealPix mark-up with regular JPEG or GIF images. You do not need to use animated GIFs.

- Cache images in RealPlayer memory.

  RealPlayer G2 does not maintain a disk cache, but caches images in memory for the duration of the RealPix presentation. A RealPix presentation that reuses an ad graphic does not use bandwidth after RealPlayer has received the graphic. Images streamed directly through

SMIL are not cached, though, and RealServer must stream each image each time it appears in the presentation.

As described in *RealPix Authoring Guide*, the RealPix file contains mark-up that specifies the locations of image files and creates the desired transitions. Within the SMIL file, you simply link to the RealPix file and assign it to appear in the appropriate region. The following example is the same as that used in "Using an Ad Banner" on page 133, except the individual GIF ads have been replaced with a RealPix presentation:

```
<smil>
  <head>
    <!--presentation with video clip and ad banner-->
    <layout>
      <root-layout width="330" height="330" background-color="black"/>
      <region id="adbanner" top="5" left="5" width="320" height="72"/>
      <region id="videoregion" top="85" left="5" width="320" height="240"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="newsvid.rm" region="videoregion"/>
      <ref src="adrotation.rp" region="adbanner"/>
    </par>
  </body>
</smil>
```

Additional Information

For more on RealPix, get *RealPix Authoring Guide* from **http://service.real.com/help/library/index.html**.

CHAPTER 10: Delivering a Presentation

When you finish building your RealSystem presentation, you place the clips on RealServer for streaming. This chapter explains how to link your Web page to your presentation. It also describes how a Web server can play back some RealSystem G2 presentations.

Using RTSP and HTTP in URLs

A RealSystem presentation served by RealServer G2 involves the HTTP and RTSP protocols. If you have experience creating Web pages, you know that HTTP is the standard protocol Web browsers and Web servers use to communicate. Most Web page URLs begin with `http://` to indicate that the file download occurs over the HTTP protocol.

As explained in "Hosting a Presentation on a Server" on page 21, RealServer G2 and RealPlayer G2 communicate primarily with RTSP, a new protocol designed specifically for streaming media. When you assemble a RealSystem presentation, it's important to understand clearly which URLs should begin with `http://` and which should begin with `rtsp://`.

Use `rtsp://` in URLs in which RealPlayer G2 requests clips from RealServer G2. These URLs occur in SMIL files (`.smi`) and RAM files (`.ram` or `.rpm`). Use `http://` in these URLs only if the clips are stored on a Web server instead of RealServer. Because a Web server doesn't use RTSP, you can't have `rtsp://` in a URL to a clip stored on a Web server.

Links to a Web server or RealServer within a Web page (`.htm`) always start with `http://`. Like Web servers, Web browsers don't use RTSP and can't interpret streaming information sent by RealPlayer over RTSP. The Web browser can connect to RealServer through HTTP, though, because RealServer G2 uses HTTP as well as RTSP.

Additional Information

For more on SMIL file URLs, see "Linking to Clips on RealServer" on page 79 and "Linking to Clips on a Web Server" on page 81.

Streaming Clips from RealServer G2

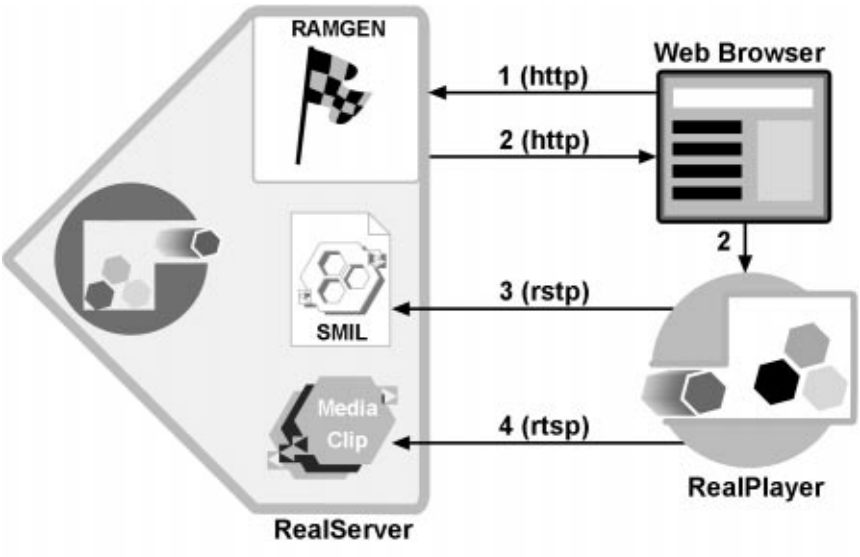
When you stream clips from RealServer, the RealServer administrator creates content directories and tells you the basic URLs to use. The administrator can also set up features such as password authentication and pay-per-view. When your media clips and SMIL file are ready, transfer them to RealServer and place them in the directories prepared by the administrator. Then link your Web page to your presentation as described below.

Tip

Because earlier versions of RealServer do not support SMIL and RTSP, ensure that RealServer G2 hosts your presentation. Also note that RealNetworks encoding tools can transfer files to RealServer automatically. See **http://www.real.com/products/tools/** for details.

When you use RealServer, the RAMGEN feature can automatically launch RealPlayer, eliminating the need to write a separate RAM file. Your Web page URL simply points to your media clip or SMIL file on RealServer and includes a `ramgen` parameter. The next figure illustrates the process of requesting a presentation through RAMGEN. The example uses a SMIL file that coordinates multiple clips, but you can also link to a single clip directly without using SMIL.

Requesting a Presentation from RealServer using RAMGEN



1. Using the HTTP protocol, the Web browser requests the SMIL file from RealServer. The URL includes a parameter that invokes RAMGEN.
2. RealServer's response causes the Web browser to launch RealPlayer as a helper application and give it the requested URL to the SMIL file.
3. RealPlayer requests the SMIL file from RealServer using the RTSP protocol.
4. With the information in the SMIL file, RealPlayer requests and receives the streaming media clips.

Linking your Web Page to RealServer

With your clips 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 through RealPlayer's Netscape plug-in, the URL occurs within an `<EMBED>` or `<OBJECT>` tag and looks like this:

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

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

URL Components in a Web Page Link to RealServer G2

URL Component	Meaning
`http://`	This makes the browser contact RealServer through the HTTP protocol. (Web browsers do not use RTSP.)
`realserver.company.com`	This address varies for each RealServer. It typically uses an identifier such as `realserver` instead of `www`. It may also use a numeric TCP/IP address such as `204.71.154.5`.
`:8080`	This is the port RealServer normally uses for HTTP connections. Separate the port and address with a colon. You can leave the port number out only if RealServer uses port `80` for HTTP connections. Include the port number if RealServer uses any port besides 80 for HTTP.
`/ramgen/`	As "Using RAMGEN" explains, this parameter launches RealPlayer without the use of a separate RAM file.
`sample.smi`	This is the SMIL file for your presentation. If you have just one clip to stream, you can link directly to that clip instead of a SMIL file.
`?altplay=`*`file.ext`*	This RAMGEN option specifies an alternate presentation created for older versions of RealPlayer. See "Listing Alternate Presentations" on page 143.
`?embed`	This RAMGEN option embeds the presentation in a Web page. See Chapter 8 starting on page 115 for full information on Web page playback.

Using RAMGEN

In your Web page hyperlink, the `/ramgen/` parameter causes the Web browser to launch RealPlayer without the use of a separate RAM file. This 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://realserver.company.com:8080/ramgen/media/sample.smi">..</a>
```

You should use RAMGEN even when linking to a single clip, such as a RealVideo (`.rm`) clip, that automatically launches RealPlayer. If you cannot use RAMGEN, you can write the RAM file as described in "Creating a RAM File Manually" on page 147.

Listing Alternate Presentations

When you update content for RealSystem G2, you can keep existing content available for earlier versions of RealPlayer. Suppose you have a RealVideo 5.0 clip that you want to encode with a RealSystem G2 codec and lay out with SMIL. After re-encoding the source file with the new codec and writing the SMIL file, you change the link to the 5.0 RAM file to point to the SMIL file, using the RAMGEN `altplay` option to list the older clip as an alternate:

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

This link instructs RealServer to point RealPlayer G2 to `sample.smi`. Earlier versions of RealPlayer receive the URL to the older `old_sample.rm` file. Note that the URL specifies the actual clip, not the old RAM file.

> **Tip**
>
> It is not necessary to keep older content available. If you do not use `altplay`, page visitors using 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 (&). The order of options does not matter. For example, the following link uses `altplay` and `embed`:

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

You can use the question mark operator to include earlier RAM file options when using `altplay`. If your RAM file URL for a RealVideo 5.0 presentation specified an end time, for example, include that option in the RAMGEN URL after `altplay`. The following shows an end time set for `old_sample.rm`:

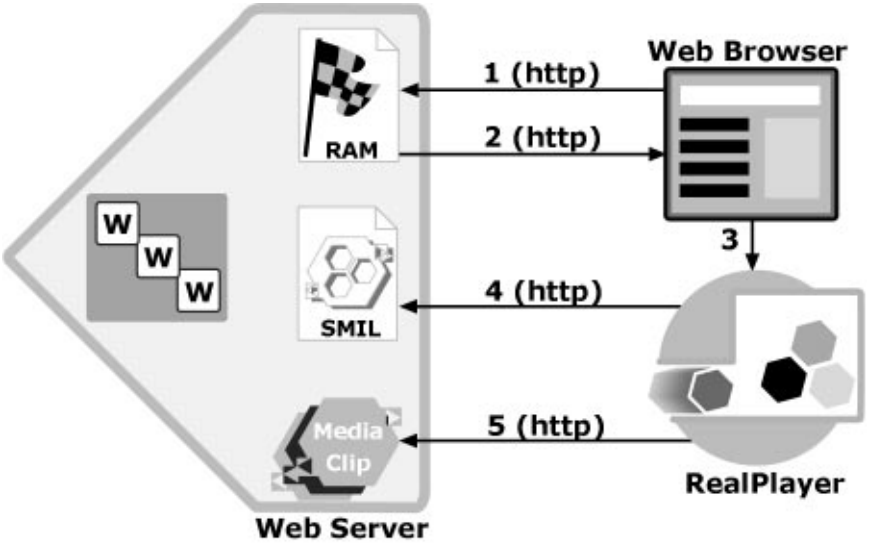
```
<a href="http://.../ramgen/sample.smi?altplay=old_sample.rm&end=7:45">
```

Playing Clips from a Web Server

If you do not have access to RealServer G2, you can host your presentation on a Web server. Although not as robust as RealServer streaming, Web server playback provides a reasonable method for sending simple presentations to a small number of users. It is not recommended for lengthy or complicated presentations, however, or for clips viewed simultaneously by large groups.

As described in "Launching a Presentation" on page 22, a RAM file launches RealPlayer when the presentation plays back from a Web server. You therefore need to write the RAM file as described in "Creating a RAM File Manually" on page 147 and place it on the Web server. The following figure illustrates the process of requesting a presentation from a Web server. All network actions use the HTTP protocol.

Requesting a Presentation from a Web Server



1. The Web browser requests the RAM file from the Web server.
2. The Web server downloads the RAM file to the browser.
3. The RAM file extension (`.ram` or `.rpm`) causes the Web browser to launch RealPlayer. A `.ram` extension launches RealPlayer as a separate application. A `.rpm` extension launches RealPlayer's Netscape plug-in to play the presentation embedded within the Web page.
4. RealPlayer receives the RAM file and requests the SMIL file from the Web server. Alternately, the RAM file can simply list a single clip or multiple clips played in sequence.
5. With the information in the SMIL file, RealPlayer requests and receives the clips from the Web server.

Limitations on Web Server Playback

As noted in "Hosting a Presentation on a Server" on page 21, a presentation downloaded from a Web server is more likely to stall than a presentation streamed by RealServer. To ensure that a presentation hosted by a Web server plays as smoothly as possible, observe the following points:

- No Clips Encoded for Multiple Bandwidths

  A Web server cannot stream just one encoding from a clip encoded for multiple bandwidths. Instead, it downloads the entire clip file, causing an unacceptably high preroll. So you must encode each clip for just one bandwidth. To support multiple bandwidths, encode separate clips for various bandwidths and use SMIL to let RealPlayer choose which clip to play.

  Additional Information

  See "Choosing RealAudio Codecs" on page 45 for information on codecs and multiply encoded clips. For more on using SMIL to list clip choices, read "Setting Bandwidth Choices" on page 92.

- Limited Ability to Keep Parallel Clips Synchronized

  Because a Web server does not consider clip timelines when downloading data and does not receive feedback from RealPlayer, Web server playback makes it harder for RealPlayer to keep clips synchronized. A presentation that plays clips in parallel may begin to stall and rebuffer data when the RealPlayer connection has little bandwidth to spare.

  For this reason, it may be difficult to deliver RealFlash, which consists of synchronized animation and audio. You can, however, combine a large clip such as RealVideo with smaller clips, such as image clips or a RealText clip. Because RealText consumes little bandwidth, a Web server can download it to RealPlayer quickly without interfering with other clips.

- RealPix Presentations Buffer Longer

  When delivered by RealServer, images in a RealPix presentation stream at different times depending on their place in the RealPix timeline. This lets you structure a RealPix presentation to keep it flowing smoothly. When delivered by a Web server, however, all RealPix images begin to download as soon as presentation playback begins. This causes a higher preroll.

Additional Information

RealPix Authoring Guide, available at **http://service.real.com/help/library/index.html**, explains bandwidth usage when RealServer G2 streams RealPix.

- SMIL File Optional

  When delivering a single clip or a few clips played in sequence, you do not need a SMIL file. You can simply list the clips in order when writing your RAM file as described in "Creating a RAM File Manually" on page 147. You can, however, have your RAM file specify a SMIL file that lists the clip locations, creates a layout, times the presentation, and so on.

- SMIL Internal Timing Commands not Usable

  Although you can use SMIL to lay out and time your presentation, you should not use the `clip-begin` and `clip-end` attributes. A Web server cannot begin to download a clip at a certain point in its timeline. If you specify `clip-begin="5min"`, for example, RealPlayer must wait until it has received the first five minutes of clip data before it can begin to play the clip back. This creates an unacceptably long wait.

  Additional Information

  "Setting Internal Clip Begin and End Times" on page 88 describes these SMIL commands.

  Because a Web server cannot jump to a new position in a clip timeline, the RealPlayer position slider cannot fast-forward the clip. If the user moves the slider forward, playback pauses as the clip continues to download at its normal rate. RealPlayer resumes playback once the clip data reaches the requested timeline position.

- No RTSP URLs

  Because Web servers do not support the RTSP protocol, all URLs in presentations hosted by a Web server should use HTTP and begin with `http://`. This includes all URLs in a SMIL file or RAM file.

- No Live Broadcast

  Live broadcast is not possible because Web servers can download only clips stored on disk.

Configuring Web Server MIME Types

To download a RealSystem presentation from a Web server, the server must be configured with the following MIME types. If you are using an ISP, ask the ISP's Web server administrator to configure the MIME types listed in the following table.

Web Server MIME Types for RealSystem Files

File Type	Extension	MIME Type
RAM	.ram	audio/x-pn-realaudio
Embedded RAM	.rpm	audio/x-pn-realaudio-plugin
SMIL	.smi	application/smil
RealMedia	.rm	application/vnd.rn-realmedia
RealAudio	.ra	audio/vnd.rn-realaudio
RealVideo	.rv	video/vnd.rn-realvideo
RealPix	.rp	image/vnd.rn-realpix
RealText	.rt	text/vnd.rn-realtext

Creating a RAM File Manually

Whenever possible, launch RealPlayer automatically with RealServer's RAMGEN feature as described in "Using RAMGEN" on page 142. In some cases, though, you may need to write a RAM file:

- Streaming from a RealServer not set up to use RAMGEN.
- Streaming multiple SMIL files in sequence.
- Hosting a RealSystem presentation from a Web server.
- Playing back clips that reside on the user's local machine.

➤ **To create a RAM file:**

1. Open any editor or word processor that can save files as plain text. On the top line, enter the full URL of the SMIL file or the media clip. As shown below, URLs vary with the playback context.

   - RealServer Streaming

     The following example links to a SMIL file on a RealServer machine that does not use RAMGEN:

     ```
     rtsp://realserver.company.com/media/sample1.smi
     ```

To deliver a few clips or SMIL files in sequence, list the URLs in their playback order:

```
rtsp://realserver.company.com/media/sample1.smi
rtsp://realserver.company.com/media/video1.rm
rtsp://realserver.company.com/media/sample2.smi
```

- Web Server Playback

  For Web server playback, you specify the HTTP protocol and the Web server name, along with the SMIL file or media clip:

  ```
  http://www.company.com/media/video1.rm
  ```

  To deliver a few clips or SMIL files in sequence, list the URLs in their playback order:

  ```
  http://www.company.com/media/video1.rm
  http://www.company.com/media/video2.rm
  http://www.company.com/media/sample1.smi
  ```

- Local Playback

  For local playback of clips residing on the user's machine, start the URL with `file://` and list clips in their locations relative to the RAM file. For example, the following specifies a clip that resides one level below the RAM file in the `media` directory:

  ```
  file://media/video1.rm
  ```

  To deliver a few clips or SMIL files in sequence, list the URLs in their playback order:

  ```
  file://media/video1.rm
  file://media/video2.rm
  file://media/sample1.smi
  ```

  Additional Information

  These local URLs are like those used in a SMIL file to locate media clips. For more information on general URL syntax, see "Specifying Clip Locations" on page 78.

2. For a presentation played back from RealServer, you can support earlier versions of RealPlayer (such as RealPlayer 4.0 or 5.0) just as RAMGEN does with the `altplay` option. To do this, add the marker `--stop--` after the RTSP URL. Then specify the URL for the older clip just as it appeared in your previous RAM file. Here's an example:

```
rtsp://realserver.company.com:554/sample.smi
--stop--
pnm://realserver.company.com:7070/old_sample.rm
```

The second URL specifies the older RealSystem protocol with `pnm://` and designates RealServer's PNA port. When RealPlayer connects, it chooses the URL based on its favored protocol. For this reason, you cannot list two URLS that both use the same protocol, whether `rtsp://`, `pnm://`, or `http://`.

3. Save the RAM file as plain text with a `.ram` extension (played in RealPlayer) or a `.rpm` extension (played in the Web browser).
4. Move your RAM file to RealServer or your Web server. Keep in mind that even if all your media clips are on RealServer, you can place the RAM file on your Web server. When the browser receives the RAM file from the Web server, it turns the file over to RealPlayer, which uses the full URLs in the RAM file to request the clips off RealServer. Hence the RAM file and the media clips do not need to reside on the same machine.
5. For `.ram` files, link your Web page to the RAM file with a standard HTML hyperlink. For `.rpm` files, incorporate the link URL in the `<EMBED>` or `<OBJECT>` tag as described in "Setting Basic Parameters" on page 117. If the RAM file is on RealServer, the URL must not use the `ramgen` parameter.

Adding Comments to a RAM File

You can add a comment to a RAM file by using a pound sign (#) as the first character on a line. The following example shows two lines commented out of the file:

```
# Two videos and a SMIL presentation
# streamed from RealServer G2.
rtsp://realserver.company.com/media/sample1.smi
rtsp://realserver.company.com/media/video1.rm
rtsp://realserver.company.com/media/sample2.smi
```

Bundling RealPlayer Presets

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 over a 28.8 Kbps modem.
2. Check that the presentation has a preroll (initial buffering) under 15 seconds. After preroll, the presentation should not rebuffer under normal network conditions.

   Additional Information

   "What is Preroll?" on page 28.

3. Verify that video and audio quality is acceptable.
4. For a multiclip presentation, verify that clips stay synchronized. Ensure that no stalling occurs because too many clips play at the same time or a single clip requires too much bandwidth. Test that clips introduced during a presentation in progress do not stall playback by requiring too much buffering when they start.

   Tip

   If clips introduced during a presentation in progress require too much buffering, see "Smoothing Transitions between Clips" on page 163.

5. Make sure that your presentation works well for an "average" CPU for your audience. For general Web delivery, test playback on both Pentium and Power Macintosh 90 MHz machines. Do not rely on MMX technology to enhance playback. Not all Web users have MMX machines.

   Tip

   If your presentation is CPU-intensive because, for example, it uses complex Flash animation or high-bandwidth video, note this in your Web page.

6. When streaming RealAudio, ensure that sound quality is acceptable. You may need to experiment with RealAudio codecs to find the best balance between clip bandwidth use and sound quality.
7. 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, which reserves the right to refuse or edit submissions. 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 on how to do this, see **http://www.real.com/help/content/livetags.html**.

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 an intranet. This chapter provides background on using broadcast applications and RealServer G2 to broadcast media. Refer to your tools and RealServer manuals for explicit instructions on setting up a broadcast.

Introduction to Broadcasting

When a streaming presentation is delivered on demand, it 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:

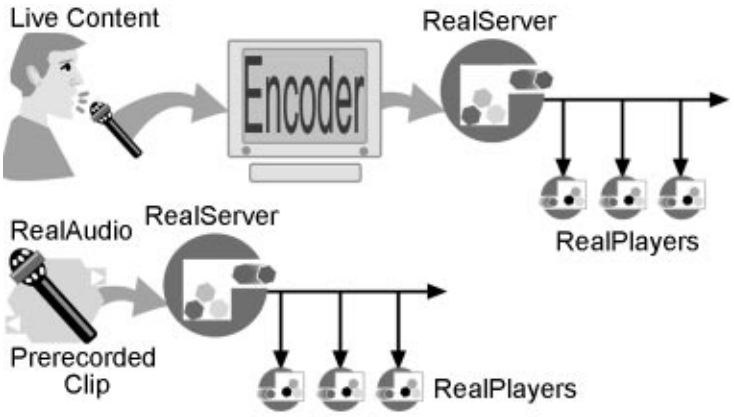
- Live content

  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 clip first.

- Prerecorded content

  Prerecorded content consists of video or audio you record and write to a digitized clip. You can then edit the clip 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 Live or Prerecorded Content



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 an 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 broadcast application.

- Broadcast application

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

machine. To broadcast prerecorded content, you typically do not need a broadcast application because RealServer can broadcast the clips itself.

RealNetworks encoding tools have live broadcast capability for audio and RGB or YUV video. Their wizards guide you through the broadcast process and let you connect to RealServer easily. RealSystem's open architecture also lets you build a broadcast application to send RealServer any type of data for broadcast. To build an application, you need the RealSystem G2 Software Development Kit (SDK).

Additional Information

Visit **http://www.real.com/products/tools/** for information on RealNetworks broadcast tools. The SDK is at **http://www.real.com/devzone/sdks/rmsdk/**.

RealPix Authoring Guide and *RealText Authoring Guide* contain applications for broadcasting RealPix and RealText presentations, respectively. To get the applications, download the bundled HTML versions of these guides from **http://service.real.com/help/library/index.html**.

- RealServer G2

RealServer streams the broadcast to your audience. From the RealServer administrator you can get the broadcast URL and the parameters for connecting a broadcast application 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 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 broadcast application for explicit instructions on how to encode and broadcast content.

Use SureStream with RealSystem G2 Codecs

SureStream technology and RealSystem G2 codecs let you broadcast RealAudio or RealVideo at multiple bandwidths. Each viewer's RealPlayer selects an encoding appropriate for its connection speed. RealNetworks encoding tools let you specify SureStream when you begin the broadcast.

To broadcast without using SureStream and the RealSystem G2 codecs, you need a separate broadcast application running on a separate computer for each bandwidth connection you intend to support. This is required because older codecs require more CPU for each encoding than do RealSystem G2 codecs. You then connect each broadcast application to RealServer and broadcast the different encodings under different URLs.

Additional Information

For more on codecs, see "Choosing RealAudio Codecs" on page 45 and "Choosing RealVideo Codecs" on page 57.

Verify that the Broadcast Machine has Enough CPU Power

Refer to your broadcast application's manual or online help for guidelines on machine requirements for broadcasting. The efficient SureStream technology can broadcast multiply encoded bandwidth streams in real-time using a moderately powerful PC.

Decide Whether to Archive a Live Broadcast

RealNetworks encoding tools let you write a live broadcast to a file. RealServer can also archive the broadcast to a file. The latter optional may be the better solution if your broadcast is long and your broadcast 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 a broadcast application 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.

Additional Information

For pointers on producing audio, see "Capturing Audio" on page 41. For video, see "Recording Video" on page 53.

During both the trial run and the live broadcast, view the broadcast output 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 broadcast output, use the broadcast application 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 broadcast application on a more powerful machine.

Create a SMIL File for a Multipart Presentation

Using SMIL, you can easily embed your broadcast in a multiclip 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 clip.

Keep in mind, however, that SMIL does not synchronize on-demand clips with the broadcast. When the SMIL presentation starts, the viewer begins to receive the on-demand clips in the order they are defined by the SMIL grouping and timing tags. The viewer joins 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 75 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 152.

APPENDIX A: Quick Steps for Streaming Clips

This appendix provides examples that show how to set up some simple streaming presentations quickly. This information will aid you if you are a beginning user. Once you are familiar with the basic steps for hosting RealSystem G2 presentations, you can easily create more complex presentations.

Note

You will need to get the correct URLs for media clips from the RealServer or Web server administrator. Substitute those values for the example URLs shown below.

Streaming a RealAudio or RealVideo Clip

It's simple to add a RealAudio or RealVideo clip to your Web page. Following the instructions below, you can stream the clip from RealServer G2 or download it from a Web server. The clip plays back in the page visitor's RealPlayer G2.

Note

Before using a Web server, read "Hosting a Presentation on a Server" on page 21 and "Limitations on Web Server Playback" on page 145.

➤ **To create the clip:**

1. Prepare your audio or video source file for encoding. This can include normalizing the audio source file or setting the video's window size.

Additional Information

"Optimizing Audio" on page 43. "Digitizing Video" on page 54.

2. Use a RealNetworks encoding tool to encode the RealAudio or RealVideo clip from your audio or video source file. A RealVideo clip uses the file extension .rm. RealAudio uses the file extension .rm or .ra. Encoding tools are available for free download or purchase at **http://www.real.com/products/tools/**.

➤ **To stream the clip from RealServer G2 using RAMGEN:**

1. Transfer the clip to the RealServer G2 directory prepared by the RealServer administrator.
2. Link your Web page to the clip with an HTML hyperlink that specifies the RealServer address, the ramgen parameter, and the HTTP protocol. You can get this information from the RealServer administrator. In your HTML source file, the link will look like this example:

```
<a href="http://realserver.company.com:8080/ramgen/content/myclip.rm">
Click here</a> to see my RealVideo presentation.
```

Additional Information
"Streaming Clips from RealServer G2" on page 140.

3. In your Web browser, click the link to verify that it works. RealPlayer will launch as a helper application and, after a few seconds of buffering, will play the streaming clip.

➤ **To play the clip back from a Web server:**

1. With any text editor, open a new file and enter the URL your clip will have on the Web server, as in the following example:

```
http://www.company.com/media/myclip.rm
```

Save this file as plain text with the file extension .ram. This is your RAM file.

Additional Information
"Creating a RAM File Manually" on page 147.

2. Transfer the clip and the RAM file to the appropriate directory on the Web server.
3. Link your Web page to the RAM file with a standard HTML hyperlink like this:

```
<a href="http://www.company.com/media/myclip.ram">
Click here</a> to see my RealVideo presentation.
```

4. In your Web browser, click the link to verify that it works. RealPlayer will launch as a helper application and, after a few seconds of buffering, will play the clip.

Additional Information

"Playing Clips from a Web Server" on page 143.

Embedding a RealVideo Clip in a Web Page

Using RealPlayer's Netscape plug-in, you can embed a RealVideo clip directly in your Web page by following the steps below after you have encoded your RealVideo clip. This example assumes that the video is 176 pixels wide by 132 pixels high. It places the video window and the full RealPlayer control panel in your Web page.

Additional Information

"Chapter 8: Playing a Presentation in a Web Page" on page 115.

➤ **To embed the clip in your Web page:**

1. In your Web page, add the `<EMBED>` tag with the RealVideo URL, window size, and `ImageWindow` control. The following example assumes RealServer G2 will stream the presentation:

```
<EMBED WIDTH=176 HEIGHT=132
SRC="http://realserver.company.com:8080/ramgen/content/myclip.rm?embed"
CONTROLS=ImageWindow CONSOLE=_master>
```

   If you intend to play the clip back from a Web server, use an HTTP URL and link to a RAM file with the extension `.rpm` (see below) as in the following example:

```
<EMBED WIDTH=176 HEIGHT=132
SRC="http://www.company.com/media/myclip.rpm"
CONTROLS=ImageWindow CONSOLE=_master>
```

2. You can then add RealPlayer controls through additional `<EMBED>` tags that all use the same URL for the `SRC` parameter. The following example embeds the full RealPlayer control panel in the Web page, linking it to the image window through the `_master` console. It assumes RealServer will stream the presentation:

```
<EMBED WIDTH=400 HEIGHT=120
SRC="http://realserver.company.com:8080/ramgen/content/myclip.rm?embed"
CONTROLS=All CONSOLE=_master>
```

Use an <EMBED> tag like the following when playing the clip back from a Web server:

```
<EMBED WIDTH=400 HEIGHT=120
SRC="http://www.company.com/media/myclip.rpm"
CONTROLS=All CONSOLE=_master>
```

Tip

Because you can place each <EMBED> tag anywhere on your Web page, you can lay out the image window and controls using HTML.

3. Transfer the clip to the appropriate RealServer G2 or Web server directory. When streaming from G2, you are ready to test the clip because the Web page already contains the link to the RealVideo clip.

➤ **To create the RAM file when playing the clip back from a Web server:**

1. With any text editor, open a new file and enter the URL your clip will have on the Web server, as in the following example:

   ```
   http://www.company.com/media/myclip.rm
   ```

   Save this file as plain text with the file extension .rpm. This is your RAM file.

2. Transfer the RAM file to the appropriate directory on the Web server.
3. In your Web browser, click the link to verify that it works.

APPENDIX B: Advanced Production Tips

This appendix will help you utilize the many features of RealSystem G2 and SMIL. Before following the production tips given here, make sure you have a good understanding of SMIL as described in Chapter 7, beginning on page 75.

Smoothing Transitions between Clips

By grouping clips played in sequence within a SMIL <par> tag, you can create smooth transitions between the clips. When clips normally play in sequence, each clip buffers data (its *preroll*) when it starts to play. By grouping the sequence within a <par> tag, though, RealSystem takes advantage of unused bandwidth to send clips' preroll before they start to playback. You can use this feature to mask preroll for high-bandwidth clips, for example, by streaming the preroll while low-bandwidth, introductory clips play.

The following SMIL example, which omits the header that defines the region layout and base URL, shows how to mask preroll for high-bandwidth clips and create smooth transitions between the initial and subsequent clips:

```
<body>
  <par>
    <seq>
      <par>
        <!-- group 1: introductory group masking preroll -->
        <audio src="intro.ra"/>
        <textstream src="titles.rt" region="left"/>
        <textstream src="credits.rt" region="right"/>
      </par>
      <par>
        <!-- group 2: main group with masked preroll -->
        <ref src="graphics.rp" region="left"/>
        <video src="story.rm" region="right"/>
```

```
    </par>
   </seq>
  </par>
</body>
```

Group 1 consists of two RealText clips and a RealAudio clip played in parallel. Because of the `<seq>` tag, the group 1 clips precede the RealPix and RealVideo clips in group 2. If the `<seq>` tag were the highest level of organization, RealServer would stream the group 1 clips without regard to group 2, streaming data for group 2 clips only after group 1 finished. Viewers would experience a delay after group 1 clips finished as RealServer streamed preroll for group 2 clips.

The outer `<par>` tag just below the `<body>` tag, however, makes RealSystem treat groups 1 and 2 as one large parallel group with subgroups played in sequence. Although this doesn't affect the order in which the clips play, the parallel grouping makes RealSystem balance bandwidth between all clips. After it starts to stream the group 1 clips, RealServer makes use of unused bandwidth by streaming data for group 2 clips while the group 1 clips play. This masks the preroll for the group 2 clips.

Additional Information

See "What is Preroll?" on page 28 and "Developing Multiclip Presentations" on page 33.

Hiding Regions with z-index

RealPlayer creates all regions defined in a SMIL file's header section when it first reads the file. So a single SMIL presentation cannot play clips in a certain set of regions, then destroy those regions and create different regions with a new layout. To accomplish dynamic layout changes, you can create separate SMIL files that define each region set, then play the SMIL files in sequence as described in "Creating a RAM File Manually" on page 147.

An alternative is to create a single SMIL file that uses the `<region>` tag's `z-index` parameter to create transparent, overlaying regions. The following SMIL header example creates a `left` region next to a `right` region. Both are in portrait orientations, their heights over twice the size of their widths. A second set of regions, `top` and `bottom`, are stacked. These regions have higher z-index values, meaning that they display in front of the `left` and `right` regions:

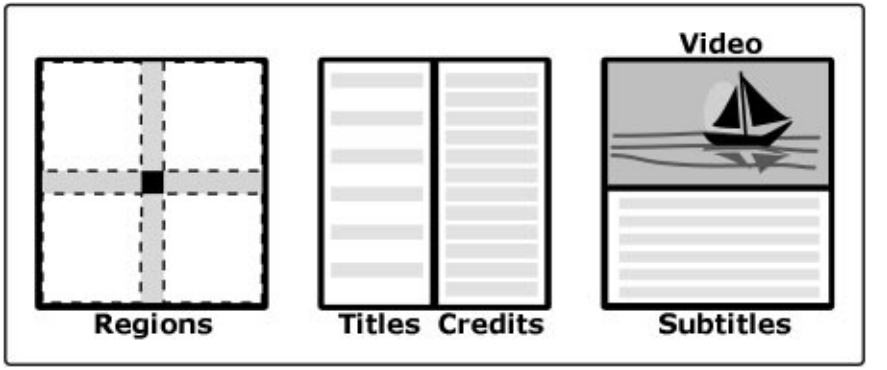
```
<head>
 <layout>
  <root-layout width="360" height="360"/>
  <!-- first two side-by-side regions -->
  <region id="left" top="10" left="10" width="165" height="340" z-index="0"/>
  <region id="right" top="10" left="185" width="165" height="340" z-index="1"/>
  <!-- second two stacked regions -->
  <region id="top" top="10" left="70" width="220" height="165" z-index="2"/>
  <region id="bottom" top="185" left="70" width="220" height="165" z-index="3"/>
 </layout>
</head>
```

As defined in the following SMIL body, RealPix and RealText clips first play in the `left` and `right` regions, which appear beneath the `top` and `bottom` regions. Because the overlaying `top` and `bottom` regions do not use background colors, they remain transparent until clips play in them. The introductory RealPix and RealText clips disappear when they finish playback, restoring to view the `root-layout` default background color of black. The group 2 clips, a RealVideo clip and another RealText clip, then play in the `top` and `bottom` regions:

```
<body>
 <par>
  <seq>
   <par>
    <!-- group 1: side-by-side titles and credits -->
    <ref src="titles.rp" region="left" fill="remove"/>
    <textstream src="credits.rt" region="right" fill="remove"/>
   </par>
   <par>
    <!-- group 2: stacked video and subtitles -->
    <video src="story.rm" region="top"/>
    <textstream src="subtitles.rt" region="bottom"/>
   </par>
  </seq>
 </par>
</body>
```

Although the `left`, `right`, `top`, and `bottom` regions exist from the start of the SMIL file playback, the use of `z-index`, `fill="remove"`, and default region transparency makes it appear as if the regions are created dynamically with each new set of clips. The following figure illustrates the initial region creation, the first set of clips, then the second set of clips playing in the RealPlayer window.

Transparent Region Creation and Ordering



Additional Information

For more on z-index, see "Ordering Overlapping Regions with z-index" on page 99. "Laying Out Multiple Clips" on page 93 discusses SMIL layouts.

Writing Complex SMIL Switch Statements

The SMIL <switch> tag is a powerful feature that lets you specify options that each RealPlayer can choose between based on its preference settings and available bandwidth. "Switching Between Alternate Choices" on page 91 explains the basics of using the <switch> tag. The following sections give tips on writing complex <switch> statements.

Switching with SureStream Clips

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

- When the presentation consists solely of a SureStream 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 SureStream clip with other clips encoded for single bandwidths. The SureStream clip is always used, but the <switch> group gives RealPlayer options for other clips. The following

example illustrates a RealAudio SureStream 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 SureStream, see "Choosing RealAudio Codecs" on page 45 and "Choosing RealVideo Codecs" on page 57. Refer to "Supporting Multiple Bandwidth Connections" on page 35 for more on developing presentations for different connection speeds.

Switching with Multiple Test Attributes

You can use multiple <switch> test attributes to have RealPlayer choose clips based on both bandwidth and language. There are two ways to do this. In this first example, each audio clip choice has two test-attributes, one for language and one for bandwidth. Both attributes must be viable for RealPlayer to choose the clip:

```
<switch>
  <!-- French language choices -->
  <audio src="sound/audio_fr2.ra" system-language="fr" system-bitrate="47000"/>
  <audio src="sound/audio_fr1.ra" system-language="fr" system-bitrate="20000"/>
  <!-- English language choices (default) -->
  <audio src="sound/audio_en2.ra" system-bitrate="47000"/>
  <audio src="sound/audio_en1.ra" system-bitrate="20000"/>
</switch>
```

Because RealPlayer evaluates the <switch> choices from top to bottom, selecting the first viable option, the last two choices do not have system-language options. This lets all RealPlayers other than those with French as their language preference choose between the two English-language clips.

The next example adds RealText files in both French and English to the presentation possibilities. Here, <switch> statements are nested so that RealPlayers with French set as their language preference play the French RealText file and choose from the set of French-language RealAudio files

based on available bandwidth. All other RealPlayers play the English RealText file and choose from a set of English-language RealAudio files:

```
<switch>
  <!-- Choose French as the language -->
  <par system-language="fr">
    <textstream src="text/credits_fr.rt"/>
    <switch>
      <!-- Choose fast or slow bit rate for French audio -->
      <audio src="sound/audio_fr2.ra" system-bitrate="47000"/>
      <audio src="sound/audio_fr1.ra" system-bitrate="20000"/>
    </switch>
  </par>
  <!-- Choose English (default) as  the language -->
  <par>
    <textstream src="text/credits_en.rt"/>
    <switch>
      <!-- Choose fast or slow bit rate for English audio -->
      <audio src="sound/audio_en2.ra" system-bitrate="47000"/>
      <audio src="sound/audio_en1.ra" system-bitrate="20000"/>
    </switch>
  </par>
</switch>
```

APPENDIX C: File Type Reference

The following table provides a quick reference to file types commonly used in RealSystem G2 media production. This is not a definitive list of all file types used in RealSystem G2, however. Plug-in technology allows RealSystem G2 to stream virtually any file type.

Additional Information

Refer to **http://www.real.com** for information on other types of media RealSystem G2 can stream.

RealSystem G2 Streaming Media Clip File Types

Extension	File Type	Reference
.ra or .rm	RealAudio	"Creating RealAudio Clips" on page 44
.rm or .rv	RealVideo	"Creating RealVideo Clips" on page 57
.rp	RealPix streaming image mark-up	"Images" on page 12
.rt	RealText streaming text	"Text" on page 13
.swf	Shockwave Flash file for RealFlash	"Producing Animation" on page 63

RealSystem G2 Information Files

Extension	File Type	Reference
.prx	RealPlayer Presets export file	"Bundling RealPlayer Presets" on page 149
.ram	RAM file to launch RealPlayer	"Streaming Clips from RealServer G2" on page 140
.rpm	RAM file for embedded presentations	
.smi	SMIL file for layout and timing	"Assembling a Presentation with SMIL" on page 75

Video File Types Encodable as RealVideo

Extension	File Type	Reference
.avi	Windows standard video	"Creating RealVideo Clips" on page 57
.mov	Uncompressed QuickTime	

Common Video File Types Streamed by RealSystem G2

Extension	File Type	Reference
.asf	Advanced Streaming Format	"Streaming Other Video Formats" on page 60
.avi	Windows standard video	
.mov	QuickTime video	
.viv	Vivo video	

Audio File Types Encodable as RealAudio

Extension	File Type	Reference
.aif	Audio Interchange Format	"Creating RealAudio Clips" on page 44
.au	Sun-standard audio	
.mov	QuickTime audio	
.snd	Sound audio	
.wav	Wave audio	

Common Audio File Types Streamed by RealSystem G2

Extension	File Type	Reference
.aif	Audio Interchange Format	"Streaming Other Audio Formats" on page 49
.au	Sun-standard audio	
.wav	Wave audio	

Image Files Types Playable Directly in RealPlayer G2

Extension	File Type	Reference
.jpg	JPEG (nonprogressive)	"Images" on page 12
.gif	GIF87, GIF89a, or animated GIF	

Image Files Types Usable in RealPix Presentations

Extension	File Type	Reference
.bmp	Windows bitmap	"Images" on page 12
.jpg	JPEG (nonprogressive)	
.gif	GIF87 or GIF89a	
.stg	STiNG	

APPENDIX D: SMIL Language Codes

As "Setting Language Choices" on page 91 explains, SMIL 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 clips created for specific languages.

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 each second.

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

bit rate
A measure of bandwidth, expressed as 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
Receiving and storing data before playing it back. The initial buffering time is called "preroll". After preroll, excessive buffering may stall the presentation.

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 G2 is a client of RealServer G2.

clip
A media file within a presentation. Clips typically have an internal timeline, as with RealAudio and RealVideo.

codec
Compressor/decompressor. Codecs convert data between uncompressed and compressed formats, reducing the bandwidth a streaming clip consumes.

D

download
To send a file over a network with a nonstreaming protocol such as HTTP. Contrast to "stream".

E

encoding
Converting a file into a compressed, streaming format. For example, you can encode WAV 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 clips 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 clips.

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. Also equal to 8 Kilobits.

P

PNA
A proprietary protocol RealServer G2 supports for backwards compatibility with RealSystem 3.0 through 5.0. URLs using PNA start with `pnm://`.

port
A connection to a server designated by a number such as 8080. RealServer G2 uses different ports for the RTSP, HTTP, and PNA protocols.

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 clip type for streaming audio over a network. RealAudio clips use the `.ra` or `.rm` extension.

RealFlash
A RealSystem clip type for streaming Shockwave Flash animation along with a RealAudio soundtrack.

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

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

RealServer
RealNetworks server software used to stream multimedia presentations to RealPlayer.

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

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

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

RealVideo
A RealSystem clip type for streaming video over a network. RealVideo clips 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 clips to RealPlayer G2 or any RTP-based client. Contrast to "HTTP".

S

server
1. A software application, such as a Web server or RealServer G2, that sends clips over a network.

2. The computer that runs server software.

Shockwave Flash
A compressed Flash file format (file extension .swf) used as the animation component in RealFlash.

SMIL
Synchronized Multimedia Integration Language. A mark-up language for specifying how and when each clip plays.

stream
v. To send a media clip over a network so that it begins playing back as quickly as possible.

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

SureStream
A technology that lets you encode a single RealAudio or RealVideo clip for multiple bandwidths. Available only in RealSystem G2.

U

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

INDEX