# SUMMARY OF CONTENTS

<table>
<thead>
<tr>
<th>Part</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PART I: GETTING STARTED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>NEW FEATURES</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>FIREWALLS</td>
<td>13</td>
</tr>
<tr>
<td>PART II: INSTALLATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>INSTALLATION</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>STARTUP AND SHUTDOWN</td>
<td>39</td>
</tr>
<tr>
<td>5</td>
<td>HELIX ADMINISTRATOR</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>SNMP</td>
<td>69</td>
</tr>
<tr>
<td>PART III: CLIENT TECHNOLOGIES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IOS DELIVERY</td>
<td>85</td>
</tr>
<tr>
<td>8</td>
<td>RATE CONTROL</td>
<td>111</td>
</tr>
<tr>
<td>9</td>
<td>PLAYLIST MANAGEMENT</td>
<td>143</td>
</tr>
<tr>
<td>PART IV: BROADCASTING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>BROADCAST METHODS</td>
<td>185</td>
</tr>
<tr>
<td>11</td>
<td>MULTICASTS</td>
<td>203</td>
</tr>
<tr>
<td>12</td>
<td>SPLITTING</td>
<td>213</td>
</tr>
<tr>
<td>13</td>
<td>SIMULATED LIVE BROADCASTS</td>
<td>229</td>
</tr>
<tr>
<td>14</td>
<td>CHANNEL SWITCHING</td>
<td>255</td>
</tr>
<tr>
<td>PART V: APPENDIXES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>NONSTANDARD INSTALLATIONS</td>
<td>271</td>
</tr>
<tr>
<td>INDEX</td>
<td></td>
<td>277</td>
</tr>
</tbody>
</table>
# CONTENTS

## INTRODUCTION

- Audience for this Guide ................................................................. 1
- How this Guide Is Organized .......................................................... 1
- Conventions Used in this Manual .................................................... 3
  - Terminology ................................................................................. 3
  - Typographical Conventions ......................................................... 3

## PART I: GETTING STARTED

### 1 NEW FEATURES

- New Features in Helix Universal Server Version 14.3 ..................... 7
- Flash Live Broadcasts ...................................................................... 7
- Encoder Connection Improvements ................................................. 8
- Updated iOS Support ...................................................................... 8
- Playlist Management for iOS Devices ........................................... 9
- MPEG-4 Broadcast Archives ......................................................... 9
- Bookmarking Updates .................................................................... 9
- Features from Earlier Releases ....................................................... 9
- Flash On-Demand Delivery ............................................................ 10
- 64-Bit Support ............................................................................. 10
- Silverlight Support ....................................................................... 10
- HTTP 1.1 .................................................................................... 10
- H.264 at High Bit Rates ............................................................... 11
- Documentation Changes ............................................................... 11
- Compatibility with Previous Versions ........................................... 11

### 2 FIREWALLS

- Media Server Placement ............................................................... 13
- Host Addresses for NAT Firewalls ................................................. 14
- Virtual IP Addresses .................................................................... 14
  - How Virtual IP Addressing Works ........................................... 14
  - The Problem with Virtual Addressing ..................................... 15
  - Resolving the Problem ........................................................... 15
- RTSP Communication .................................................................. 15
- RTSP Control Channel ............................................................... 16
PART II: INSTALLATION

3 INSTALLATION

Operating System Modifications ................................................................. 23
Hyperthreading ...................................................................................... 23
File Descriptors for Linux and Solaris ......................................................... 23
Extended FILE Support on Solaris 10 for iOS Devices ............................... 26
Solaris 10 Update 6 for SNMP ................................................................. 26
Windows 2008 Account Privileges .......................................................... 26
Windows Default Buffer Size ................................................................ 26
Installing Helix Server ............................................................................ 27
Installing Helix Proxy ............................................................................... 29
Obtaining a License Key ......................................................................... 31
Selecting Port Values ............................................................................ 31
Administration Ports ............................................................................. 32
Streaming Protocol Ports ...................................................................... 32
Feature Control Ports ........................................................................... 33
Transport Layer Protocols .................................................................... 34
Creating Certificate Files ...................................................................... 34
Specifying Certificate Information ......................................................... 35
Requesting a Signed Certificate .............................................................. 35
Using Self-Signed Certificates ................................................................. 36
Running the OpenSSL Program ............................................................... 36
Reinstalling the Server or Proxy ............................................................ 36

4 STARTUP AND SHUTDOWN

Starting the Server or Proxy on Windows ................................................. 39
Running a Windows Service ................................................................ 39
Modifying Access Privileges ................................................................ 40
Using the Start Menu or a Desktop Icon ............................................... 41
Starting Up from the Windows Command Line ...................................... 42
Starting the Server or Proxy on UNIX or Linux ...................................... 42
Defining Master Agent Addresses and Ports..................................................73
Setting Up SNMP Security............................................................................74
Defining a View Access Control Model..........................................................75
Running the Master Agent on Windows.......................................................78
Starting the Master Agent on UNIX ..............................................................79
Running a Management System ....................................................................79
Monitor Tree ..............................................................................................80
Configuration Tree on Helix Server ...............................................................81
Configuration Tree on Helix Proxy .................................................................81
Control Tree ...............................................................................................82

PART III: CLIENT TECHNOLOGIES

7 IOS DELIVERY .........................................................................................85
Segment Delivery Overview..........................................................................85
iOS Source Media Requirements .................................................................86
Multi-Rate Streams .....................................................................................86
Gzip Support for Playlists.............................................................................87
Compatibility with Other Features ...............................................................88
On-Demand Segmentation Steps ...................................................................88
Broadcast Segmentation Steps .......................................................................90
Using the m3ugen Mount Point .....................................................................91
Requesting Playlist Files Directly .................................................................92
Selecting the Initial Rate ..............................................................................92
Bookmarking Presentations ..........................................................................93
Using Third-Party Query Parameters .............................................................94
Defining Segmentation Mount Points .............................................................94
Predefined Mount Points for Segmentation ..................................................94
User-Defined Mount Points for Segmentation ..............................................95
Defining Source and Segmentation Mount Points ........................................96
Using Archive Mount Points ........................................................................97
Purging Files from Mount Points .................................................................98
Redirecting Clients on Errors .......................................................................100
Setting Up Mount Points in Helix Administrator ..........................................101
File Names and Directory Structures ..........................................................102
Subdirectory Creation ..................................................................................103
Playlist File ...............................................................................................103
Timestamp Format ......................................................................................104
Top Segment Directory ...............................................................................105
Multi-rate Playlist Files ..............................................................................105
Multi-rate Subdirectories ............................................................................105
Segment File Names ....................................................................................106
## Contents

- Omitted Characters ................................................................. 106
- Encrypting Segments ................................................................. 106
- Key File .......................................................................................... 107
- Writing Key Files to Alternate Locations ............................................... 108
- Progressive Download Alternative ...................................................... 109
  - File Formats for Progressive Download .............................................. 109
  - Codecs for Progressive Download ......................................................... 109
  - Hosting Downloadable Content ............................................................ 109

### 8 RATE CONTROL

- Server-Side Rate Control Basics ..................................................... 111
  - Rate Control Basic Requirements ....................................................... 111
  - Types of Rate Control .......................................................................... 112
  - Live Broadcasts and Rate Control ......................................................... 113
  - Media Formats Used with Server-Side Rate Control .............................. 114
  - Media Encoding Guidelines ................................................................. 115
- User Agent Profiles ........................................................................... 116
  - UAS Profiles ....................................................................................... 116
  - Configuration File Profiles ................................................................. 117
  - RDF Files ............................................................................................ 117
  - Profile Value Inheritance ...................................................................... 117
  - User Agent Strings ............................................................................ 117
  - Profiles Shipped with Helix Server ....................................................... 120
- Streaming Rate Selection .................................................................... 122
  - Maximum Sending Rate ....................................................................... 122
  - Initial Rate Selection ........................................................................... 123
  - Initial Oversend Rate ........................................................................... 124
- Receiver and Sender Reports ............................................................... 124
  - Report Frequency ............................................................................... 125
  - NADU Feedback .................................................................................. 126
  - Receiver Report Timeout ..................................................................... 126
- Buffer Modeling .................................................................................. 127
  - Data Size Versus Playback Time ........................................................... 127
  - Buffer Size ......................................................................................... 128
  - Preroll ................................................................................................ 128
  - Target Time ......................................................................................... 130
- Downshifting ....................................................................................... 131
  - Downshift Depth Settings ................................................................... 132
  - Stepwise Downshifting ....................................................................... 133
  - Client-Initiated Downshift .................................................................... 134
  - Streaming Below the Lowest Media Rate .............................................. 134
- Upshifting ............................................................................................ 135
### Contents

- Error Conditions .......................................................................................176
- HTTP Status Codes ...................................................................................177
- Configuring Helix Server for Playlist Control ....................................................178
  - Set Playlist HTTP Ports ..............................................................................178
  - Define Playlist Mount Points ......................................................................179
  - Secure the Playlist Management System ......................................................179
  - Enable iOS Segmentation ..........................................................................181
  - Enable Playlist Control Requests ..............................................................181

### PART IV: BROADCASTING

#### 10 Broadcast Methods

- Understanding Unicasts .............................................................................185
  - H.264/AAC Broadcasts .............................................................................185
  - Broadcast Archives ....................................................................................186
  - Broadcast Redundancy ..............................................................................187
  - Standby Messages .....................................................................................187
  - Bandwidth Constraints .............................................................................187
  - Helix Push Broadcast Method ..................................................................187
  - Helix Push Broadcast Connection Steps ...............................................188
  - Helix Push Setup .......................................................................................189
  - Helix Advanced Push and Multicast Push Broadcast Methods ......................189
  - Helix Advanced Push Broadcast Connection Steps ..................................189
  - Helix Advanced Push Setup ....................................................................190
  - Helix Pull Broadcast Method ..................................................................191
  - Helix Pull Broadcast Connection Steps ................................................191
  - Helix Pull Setup .........................................................................................192
  - RTP Push Broadcast Method ..................................................................192
  - RTP Push Broadcast Requirements .........................................................193
  - RTP Push Broadcast Connection Steps ....................................................193
  - RTP Push Setup .........................................................................................194
  - SDP Requirements ...................................................................................195
- Flash Broadcasts ...........................................................................................199
  - Flash Broadcast Connection Steps ........................................................200
  - Flash Broadcast Setup ...............................................................................200
- Windows Media Broadcasts ...........................................................................200
  - Windows Media Push and Pull Modes .......................................................201
  - Windows Media Broadcast Setup ............................................................201

#### 11 Multicasts

- Understanding Multicasts ........................................................................203
- Publicizing Multicasts ...............................................................................203
- Multicasting Used with Other Features .........................................................204
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast Resources</td>
<td>205</td>
</tr>
<tr>
<td>Network Configuration for Multicasts</td>
<td>206</td>
</tr>
<tr>
<td>Multicast Addresses</td>
<td>206</td>
</tr>
<tr>
<td>Packet Time to Live</td>
<td>207</td>
</tr>
<tr>
<td>Multicasts with Multiple Network Interface Cards</td>
<td>207</td>
</tr>
<tr>
<td>Back-Channel Multicasting</td>
<td>207</td>
</tr>
<tr>
<td>Unicast Failovers</td>
<td>208</td>
</tr>
<tr>
<td>Back-Channel Multicasting Configuration</td>
<td>209</td>
</tr>
<tr>
<td>Scalable Multicasting</td>
<td>209</td>
</tr>
<tr>
<td>Session Description Files</td>
<td>209</td>
</tr>
<tr>
<td>Authentication and Statistics</td>
<td>210</td>
</tr>
<tr>
<td>Unicast Failovers</td>
<td>210</td>
</tr>
<tr>
<td>Scalable Multicasting Configuration</td>
<td>210</td>
</tr>
<tr>
<td>Windows Media Multicasts</td>
<td>211</td>
</tr>
<tr>
<td>Windows Media Multicast Configuration</td>
<td>211</td>
</tr>
<tr>
<td><strong>12 SPLITTING</strong></td>
<td></td>
</tr>
<tr>
<td>Understanding Splitting</td>
<td>213</td>
</tr>
<tr>
<td>Definitions</td>
<td>213</td>
</tr>
<tr>
<td>Push Splitting</td>
<td>214</td>
</tr>
<tr>
<td>Pull Splitting</td>
<td>215</td>
</tr>
<tr>
<td>One-to-Many Splitting</td>
<td>216</td>
</tr>
<tr>
<td>One-to-One Chaining</td>
<td>217</td>
</tr>
<tr>
<td>Setting Up Split Broadcasts</td>
<td>217</td>
</tr>
<tr>
<td>Helix Producer Broadcasts</td>
<td>218</td>
</tr>
<tr>
<td>SLTA Broadcasts</td>
<td>219</td>
</tr>
<tr>
<td>RTP-based Broadcasts</td>
<td>219</td>
</tr>
<tr>
<td>Flash Broadcasts</td>
<td>220</td>
</tr>
<tr>
<td>Windows Media Broadcasts</td>
<td>220</td>
</tr>
<tr>
<td>Splitting Different Streams to Different Receivers</td>
<td>221</td>
</tr>
<tr>
<td>Virtual Paths for Stream Direction</td>
<td>221</td>
</tr>
<tr>
<td>Stream Routing Example</td>
<td>222</td>
</tr>
<tr>
<td>Multicasting Split Streams</td>
<td>223</td>
</tr>
<tr>
<td>Unicast Delivery, Multicast Distribution</td>
<td>223</td>
</tr>
<tr>
<td>Dual Unicast and Multicast Transport Methods</td>
<td>224</td>
</tr>
<tr>
<td>Creating Redundant Streams</td>
<td>224</td>
</tr>
<tr>
<td>Transmitter Redundancy</td>
<td>225</td>
</tr>
<tr>
<td>Transport Redundancy</td>
<td>226</td>
</tr>
<tr>
<td><strong>13 SIMULATED LIVE BROADCASTS</strong></td>
<td></td>
</tr>
<tr>
<td>Understanding Simulated Live Broadcasts</td>
<td>229</td>
</tr>
<tr>
<td>Broadcast Formats</td>
<td>230</td>
</tr>
<tr>
<td>Basic and Advanced Modes</td>
<td>230</td>
</tr>
</tbody>
</table>
Contents

Helix Server Setup ......................................................................................231
Command Line Operation ............................................................................231
SLTA Quick Start Tutorials .........................................................................232
  Quick Start for SLTA Basic Mode ..........................................................232
  Quick Start for SLTA Advanced Mode ....................................................233
Configuring SLTA for Advanced Mode .......................................................235
  Using the Configuration Template ..........................................................236
  Setting Basic Transmitter Properties ......................................................237
  Defining Push Splitting ...........................................................................238
  Defining Pull Splitting ............................................................................241
Creating a Playlist .......................................................................................242
  Writing a Basic Playlist .........................................................................242
  Adding Title, Author, and Copyright Information ....................................243
Running SLTA ...............................................................................................244
  Running SLTA on a Different Machine ....................................................244
  Setting Environment Variables ..............................................................245
  Starting SLTA ..........................................................................................246
  Using SLTA Options ..............................................................................249
  Monitoring and Stopping SLTA ...............................................................252
Linking to the Simulated Live Broadcast ...................................................252
  Writing Basic Mode Links ....................................................................252
  Creating Push Splitting Links .................................................................253
  Writing Pull Splitting Links ....................................................................253

14 CHANNEL SWITCHING ...........................................................................255

Understanding Channel Switching ............................................................255
Third-Party Components ............................................................................257
Codecs and File Formats ...........................................................................257
Protocols and Transports ...........................................................................259
User Authentication ....................................................................................259
Error Conditions ........................................................................................259
Compatibility with Other Features ............................................................260
Channel Controller .....................................................................................261
  Channel Switching Control Port ............................................................261
  HTTP Request Parameters ...................................................................261
  HTTP Status Codes ..............................................................................264
Channel Switching Workflows ....................................................................265
  Initial Channel Playback .......................................................................265
  Switch to a New Channel .......................................................................266
  End Session ............................................................................................268
Configuring Channel Switching .................................................................268
# PART V: APPENDIXES

## A  NONSTANDARD INSTALLATIONS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installing the Server and Proxy on the Same Machine</td>
<td>271</td>
</tr>
<tr>
<td>Machine Requirements for Dual Use</td>
<td>271</td>
</tr>
<tr>
<td>Installing the Server for Dual Use</td>
<td>272</td>
</tr>
<tr>
<td>Installing the Proxy for Dual Use</td>
<td>273</td>
</tr>
<tr>
<td>Starting the Server with a Heartbeat Option</td>
<td>274</td>
</tr>
<tr>
<td>Installing Helix Server with a Web Server</td>
<td>275</td>
</tr>
<tr>
<td>Use Nonstandard Ports for Helix Server</td>
<td>275</td>
</tr>
<tr>
<td>Bind Helix Server to Specific IP Addresses</td>
<td>275</td>
</tr>
</tbody>
</table>

## INDEX

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX</td>
<td>277</td>
</tr>
</tbody>
</table>
INTRODUCTION

Welcome to Helix™ Universal Server Version 14.3. This guide explains how to install and run Helix Server and Helix Proxy. It covers complex feature setup and describes how to integrate the server and proxy with third-party software.

Audience for this Guide

This guide is intended for technical system administrators who will manage media stream but not necessarily create the content. Information services professionals, server administrators, and Web masters may also find this book useful.

How this Guide Is Organized

This administration guide contains the following chapters and appendixes.

Chapter 1: New Features
If you’re familiar with previous versions of Helix Server, this chapter will give you a quick update on the new features found in Helix Server.

Chapter 2: Firewalls
If you’re streaming media to users on the Internet, you’ll need to know how Helix Server, Helix Proxy, and other RealNetworks products interact with firewalls.

Chapter 3: Installation
This chapter explains how to prepare your operating system and install Helix Server or Helix Proxy.

Chapter 4: Startup and Shutdown
Helix Server and Helix Proxy support a number of startup parameters that affect memory use and other features.
Chapter 5: Helix Administrator
Helix Administrator is the browser-based administration tool that you use to configure Helix Server or Helix Proxy.

Chapter 6: SNMP
This chapter explains how to configure the Simple Network Monitoring Protocol (SNMP) plug-in and master agent to monitor Helix Server activity using third-party monitoring software.

Chapter 7: iOS Delivery
Helix Server can automatically segment content and deliver it to iOS devices such as iPhones and iPads.

Chapter 8: Rate Control
The server-side rate control feature can vary the streaming rate of a clip that is encoded for multiple bandwidths. This chapter explains how to modify rate control for specific media players.

Chapter 9: Playlist Management
The server-side playlist feature streams a sequence of clips or broadcasts in a single RTSP session.

Chapter 10: Broadcast Methods
This chapter explains the various broadcast methods that you can use to deliver a stream from a live encoder to Helix Server.

Chapter 11: Multicasts
Multicasts conserve bandwidth by delivering a single stream to all media players on a multicast-enabled network.

Chapter 12: Splitting
Splitting is a method of delivering a stream from a RealNetworks encoder to Helix Server or from one Helix Server to another.

Chapter 13: Simulated Live Broadcasts
This chapter explains how to deliver archived or other on-demand content as if it were a live broadcast.

Chapter 14: Channel Switching
This chapter covers fast channel switching, which allows a user to change streams within a single RTSP session.
Appendix A: Nonstandard Installations

This appendix complements Chapter 3 by explaining less common methods for installing Helix Server and Helix Proxy.

Conventions Used in this Manual

This section explains some conventional terms and formats used throughout the book.

Terminology

- Because this guide is designed for the Helix Server administrators, the term you refers to the administrator. Persons who play clips served by Helix Server are referred to as visitors, viewers, or users.

- Media players such as RealPlayer or Windows Media Player are referred to as media players or, more generically, as clients. Where information applies specifically to the RealNetworks® RealPlayer, this is clearly stated.

- The terms clips, content, media clips, and media files are used interchangeably to indicate the material that Helix Server streams.

- Production tools used to create the media clips that Helix Server streams are referred to collectively as encoders.

Typographical Conventions

The following table explains the typographic conventions used in this manual.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>syntax</td>
<td>This font is used for syntax of configuration files, URLs, or command-line instructions.</td>
</tr>
<tr>
<td>variables</td>
<td>Italic text represents variables. Substitute values appropriate for your system.</td>
</tr>
<tr>
<td>emphasis</td>
<td>Bold text is used for emphasis.</td>
</tr>
<tr>
<td>. . .</td>
<td>Ellipses indicate nonessential information omitted from examples.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Square brackets indicate optional material. If you choose to use the material within the brackets, don’t type the brackets themselves. An exception to this is in the basic access log, where statistics generated by the StatsMask variable are enclosed in regular brackets.</td>
</tr>
</tbody>
</table>
In this section, you learn about basic installation issues, such as how to integrate Helix Server and Helix Proxy with firewalls. You will find it useful to review this information before installing the products.
NEW FEATURES

This chapter describes new features in Helix Server, and covers upgrade issues from previous versions of Helix Server.

New Features in Helix Universal Server Version 14.3

The following sections describe features added to Helix Universal Server Version 14.3 and Helix Universal Proxy Version 14.3. For additional details about new features, see the New Features section of the Helix Administrator online help.

Flash Live Broadcasts

Helix Server now supports live, RTMP-based streams from Flash encoders in addition to on-demand Flash streaming:

- Helix Server can stream live Flash streams using VP6 video along with audio codecs such as MP3 and Nellymoser to Flash Player.
- Helix Server can deliver live H.264/AAC content generated by a Flash encoder to Flash Player, RTSP media players, and iOS devices.
- Helix Server can deliver live H.264/AAC content generated by a different encoder (such as Helix Producer) to Flash Player as if the stream originated from a Flash encoder.
- Helix Administrator includes a setup page (Broadcasting > Flash Media Encoding) where you define allowable Flash encoder connections.

Note: Helix Proxy can cache on-demand F4V files or split live Flash content (H.264/AAC) for RTSP players. Proxying Flash content to Flash Player over RTMP is not supported, however. For information about delivering Flash content over RTSP, refer to the Helix Server online help.
For More Information: See “Flash Broadcasts” on page 199. For detailed Flash broadcasting information, refer to the Helix Administrator online help section Helix Administrator > Broadcasting > Flash Media Encoding.

Encoder Connection Improvements

Helix Server has been optimized to utilize all server hardware CPUs (including multiple cores) when receiving live broadcast streams from any encoder, such as Helix Producer or a Flash live encoder. This allows Helix Server to receive up to four times the number of live, incoming streams than in previous releases, depending on the CPU.

Updated iOS Support

Helix Server can automatically segment on-demand and live broadcast contents into MPEG-2 transport streams for delivery to iOS devices. Chapter 7 explains the features and options of the segmentation feature. In Helix Administrator, you define segmentation under Content Management > Media Segmentation.

This version of Helix Server includes several changes to how media is delivered to iOS devices:

- You can add an ir=streaming_rate parameter to a request to set the initial delivery rate of the audio and video combined. See “Selecting the Initial Rate” on page 92.
- The MIME type for playlist files (.m3u8) has changed. MIME types for iOS content are now set in the MPEG2_Transport_Stream list of the Helix Server configuration file. See “Verifying MIME Types” on page 101.
- Gzip compression for playlist files is supported.
- An optional startup purge can delete content segmented for selected source mount points. See “Purging Segments on Restart” on page 99.
- The /rtmplive/ broadcast mount point for delivering live Flash content is now a default source for segmentation.

Note: iOS support is not available with Helix Proxy.
Playlist Management for iOS Devices

The playlist management feature described in Chapter 9 has been extended to iOS devices. These devices can use internally controlled or noncontrolled sessions (see “Types of Playlist Sessions” on page 144). External directives such as chapter jumping are not supported.

For More Information: For information about playlist management features specific to iOS devices, see the section “Segment Handling with iOS Devices” on page 148.

MPEG-4 Broadcast Archives

A Helix Server broadcasting H.264/AAC content can now create an archive file (or set of files) in the MPEG-4 (.mp4) format. For details, refer to the Helix Administrator online help topic Helix Administrator > Broadcasting > Live Archiving.

Bookmarking Updates

The Helix Server bookmarking feature has been updated:

• Helix Server now supports bookmarks in RTMP URLs for Flash Player.

• Bookmarking for iOS devices now uses a timing value in seconds rather than the segment number. This makes it consistent with bookmarks for RTSP and RTMP media players.

• Helix Administrator includes a bookmarking configuration page (Content Management > Bookmarking) where you can define the bookmark query parameter (start by default) and set optional buffering values for RTSP, RTMP, and iOS media clients.

For More Information: For details about using content bookmarking, see the Helix Administrator online help topic Helix Administrator > Content Management > Bookmarking.

Features from Earlier Releases

Helix Universal Server Version 14.3 and Helix Universal Proxy Version 14.3 include the following features, which were introduced in earlier releases of Version 14 software.
Flash On-Demand Delivery

Helix Server supports native delivery of FLV and F4V content to Flash clients using the RTMP protocol. It can also stream H.264/AAC content in an MPEG-4 or 3GPP file format to Flash clients. Likewise, it can stream H.264/AAC content in the F4V format to RTSP-based clients.

**Tip:** The SLTA utility has a new option, `-fp`, which is needed when streaming AAC audio to Flash clients. See “Forcing RTP Packetization for AAC Audio” on page 251.

**For More Information:** For detailed information about on-demand Flash delivery, refer to the Helix Administrator online help section Streaming Basics > Media Types > Flash Content.

64-Bit Support

Helix Server and Helix Proxy are available as 32-bit applications for 32-bit operating systems and as 64-bit applications on 64-bit operating systems. For the different versions, installation, set-up, and feature availability are identical. The primary differences are the amount of system memory the applications can access and the sizes of files they can handle.

**For More Information:** For details about memory use, see “Memory Allocation” on page 47. After installing the products, refer to the Helix Administrator online help topic Streaming Basics > 32-Bit and 64-Bit Application Differences.

Silverlight Support

Helix Server supports Silverlight clients with content in the Windows Media format delivered over WM-HTTP. H.264/AAC content is also supported using HTTP progressive download.

HTTP 1.1

Helix Server and Helix Proxy support HTTP 1.1. This includes support for HTTPS, pipelining, byte range requests, and persistent connections. Changes include the following:

• HTTPS port value set by default to 443.
• HTTPS port for Helix Administrator created during installation.
• Ability to enable HTTP and HTTPS delivery separately for each on-demand mount point on Helix Server.

H.264 at High Bit Rates
Helix Server and Helix Proxy now support the streaming of H.264 in the MPEG-4 or 3GPP Release 6 format at speeds up to 3.7 Megabits per second (300 Mbps video maximum with 700 Kbps audio maximum). Formerly, the limit was 384 Kbps.

Documentation Changes
The Helix Server and Helix Proxy documentation consists of this *Helix Media Delivery Platform Integration Guide* in PDF format and the Helix Administrator online help system.

**Note:** Information formerly available in separate configuration and troubleshooting guides is now available in either *Helix Media Delivery Platform Integration Guide* or the Helix Administrator online help system.

Compatibility with Previous Versions
Because of new features, enhancements to protocol support, and changes to mount points, RealNetworks recommends using the new configuration file installed with Helix Universal Server Version 14.3 or Helix Proxy. If you are upgrading from a previous version, import changes to existing features (such as customized reports) into the new configuration file by hand.

**Tip:** If you are upgrading an older version of the server or proxy in an existing location, delete the HelixAdministrator/Docs directory within the main installation directory. See also “Reinstalling the Server or Proxy” on page 36.

**Note:** RealNetworks recommends using identical versions of server products for server-to-server features like splitting.
Chapter 2: FIREWALLS

Firewalls may present communications problems for Helix Server and Helix Proxy. This chapter provides background on firewalls and port configurations. It recommends ways to work with firewalls to give viewers the best possible streaming media experience.

Tip: After installing Helix Server or Helix Proxy, you can find additional information about ports by navigating to Server Setup > Ports or Proxy Setup > Ports in Helix Administrator and clicking the Help link.

Media Server Placement

If you are streaming content only to media clients inside your organization, place media servers and proxies inside your firewall. This requires no special configuration other than to provide virtual private network (VPN) access to any clients outside the firewall.

To stream content to clients on the Internet, it’s better not to locate Helix Server behind a firewall. For optimal streaming, Helix Server needs to use streaming protocols and to process incoming and outgoing UDP connections on a variety of ports. Although you may be able to change your organization’s security policy to enable optimal communication, this may hamper the effectiveness of the firewall.

The best solution may be to place Helix Server or Helix Proxy in a perimeter network, sometimes known as a De-Militarized Zone (DMZ). In this scenario, you fortify the connection between main and perimeter networks but allow a less stringent security policy in the perimeter. This keeps the main network secure while maintaining optimal connections between the Internet and media servers.
Host Addresses for NAT Firewalls

If Helix Server resides on a local network and its IP address is not directly accessible from the Internet, media links that include client launch utilities may fail. This occurs because Helix Server instructs media players to contact it on its IP address, which is accessible only from within the local network. This situation can arise for numerous reasons:

- Helix Server sits behind a Network Address Translation (NAT) firewall.
- The network address exposed to Internet clients is a virtual IP (VIP).
- Media requests are routed to different servers using a load-balancing mechanism.

In these cases, add a HostName address to client launch utility configurations. This provides the media client with the address of the firewall or VIP, which is configured to route the request to the appropriate Helix Server.

For More Information: For information about client launch utilities, refer to the online help topic Streaming Basics > Predefined Mount Points > Client Mount Points. See the topic Configuration File > Basic Setup Configuration > File System Configuration > Client Launch Utility Configuration for examples of how to add a HostName value to the configuration file.

Virtual IP Addresses

If you need to place a cluster of Helix Servers behind a virtual IP address, the network topology may have unintended consequences for RTSP traffic behind a restrictive firewall.

How Virtual IP Addressing Works

A typical IP address resolves to a single server. A virtual IP address resolves to a cluster of servers, typically through a hardware switch. Consider the case of a cluster of servers behind a hardware switch, where all servers share the same content but are configured with private IP addresses. In this scenario, only the hardware switch is assigned a public IP address. The switch receives all public communication, passing each request to a host in the cluster.
The Problem with Virtual Addressing

A problem arises from the combination of public and private IP addresses. If a firewall blocks streaming media protocols, the client communicates through HTTP cloaking. In most cases, this effectively bypasses firewall security, which typically allows HTTP traffic to pass. For cloaking to work, the client must be able to make two HTTP connections to the same Helix Server.

When a client uses HTTP cloaking, Helix Server replies to the initial HTTP connection with its actual IP address, not the virtual IP of the cluster. This allows the client to circumvent the hardware switch and establish its second HTTP connection directly with the Helix Server handling the request. But if that server uses a private IP address, the client cannot make the second, necessary connection, and HTTP cloaking fails.

For More Information: See “HTTP Cloaking for RTSP and RTMP” on page 17.

Resolving the Problem

There are two ways to resolve the virtual addressing problem:

- Configure firewalls to allow connections by streaming media protocols. This is the ideal solution because communication between the server and clients will be the most efficient.

- Use globally routable IP addresses on all hosts behind a virtual IP address. This way clients can make HTTP-cloaked requests behind firewalls that restrict streaming media protocols. This requires your organization to register for a larger number of public IP addresses, however.

Note: If neither of these solutions is possible, some RTSP clients residing outside of highly restrictive firewalls may not be able to access content.

RTSP Communication

Helix Server and Helix Proxy use two connections to communicate with RTSP-based media clients, a control channel and a data channel. For data transport, RTSP clients typically prefer UDP, which may be blocked by restrictive firewalls.
Communication with an RTSP Media Player

RTSP Control Channel

Using the control channel, the server can request and receive passwords. Media Players use the control channel to send instructions such as pause or stop. Most players can work around control channel failures that arise when a firewall blocks the preferred protocol. The player typically shifts to TCP, which is less likely to be blocked than UDP.

Note: When a firewall exists between a media client and the media server, the IP address that appears in the access log’s IP address field may be the firewall address rather than the true client address.

For More Information: The section “Streaming Protocol Ports” on page 32 lists the default ports used by media players to contact the server and set up a control channel.

UDP Data Channel

Once the RTSP control connection is established, the media player negotiates the data channel. Optimally, the data channel will use the UDP transport. If this fails, the player uses the established control channel for data. By default, servers use the following ports for UDP data transfer:

<table>
<thead>
<tr>
<th>Server</th>
<th>Port Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Server</td>
<td>ports in the range 6970 to 32000</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>ports in the range 6970 to 65535</td>
</tr>
</tbody>
</table>

After you install a server, you can modify the UDP port range by adding MinUDPPort and MaxUDPPort port values to the configuration file. For example:

```xml
<Var MinUDPPort="7240"/>
<Var MaxUDPPort="24100"/>
```
For More Information: Refer to the Helix Administrator online help topic Configuration File > Basic Setup Configuration > Protocols, Ports, and Password Configuration.

Tip: Helix Server and Helix Proxy can use a limited range of UDP ports to receive media player requests for lost packet resends. Specify this range in the Helix Administrator Ports page (Server Setup > Ports or Proxy Setup > Ports).

HTTP Cloaking for RTSP and RTMP

Some firewalls restrict streaming media protocols like RTSP or RTMP, preventing a media player from establishing the control connection. In these cases, Helix Server and the player circumvent the problem by disguising streaming media traffic as HTTP, a solution known as HTTP cloaking.

Note: HTTP is not designed for streaming long media files. As a result, HTTP cloaking of RTSP or RTMP streams may lower playback quality.

GET and POST Methods

The HTTP cloaking method must work around limitations in the HTTP protocol. For example, media players use two HTTP streams to connect to Helix Server. Because the player initiates both streams, the client firewall typically allows these connections as outgoing HTTP traffic.

The first HTTP connection uses the GET method, the standard means for a browser to request a Web page. At the receiving end, Helix Server strips off the HTTP disguise, using the encapsulated RTSP or RTMP information to determine what information to send the player.

Helix Server must then wait for the second HTTP connection from the same player to proceed with streaming the media. This second connection uses the HTTP POST method, the standard means for a Web server to send data to a browser.

Once both of these player-initiated streams are established, the media player and Helix Server can pass packets in two directions through a firewall that blocks RTSP or RTMP but allows streaming data that is cloaked as HTTP.
Port 80 For HTTP Traffic

For HTTP cloaking to work, the media player must connect to the Helix Server HTTP port. RealNetworks recommends setting the HTTP port on Helix Server to the standard port 80 during installation to provide the widest support of all media players.

**Warning!** When you install Helix Server and a Web server on the same machine, you need to take precautions before assigning port 80 to Helix Server. For more information, see “Installing Helix Server with a Web Server” on page 275.

Port Hinting on Helix Server

Port hinting offers a solution for a Helix Server that cannot use default port values. It allows Helix Server to send the proper port numbers to certain types of media players when a client launch utility is used. This feature is enabled by default.

**For More Information:** See the Helix Administrator online help topic Helix Administrator > Server Setup > Ports > Communication Through Nonstandard Ports.

Encoders, Receivers, and Proxies

Once you have placed Helix Server in relation to your firewall, you need to consider the placement of other encoders, servers, and proxies. Generally, the same rules and limitations discussed in the preceding sections apply to placing these components as well.

Communicating With Live Media Encoders

If possible, place live media encoders in the perimeter network along with Helix Server. This may not always be possible, however. For example, encoders recording a remote, live event may need to reside on the public side of the firewall.

UDP and TCP Streams

Flash Media encoders typically use TCP for data transport. Helix Producer typically uses UDP for data transport although TCP is available as an option for traversing restrictive firewalls or transmitting data on a lossy network.
Default Server Ports for Live Data

The following are the default Helix Server ports used to receive data from RealNetworks encoders:

<table>
<thead>
<tr>
<th>Port Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Producer in Helix Push mode</td>
<td>50001 to 50050</td>
</tr>
<tr>
<td>Helix Producer in Advanced Push or Multicast mode</td>
<td>30001 to 30020</td>
</tr>
<tr>
<td>Flash Media encoders</td>
<td>1935</td>
</tr>
<tr>
<td>Windows Media encoders in push mode</td>
<td>7077</td>
</tr>
</tbody>
</table>

For More Information: Chapter 10 explains the Helix broadcast methods. Third-party encoders may transmit on a different set of ports. Refer to the specific encoder documentation for details.

HTTP Broadcasts from Windows Media Encoder

When Windows Media Encoder delivers a live Windows Media stream, the only protocol option is HTTP over TCP. With pull broadcasting, Helix Server pulls the live feed from a specific encoder port and receives the stream on its default HTTP port.

Note: With push broadcasting, Windows Media Encoder delivers the HTTP stream to a predefined port on Helix Server. Refer to the Helix Administrator online help topic Helix Administrator > Broadcasting > Windows Media Encoding > Setting up a Windows Media Push Broadcast.

Communicating With Receivers

By default, Helix Servers functioning as transmitters and receivers communicate using UDP. An option is available for them to use TCP instead. The data ports used by and transmitters and receivers are fully configurable. The default range is 30001 to 30020.

For More Information: Chapter 12 explains splitting.

Communicating With Helix Proxies

Helix Proxies commonly work behind a firewall. In this respect, a proxy-to-server connection behaves like a client-to-server connection. Helix Proxy first
tries to connect to Helix Server with RTSP using UDP for data transport. If the firewall prohibits UDP connections, Helix Proxy tries TCP.

**Note:** Helix Proxy has no option for HTTP delivery. If a firewall prohibits RTSP between the origin Helix Server and Helix Proxy, the proxy will not be able to cache or split streams for media clients.
INSTALLATION

This section explains how to install and start Helix Server and Helix Proxy. It also covers Helix Administrator, the graphical tool you use to set up each server.
Chapter 3: INSTALLATION

This chapter explains how to install Helix Server or Helix Proxy on a dedicated machine.

**For More Information:** Refer to Appendix A for information about installing both Helix Server and Helix Proxy on the same machine or on a machine that also includes a Web server.

**Operating System Modifications**

The following sections provide information specific to various operating systems running Helix Server or Helix Proxy. This information will help you to modify the operating system to optimize server performance. RealNetworks recommends that you review this information and make any necessary changes before installing the server or proxy.

**Hyperthreading**

*Hyperthreading* is a feature of many Intel x86 processors, including Xeon processors, as well as Pentium 4 and later processors. It does not provide benefits when running Helix Server and Helix Proxy, however. RealNetworks recommends that you disable hyperthreading. This is typically done through the computer BIOS.

**Tip:** Hyperthreading is different from *multicore processing*. Helix Server and Helix Proxy take advantage of multicore processors on Intel-based and Sun-based hardware. The use of multicore processors significantly improves system performance.

**File Descriptors for Linux and Solaris**

RealNetworks recommends increasing the default file descriptor limits for Solaris and Linux operating systems. Helix Server and Helix Proxy heavily use file descriptors for many operations, such as file reads and socket connections.
The recommended, minimum number of file descriptors to set is 32767 for each CPU, as summarized in the following table.

### Recommended Minimum File Descriptors

<table>
<thead>
<tr>
<th>Processor or Core Count</th>
<th>Minimum File Descriptors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32767</td>
</tr>
<tr>
<td>2</td>
<td>65537</td>
</tr>
<tr>
<td>4 to 7</td>
<td>131074</td>
</tr>
<tr>
<td>8 to 32</td>
<td>262148</td>
</tr>
</tbody>
</table>

**Setting File Descriptors for Red Hat Enterprise Linux**

To set file descriptors for Red Hat Enterprise Linux, open a command-line prompt and type the following to display the system file descriptor limit:

```bash
$ cat /proc/sys/fs/file-max
```

Follow the next procedure if the configured number of file descriptors does not meet or exceed the recommended minimum.

➤ To change the number of file descriptors on RHEL:

1. Log in as the root user.

2. Using any text editor, open the file `/etc/sysctl.conf` and add the following line:
   ```
   fs.file-max = descriptor_number
   ```
   For example:
   ```
   fs.file-max = 65537
   ```

3. Open the file `/etc/security/limits.conf` and add the following lines:
   ```
   * soft nofile descriptor_number
   * hard nofile descriptor_number
   ```
   For example:
   ```
   * soft nofile 65537
   * hard nofile 65537
   ```

4. Open the file `/etc/pam.d/login` and add the following line:
   ```
   session required pam_limits.so
   ```

5. Open the file `/etc/pam.d/sshd` and add the following line:
   ```
   session required pam_limits.so
   ```
6. Save and close all files.

Setting File Descriptors for Solaris 10

To set file descriptors for Solaris 10, open a command-line prompt and type the following to display the system file descriptor limit:

```bash
$ ulimit -Hn
```

If the configured number of file descriptors does not meet or exceed the recommended minimum, log in as `root` and open the file `/etc/system`. Add the following line:

```
rlim_fd_max=descriptor_number
```

For example:

```
rlim_fd_max=131074
```

Setting SLTA Soft File Descriptors for Solaris 10

If you plan to use the SLTA utility to run simulated-live broadcasts on Solaris 10, increase the soft limit for file descriptors to 512 on the machine that hosts the SLTA executable. You can set soft limits in the shell used to run SLTA, or system-wide for all processes.

**For More Information:** For more on SLTA, see Chapter 13.

Setting Soft File Descriptor Limits in a Single Shell

To set soft file descriptor limits for a single SLTA instance, give the following command before starting SLTA, or add the command to the shell login script:

```bash
$ ulimit -Sn 512
```

Setting Soft File Descriptor Limits for All Shells

To raise the number of soft file descriptors for all shells, log in as `root` and open the file `/etc/system`. Add the following line:

```
rlim_fd_max=131074
```

After you save and close the file, reboot the operating system.
Extended FILE Support on Solaris 10 for iOS Devices

To support Apple iOS devices as described in Chapter 7, Solaris 10 operating systems must support the extended FILE facility to allow Helix Server to access more than 256 file descriptors in the fopen() I/O call. Providing this support requires patching Solaris 10 versions 3/05 through 11/06.

**For More Information:** Refer to the documentation at http://developers.sun.com/solaris/articles/stdio_256.html

Solaris 10 Update 6 for SNMP

If you plan to use the SNMP feature described in Chapter 6, apply update 6 or higher to the Solaris 10 operating system to ensure that the master agent functions correctly.

Windows 2008 Account Privileges

The account that runs the Helix Server or Helix Proxy installation program on Windows 2008 Server must have administrator privileges. The account that runs Helix Server or Helix Proxy does not require administrator privileges, however.

**Note:** After installation, you can modify the configuration file access privileges to allow users other than the Administrator account to run Helix Server or Helix Proxy. See “Modifying Access Privileges” on page 40.

Windows Default Buffer Size

When running Helix Server or Helix Proxy on a supported Windows operating system, increase the default send buffer size in the Windows Registry. This change prevents poor quality of service for media players connecting to live broadcasts over TCP.

➤ To increase the default send buffer size on Windows:

1. Choose **Start > Run**.

2. Open the Windows Registry Editor by entering the following command: regedit

3. Navigate to the following registry key:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters
4. To the key called DefaultSendWindow, add a new DWORD value.

5. Set the DWORD value to 32767 (decimal).


Installing Helix Server

To install Helix Server, you need a binary installation file and a license file that enables the Helix Server features. License files are delivered by e-mail after you receive your Helix Server installer and generate your machine ID as described in Step 4 below.

Tip: If you are upgrading from an earlier version, log into your PAM account at https://pam.realnetworks.com and see knowledge base article 905 for tips about preparing an existing installation for upgrade.

➤ To install Helix Server:

1. Log into the account that will own the Helix Server process.

   For More Information: If you are installing Helix Server on a Windows operating system, see “Windows 2008 Account Privileges” on page 26.

2. Launch the setup file. On Windows, double-click the file. On UNIX or Linux, ensure that the program has executable permission. Then open a command line and run the installer:

   ./program_name.exe

3. Read the installation recommendations and click Next.

4. Click Browse to navigate to the license key file you received from RealNetworks. Then click Next.

   Tip: The installation process copies the specified license file to the License subdirectory under the main Helix Server directory. On startup, Helix Server reads this local copy of the license. The original license file is no longer required but should be stored securely in case reinstallation is required.

   For More Information: See “Obtaining a License Key” on page 31 if you do not yet have a license key.
5. Read the end-user license agreement, signifying your agreement to its terms by clicking **Accept**.

6. Enter or browse to the path where you want to install Helix Server. On Windows, you may choose to accept the default path in the **C:\Program Files** directory.

   **Note:** Examples in this guide assume that you’ve chosen the default path.

7. Enter a user name and password, which are required to access various Helix Server features, such as Helix Administrator. Confirm your case-sensitive password by entering it again.

   In the bottom two fields, the installer generates random values for the ports used to access Helix Administrator. Accept the values or choose new values.

   **For More Information:** See “Administration Ports” on page 32.

8. In the next set of screens, you select the default ports that Helix Server uses to listen for requests for specific streaming protocols (RTSP, RTMP, HTTP, HTTPS).

   **For More Information:** See “Streaming Protocol Ports” on page 32.

9. On the certificate files screen, you enter location information used to generate a request file sent to a certificate signing authority. You can leave these fields blank only if you do not plan to use the secure HTTP (HTTPS) features of Helix Server.

   **For More Information:** The section “Creating Certificate Files” on page 34 explains the information fields and the certification process.

10. Select port values used for HTTPS requests to specific Helix Server features. To use unsecured HTTP for these ports, uncheck the **Enable Control Port Security** box.

    **For More Information:** See “Feature Control Ports” on page 33.
11. On Windows, the default installation sets up Helix Server as a service. This is recommended, but you can prevent this by unchecking the **Install Helix Server as an NT service** box.

This installer page also presents the option to **Install SNMP Master Agent as an NT service**. If you check this box, the Simple Network Monitoring Protocol master agent is installed as a service.

**Tip:** You can later set up Helix Server to run as a service as described in the section “Windows Service Options” on page 51.

**For More Information:** The SNMP option is significant only if you have licensed the SNMP feature, which Chapter 6 explains.

12. In the final confirmation screen, review the installation information. Click **Install** to complete the installation process.

### Installing Helix Proxy

To install Helix Proxy, you need a binary installation file and a license file that enables the Helix Proxy features. License files are delivered by e-mail after you receive your Helix Proxy installer and generate your machine ID as described in Step 4 below.

**To install Helix Proxy:**

1. Log into the account that will own the Helix Proxy process.

   **For More Information:** If you are installing Helix Proxy on a Windows operating system, see “Windows 2008 Account Privileges” on page 26.

2. Launch the setup file. On Windows, double-click the file. On UNIX or Linux, ensure that the program has executable permission. Then open a command line and run the installer:

   ```bash
   ./program_name.exe
   ```

3. Read the installation recommendations and click **Next**.

4. Click **Browse** to navigate to the license key file you received from RealNetworks. Then click **Next**.
**Tip:** The installation process copies the specified license file to the License subdirectory under the main Helix Proxy directory. On startup, Helix Proxy reads this local copy of the license. The original license file is no longer required, but should be stored securely in case reinstallation is required.

**For More Information:** See “Obtaining a License Key” on page 31 if you do not yet have a license key.

5. Read the end-user license agreement, signifying your agreement to its terms and conditions by clicking **Accept**.

6. Enter a path where you want to install Helix Proxy. On Windows, you may choose to accept the default path, or specify a different path.

   **Note:** Examples in this guide assume that you’ve chosen the default path.

7. Enter a user name and password required to access Helix Administrator. Confirm your case-sensitive password by entering it again. In the bottom fields, the installer generates values for the ports used to access Helix Administrator. Accept the values or choose new values.

   **For More Information:** See “Administration Ports” on page 32.

8. In the next screen, you define the port that Helix Proxy uses for the RTSP protocol. RealNetworks recommends accepting the default port unless that value will cause conflicts with other applications.

   **For More Information:** See “Streaming Protocol Ports” on page 32. If installing on UNIX or Linux, see also “Starting as root in UNIX and Linux” on page 33.

9. On the certificate files screen, you enter location information used to generate a request file sent to a certificate signing authority. You can leave these fields blank only if you do not plan to use the secure HTTP (HTTPS) features of Helix Proxy.

   **For More Information:** The section “Creating Certificate Files” on page 34 explains the information fields and the certification process.
10. On Windows, the default installation sets up Helix Proxy as a service. This is recommended, but you can prevent this by clearing the **Install as NT Service** box.

This installer page also presents the option to **Install SNMP Master Agent as an NT service**. If you check this box, the Simple Network Monitoring Protocol master agent is installed as a service.

**Tip:** You can later set up Helix Proxy to run as a service as described in the section “Windows Service Options” on page 51.

**For More Information:** The SNMP option is significant only if you have licensed the SNMP feature, which Chapter 6 explains.

11. In the final confirmation screen, review the installation information. Click **Install** to complete the installation process.

### Obtaining a License Key

If you have not received your license file from RealNetworks, copy the 32-character machine ID shown on the second screen of the installation program. Then follow the instructions in the e-mail sent to you when you downloaded the server or proxy installer. This generates a license file valid for the machine on which the installer is running.

If you did not receive an e-mail message, log into your product and maintenance account (PAM) at the following Web page:

https://pam.realnetworks.com

Navigate to the download section for your server and click on the **Download license key** button.

**For More Information:** If you have problems generating the machine ID or receiving a license, contact Technical Support at http://realnetworks.com/helix/support.aspx.

### Selecting Port Values

The following sections provide guidelines for choosing port values when installing Helix Server. RealNetworks recommends accepting the default ports
unless those values conflict with existing port assignments. You can change the port settings once Helix Server is running.

**For More Information:** See the Helix Administrator online help topic [Helix Administrator > Server Setup > Ports > Port Assignments](#) for instructions on changing port values.

### Administration Ports

Helix Administrator can connect to Helix Server using HTTP or HTTPS. The installation program generates a random port value used for each protocol. RealNetworks recommends that you accept the default values, but you may choose other port values.

**Tip:** Be sure to remember the port numbers or record them in a secure location. The section “Starting Helix Administrator” on page 57 explains how you use one of the port numbers to access the Helix Administrator interface.

### Streaming Protocol Ports

The following are the standard ports used for streaming protocols. RealNetworks recommends using these port values whenever possible. Media clients use these ports by default when requesting content:

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>80</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
</tr>
<tr>
<td>RTSP</td>
<td>554</td>
</tr>
<tr>
<td>MMS</td>
<td>1755</td>
</tr>
<tr>
<td>RTMP</td>
<td>1935</td>
</tr>
</tbody>
</table>

### Nonstandard Values for Protocol Ports

If you choose a different port number for a protocol, you may need to include the port number in URLs used to request media. Once you start Helix Server, you can find more information about nonstandard port use or change any port values selected during installation. Do this by logging into Helix Administrator, navigating to the [Server Setup > Ports](#) page ([Proxy Setup > Ports](#) page on Helix Proxy), and clicking the **Help** link.
Starting as root in UNIX and Linux

On UNIX and Linux, you have to log in as root to start Helix Server if you set port values lower than 1000. You can, however, change the ownership of the Helix Server process to a different user after startup. After you log into Helix Administrator, navigate to the Server Setup > User/Group Name page (Proxy Setup > User/Group Name page on Helix Proxy) to set up this feature.

Feature Control Ports

The following port assignments are used only with specific Helix Server features.

FCS Port (8008)

This port is used with the fast channel switching feature described in Chapter 14. Helix Server listens on this port for HTTP requests to switch a channel stream to a new source.

Tip: If you do not use the channel switching feature, you do not need to define this port value.

SSPL Port (8009)

Helix Server uses this port with the server-side playlist feature described in Chapter 9. It listens on this port for HTTP requests to switch to different parts of an externally controlled playlist. The default value is 8009. Note the following about this port:

• If you do not use the server-side playlist feature, you do not need to define this port value.

• You can set this port to the same value as another HTTP port, such as the main HTTP port (default value of 80), the FCS port (default value of 8008), or the content management port (default value of 8010).

• If you leave the port value undefined, any Helix Server port capable of handling HTTP communication can accept playlist controller requests.

Content Management Port (8010)

This port is also used with the server-side playlist feature. Helix Server listens on this port for HTTP requests to add or delete a playlist. The default value is 8010. Note the following about this port:
• A port value **must** be assigned even if you do not use the playlist management feature.

• You can set this port to the same value as another HTTP port, such as the main HTTP port (default value of 80), the FCS port (default value of 8008), or the playlist control port (default value of 8009).

• If Helix Server runs on a multi-homed machine, RealNetworks recommends that you configure your firewall to block this port on any public IP address that Helix Server uses. The Web portal should then contact Helix Server on this port using a private IP address.

**Control Port Security**

If you check the **Enable Control Port Security** box, all feature ports described in the preceding sections accept only HTTPS connections. If you leave the box unchecked, the ports accept only HTTP connections.

**Transport Layer Protocols**

You do not need to configure transport-layer protocols such as TCP and UDP. Helix Server and the client automatically select the transport protocol.

**Tip:** After installation, you can restrict the UDP port range on the Helix Administrator **Ports** page (**Server Setup > Ports**).

**Creating Certificate Files**

During installation, the Helix Server or Helix Proxy installer generates 2048-bit security keys and a certificate signing request (a .csr file). A certificate signing authority uses this request file to validate the identity of your server for secure HTTP (HTTPS) transactions.

**Tip:** After you install Helix Server or Helix Proxy, you can modify HTTPS settings by, for example, turning off support for certain versions of SSL. In the Helix Administrator online help, refer to the topic **Configuration File > Basic Setup Configuration > Security Key Configuration.**
Specifying Certificate Information

The Helix Server or Helix Proxy installation program asks for the following information needed to generate a certificate request:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country Code</td>
<td>Two-letter ISO code used to identify the country in which the server resides. For the United States, for example, the code is US.</td>
</tr>
<tr>
<td>State/Province</td>
<td>Your state or province spelled out fully, as in Washington.</td>
</tr>
<tr>
<td>Locality</td>
<td>The full name of your city, as in Seattle.</td>
</tr>
<tr>
<td>Organization</td>
<td>The official business name of your organization, as in RealNetworks, Inc. Do not include special characters such as &amp; or @ in the organization name. Spell out these symbols or omit them.</td>
</tr>
<tr>
<td>Org. Unit</td>
<td>Organizational unit of your company, as in Technology and Products. This information is optional.</td>
</tr>
<tr>
<td>Common Name</td>
<td>The name of the server, as in helixserver.example.com. Include the host name (for example, helixserver) along with the domain name (for instance, example.com). If you register only the domain name, browsers contacting the specific host will warn of a certificate mismatch. The signing authority may accept all host names if you enter a wild card, as in *.example.com. Verify this with your signing authority first.</td>
</tr>
<tr>
<td>Contact E-mail</td>
<td>Your e-mail address, as in <a href="mailto:jsmith@real.com">jsmith@real.com</a>.</td>
</tr>
<tr>
<td>CSR Name</td>
<td>Name of the authority that will sign the certificate, as in Verisign.</td>
</tr>
</tbody>
</table>

**Warning!** Leaving non-optional fields blank invalidates the certificate request.

**For More Information:** You can find a list of ISO country codes at [http://www.iso.org/iso/country_codes/](http://www.iso.org/iso/country_codes/).

Requesting a Signed Certificate

The installation process generates the certificate signing request file, key.csr, in the Certificates subdirectory of the main installation directory. You send this file to the certificate signing authority to receive an officially signed certificate.

**For More Information:** You can learn more about the signing process at the Web site of any signing authority, such as [http://www.verisign.com/](http://www.verisign.com/).
Note: RealNetworks does not endorse the use of any specific signing authority.

Using Self-Signed Certificates

Until you receive a response from a signing authority, your certificates are self-signed. These certificates will function with HTTPS requests. Note the following about using self-signed certificates, however:

- Browsers warn the user that the server is untrusted. Users may then choose to add an exception for the server.
- HTTPS requests for content segmented for iOS devices (see Chapter 7) does not work with self-signed certificates.

Running the OpenSSL Program

The Helix Server or Helix Proxy generates .pem and .key files using the OpenSSL program provided by the OpenSSL Project. The executable file, openssl.exe, is included in the Bin subdirectory of the main installation directory. This program has many uses, including regenerating the .csr and .pem files without running the installation program.

Tip: On Windows, the install.log file in the main installation directory records the commands used with the openssl.exe program to generate the .csr and .pem files. On Linux and UNIX, this information is written to stdout and not saved unless you redirect the installer output to a file.


Reinstalling the Server or Proxy

Reinstallation of the same version of the server or proxy is generally not necessary. If you must perform a reinstallation, the process does not affect content under the server mount points or in the proxy cache. However, it does reset your system configuration.

Note the following tips to make your reinstallation process smoother:
• Back up the server or proxy configuration file (rmserver.cfg or rmproxy.cfg). After reinstalling the product, either replace the files created by the installer with your backups or migrate any necessary changes to the new files by hand.

• If using SNMP, back up the SNMP configuration file (master.cfg).

• Back up the slta.cfg file if you used that file for running SLTA.

  **For More Information:** For more information about SLTA, see Chapter 13.

• Back up any authentication databases (adm_b_db, con_r_db, and so on) that you’ve revised or added. This step is necessary only if you’ve added more users and passwords for authentication than those added during installation.

  **For More Information:** The Helix Administrator online help topic **Helix Administrator > Security > User Databases** explains authentication databases.

• Certificate files in the Certificates subdirectory are **not** overwritten on a reinstall. If you wish to generate new files, remove the .csr and .pem files from this directory.

• If you bookmarked Helix Administrator in your browser, specify the same Helix Administrator port during the reinstall to keep the bookmark functional.

• A reinstall does not affect cache files, access logs, error logs, or HTTPS certification keys. It is therefore not necessary to back up these files before reinstall. These files typically reside in the Cache, Logs, and Certificates subdirectories of the main installation directory.
Chapter 4: STARTUP AND SHUTDOWN

This chapter explains how to start up and shut down Helix Server and Helix Proxy. It covers startup options and information about memory use.

Starting the Server or Proxy on Windows

The following sections explain how to start Helix Server or Helix Proxy as a Windows service, from the Start menu, or from the Windows command line.

Running a Windows Service

In its default Windows installation, Helix Server or Helix Proxy is set up as a service under the product name. The application always runs in the background, and you do not need to start it. You may, however, want to add startup parameters to change the maximum allowable memory, for example.

For More Information: The section “Command Options” on page 45 explains all of the possible startup options. For more about memory usage, see “Memory Allocation” on page 47

Modifying Startup Parameters

The following procedure explains how to edit the Windows registry to add start-up options to a server or proxy running as a Windows service.

Note: Do not add start-up parameters to the Start parameters field of the service’s Properties dialog.

➤ To add options to a Windows service:

1. Choose Start > Settings > Control Panel.
2. Double-click Administrative Tools.
3. Launch Services.
4. Right-click the service name, such as Helix Server or Helix Proxy, and choose Stop.

5. Choose Start > Run to open the Windows Registry Editor.

6. Enter the following command:
   regedit

7. Navigate to the following registry key, in which ServiceName is the name of the service, such as Helix Server:
   HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\ServiceName

8. Click the key name, then click the StartupParams entry in the right-hand pane.

9. Choose Modify.

10. Add the command option to the end of the Value data field. For example:
    "C:\Program Files\Real\Helix Server\rmserver.cfg" -m 1024

11. Click OK.

12. Return to the Services list, highlight the service name, right-click, and choose Start.

**Modifying Access Privileges**

You will need to change the access privileges on the Helix Server or Helix Proxy configuration file if the following are true:

- You plan to run Helix Server or Helix Proxy as an application rather than a service.

- You want to run Helix Server or Helix Proxy under an account other than Administrator. This includes all users in the administrator group other than the actual Administrator account.

To change configuration file access privileges:

1. Log into Windows Server with the Administrator account.

2. Open Windows Explorer and browse to the main Helix Server or Helix Proxy installation directory.

3. Right-click the configuration file (rmserver.cfg or rmproxy.cfg) and choose Properties.
4. On the **Security** tab, click **Edit** and mark the file for full control by the **Users** group or another group as appropriate for your organization’s security policies.

5. Click **OK** to exit the dialog.

**Using the Start Menu or a Desktop Icon**

To start up from the Windows desktop, double-click the server or proxy icon added to the desktop by the installation program. Or, from the **Start** menu, select the **Helix** server or proxy entry.

**Modifying Startup Parameters**

By default, the server or proxy uses the configuration file created by the installation program (**rmserver.cfg** or **rmproxy.cfg**). You can add additional startup options described in the section “Command Options” on page 45.

> **To change the startup options in an icon:**

1. Stop the server if it is running.

   **For More Information:** See “Stopping the Server or Proxy” on page 44.

2. Right-click the application icon on the desktop or in the **Start** menu.

3. Select **Properties**.

4. In the **Target** field, change the name of the configuration file or add the command options to the end of the command line syntax. For example:

   "C:\Program Files\Real\Helix Server\Bin\rmserver.exe" "C:\Program Files\Real\Helix Server\rmserver.cfg" -m 1024

   **For More Information:** For more about memory usage, see “Memory Allocation” on page 47.

5. Save the properties and close the dialog.

6. Repeat this procedure for each shortcut icon that you use to start the server or proxy.

7. Double-click the icon to restart the server.
Starting Up from the Windows Command Line

From the Start menu, open the command prompt. Navigate to the main installation directory. Enter the following command to start Helix Server with its default configuration file and memory use:

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

The following command starts Helix Proxy:

```
Bin\rmproxy rmproxy.cfg
```

**Tip:** If configuration file variables contain relative paths, you must start the server or proxy from its main installation directory as shown here to ensure that the paths resolve.

Modifying Startup Parameters

Optionally, you can include any startup options described in the section “Command Options” on page 45. On the command line, you list options after the executable name and the configuration file name, preceding each option with one or two hyphens. For example:

```
Bin/rmserver rmserver.cfg -option1 --option2 ...
```

A common practice is to change the server or proxy memory allotment by including the \-m\ parameter. For example, the following allows Helix Server to use up to 1 Gigabyte of memory:

```
Bin\rmserver rmserver.cfg -m 1024
```

**For More Information:** For more about memory usage, see “Memory Allocation” on page 47.

Starting the Server or Proxy on UNIX or Linux

If you performed a default installation of Helix Server or Helix Proxy, the RTSP port is set lower than 1024, requiring the user who starts the server or proxy to log in as root.

**Tip:** If you do not want the server or proxy to inherit root privileges, you can switch it to another user and group name immediately after it starts up changing the settings in the User/Group Name page in the Setup section of Helix Administrator.
To start Helix Server or Helix Proxy on UNIX:

1. Start any command shell.
2. Navigate to the main Helix Server or Helix Proxy installation directory.
3. Choose one of the following options:
   a. Start the server or proxy as a background process. Use the following command for Helix Server:
      
      Bin/rmserver rmserver.cfg &
      
      Use the following command with Helix Proxy:
      
      Bin/rmproxy rmproxy.cfg &
   b. Start the server or proxy as a standard application. Use the following command for Helix Server:
      
      Bin/rmserver rmserver.cfg
      
      Use the following command with Helix Proxy:
      
      Bin/rmproxy rmproxy.cfg

   **Tip:** If configuration file variables contain relative paths, you must start the server or proxy from its main installation directory as shown here to ensure that the paths resolve.

**Modifying Startup Parameters**

Optionally, you can include any startup parameter described in the section “Command Options” on page 45. On the command line, you list options after the executable name and the configuration file name, preceding each option with one or two hyphens. For example:

Bin/rmserver rmserver.cfg -option1 --option2 ... &

A common practice is to change the amount of memory that the server or proxy can use. Do this by including the `-m` parameter and the amount of memory in Megabytes. For example, the following allows Helix Server to use up to 4 Gigabytes of memory:

Bin/rmserver rmserver.cfg -m 4096 &

**For More Information:** For more about memory usage, see “Memory Allocation” on page 47.
Stopping the Sever or Proxy

It’s generally not necessary to stop Helix Server or Helix Proxy when it’s running. You can make configuration changes while the application runs. If changes require a restart, you can restart the server or proxy using Helix Administrator as described in “Restarting the Server or Proxy” on page 59.

Tip: By implementing delayed shutdown on the server or proxy, you can allow media players to report playback statistics before the shutdown commences. In Helix Administrator, navigate to the Delayed Shutdown page of the Setup section.

Shutting Down on Windows

If Helix Server or Helix Proxy was started as a Windows service, stop it through the Services control panel. Give the Start > Settings > Control Panel command. Then double-click Administrative Tools and Services. Locate the service name on the list (such as Helix Server or Helix Proxy), highlight it, and click Stop.

If you started Helix Server or Helix Proxy manually, switch to the command window and press Ctrl+c. You can also use the Task Manager (Ctrl+Alt+Del) to stop the application.

Shutting Down on UNIX or Linux

To stop Helix Server or Helix Proxy on UNIX, obtain the parent process identification number, and then issue the kill command with that process number.

PID File Location

A text file records the current value of the process ID of the parent process (rmserver or rmproxy) The file, named rmserver.pid or rmproxy.pid, is stored in the directory indicated by the PidPath variable.

Tip: If PidPath is omitted from the configuration file, the server or proxy stores the information in the directory specified by the LogPath variable.

Kill Command Using the PID File

From the command line, navigate to the directory that contains the PID file, and type the following, where pidfile is the name of the PID file:
CHAPTER 4: Startup and Shutdown

kill ‘cat pidfile’

Command Options

On all supported platforms, you can add various options to the Helix Server or Helix Proxy start command. The following table summarizes the common startup options. Most options have both short names and long names.

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Long Name</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>--help</td>
<td>Display help and exit.</td>
<td>page 50</td>
</tr>
<tr>
<td>--hbi IP</td>
<td>--heart-beat-ip IP</td>
<td>Check for server heartbeat on the specified IP address.</td>
<td>page 49</td>
</tr>
<tr>
<td>-m MB</td>
<td>--memory MB</td>
<td>Set maximum memory usage in Megabytes.</td>
<td>page 47</td>
</tr>
<tr>
<td>--out file</td>
<td>--output-file file</td>
<td>Redirect console output to the specified file.</td>
<td>page 49</td>
</tr>
<tr>
<td>--rss n</td>
<td>--report-server-stats n</td>
<td>Report server statistics every n seconds. The default is 60.</td>
<td>page 49</td>
</tr>
<tr>
<td>-v</td>
<td>--version</td>
<td>Print version information.</td>
<td>page 50</td>
</tr>
<tr>
<td>-V</td>
<td>--Version</td>
<td>Print verbose version information.</td>
<td>page 50</td>
</tr>
</tbody>
</table>

**Note:** Always add command options *after* the configuration file option on the command line.

**For More Information:** For background on server statistics, refer to the RSS Statistics section of the online help.
Debugging Options

The following table summarizes the debugging options for Windows, UNIX, and Linux. These options should be used only under the direction of RealNetworks support personnel.

<table>
<thead>
<tr>
<th>Short Name</th>
<th>Long Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>--acd</td>
<td>--allow-core-dump</td>
<td>Allow the server to create core dump files.</td>
</tr>
<tr>
<td>--cp</td>
<td>--crash-avoidance-print</td>
<td>Print information about crash avoidances. Refer to the online help topic RSS Statistics &gt; RSS Network Statistics for information about crash avoidance events.</td>
</tr>
<tr>
<td>--debug</td>
<td>--debug</td>
<td>Combination of --acd, --cp, --mca, -rss, and --sdm.</td>
</tr>
<tr>
<td>--dhb</td>
<td>--disable-heart-beat</td>
<td>Disable the heart-beat check.</td>
</tr>
<tr>
<td>--mca n</td>
<td>--max-crash-avoidance n</td>
<td>Sets n as the number of crash avoidances that can occur before the server automatically restarts.</td>
</tr>
<tr>
<td>--nar</td>
<td>--no-auto-restart</td>
<td>Do not restart after a fatal crash.</td>
</tr>
<tr>
<td>--nca</td>
<td>--no-crash-avoidance</td>
<td>Disable the crash avoidance feature.</td>
</tr>
<tr>
<td>--sct</td>
<td>--skip-cpu-test</td>
<td>Do not use automatic CPU test.</td>
</tr>
<tr>
<td>--sdm</td>
<td>--show-debug-messages</td>
<td>Show additional runtime debug messages.</td>
</tr>
</tbody>
</table>

Troubleshooting Command Options

If you experience problems with Helix Server or Helix Proxy after using various startup options, check the following:

- Check for missing option values. For example, the -m and -rss options require values that set the memory amount and statistics reporting period, respectively.

- Some command options, especially debugging options, depend on other command options. Ensure that you have provided all required options.

- Check for bad parsing of command line parameters. In some cases, all that may be required for Helix Server or Helix Proxy to run correctly is to rearrange the order of the options.
Memory Allocation

By default, the 32-bit or 64-bit version of Helix Server or Helix Proxy starts up with the following memory allocation maximum:

- up to 80 percent of physical RAM with an absolute limit of 1.6 GB.

For example, on a machine with 1 GB of physical RAM, Helix Server or Helix Proxy may request up to 800 MB of memory. On a machine with 2 GB or 4 GB of RAM, it will request up to 1.6 GB of memory as needed.

Changing the Default Memory Allocation

Using the `-m` option in the startup command, you can set a different maximum memory in Megabytes for Helix Server or Helix Proxy. For example, on 64-bit systems with a large amount of RAM, you may want to set a higher maximum such as 4 Gigabytes:

```
Bin\rmserver.exe rmserver.cfg -m 4096
```

On 32-bit systems, you may want to set a lower maximum value, such as the 512 Megabytes shown here:

```
Bin\rmserver.exe rmserver.cfg -m 512
```

**Note:** You cannot set the memory allocation for 32-bit versions of Helix Server or Helix Proxy lower than 256 MB. For 64-bit versions, the minimum value for the `-m` option is 512 MB.

Memory-Mapped I/O

Helix Server and Helix Proxy also use memory-mapped I/O for each on-demand stream. This memory allocation is not counted in application memory set by the `-m` option value. The amount of memory needed for I/O depends on the number of streams and the stream bit rate. High-bandwidth clips require more memory-mapped I/O for each stream.

RealNetworks recommends having 30 percent of the value of the maximum application memory available for I/O. For example, if you limit application memory to 1 GB using the `-m` option, you should have at least 300 MB of additional memory available for memory-mapped I/O.

**Tip:** The memory statistics reported by RSS indicate the amount of memory used for memory-mapped I/O. If you run out of available system memory for these operations, you may
need to set the value of the -m option lower. For details, see the RSS Statistics topic in the Helix Administrator online help.

32-Bit Application Memory Maximum

With 32-bit applications and operating systems, application memory is limited by the absolute addressable limit of 4 GB. However, you must also take into account the memory requirements of the operating system kernel and any additional third-party applications.

The following table lists 32-bit operating systems along with the maximum amount of system memory that Helix Server or Helix Proxy can use. The value in the last column is the maximum, recommended setting for the -m option. This setting allows the server to consume as much memory as possible while reserving adequate memory for memory-mapped I/O.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Application and Memory-Mapped I/O Maximum</th>
<th>Maximum Recommended Memory Value (-m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux i386 32-bit</td>
<td>2.8 GB</td>
<td>2155 (2.2 GB)</td>
</tr>
</tbody>
</table>

32-Bit Applications on 64-Bit Operating Systems

If you install a 32-bit Helix Server or Helix Proxy application on a 64-bit operating system, the application is subject to 32-bit memory addressing limitations of 4 GB. For higher memory utilization, use a 64-bit version of Helix Server or Helix Proxy on a 64-bit operating system.

Tip: The -v startup flag indicates if the application is 32-bit or 64-bit. See “Help and Version Options” on page 50.

Memory Use Characteristics

The 32-bit and 64-bit versions of Helix Server and Helix Proxy use memory as needed up to the default maximum or the maximum defined by the -m option. Once the server uses system memory, it does not release that memory back to the operating system.

After you start the server, you typically see application memory as reported by the operating system rise steadily until it reaches the maximum value defined by the -m option. At this point, memory use for Helix Server or Helix Proxy (as
reported by the operating system) levels off but does not decline regardless of the streaming load.

**Tip:** Memory used for memory-mapped I/O is not reported as application memory and may vary with the streaming load.

**For More Information:** RSS statistics report the actual amount of memory in use by the server. See the **RSS Statistics** topic in the Helix Administrator online help for details.

### Heartbeat Check

The heartbeat check is a standalone program installed with Helix Server or Helix Proxy. This program, which starts up along with the server, monitors the localhost address by default, periodically checking that the server is still running. If the heartbeat program determines that the server process has died, it attempts to restart the server.

The **--hbi** option specifies an IP address that the heartbeat program monitors. You need to specify the specific address used by Helix Server or Helix Proxy if the server does not bind to the machine’s localhost address. The following example sets an IPv4 address that the heartbeat program uses to test whether Helix Server is still running:

```
Bin\rmserver rmserver.cfg --hbi 197.168.0.100
```

**Note:** Two debugging options, **--dhb** and **--nar**, disable the heartbeat check and the automatic restart option, respectively. These should be used only under supervision of RealNetworks support personnel.

**For More Information:** You need to specify a heartbeat IP at startup if you install both Helix Server and Helix Proxy on the same machine. Appendix A explains this installation and configuration process.

### Server Statistics and Output Files

The **--rss** option logs RSS statistics to standard output (**stdout**). As well, you can save the RSS output to a file by using a shell redirect or the **--out** option. Using the **--rss** option is typically not necessary because Helix Server and Helix Proxy automatically log RSS information to the **Logs/rsslogs** directory
regardless of whether you use the --rss command option at startup. Using --rss at startup may be useful for debugging, however.

The following example starts Helix Server and writes RSS output to the text file stats.txt every minute:

Bin/rmserver rmserver.cfg --rss 60 --out stats.txt

**Note:** RSS output sent to stdout ignores the settings of the RSS values in the configuration file. No file rolling occurs for screen output redirected to a file.

**For More Information:** Refer to the Helix Administrator online help topic **RSS Statistics** for information about automatic RSS logging, as well as explanations of each line of RSS output.

### Help and Version Options

The -h option prints to the screen the server version information and a summary of the available command options. The version options, -v and -V, print standard and verbose version information, respectively. Add one of these options as the first argument after the executable server name. Do not include the configuration file name. The server prints the information and exits without starting up.

For example, use the following command from the main installation directory to display Helix Server version information and help:

Bin/rmserver -h

The following command displays version information only:

Bin/rmserver -v

The output for the standard version information is the following:

Name (c) Date RealNetworks, Inc. All rights reserved.
Version: Name (RealNetworks) (Full_Version) (Build Build_ID)
Platform: Platform (32-bit|64-bit)
The following table describes the field values for the version information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>Copyright date.</td>
</tr>
<tr>
<td>Name</td>
<td>The product name.</td>
</tr>
<tr>
<td>Full_Version</td>
<td>Full product version. For example: 14.0.2.2347</td>
</tr>
<tr>
<td>Build_ID</td>
<td>The specific ID for the software build. For example: 204091/14732</td>
</tr>
<tr>
<td>Platform</td>
<td>The supported platform.</td>
</tr>
</tbody>
</table>

The following is an example of the version output:

Helix Server (c) 1995-2011 RealNetworks, Inc. All rights reserved.
Version: Helix Server (RealNetworks) (14.2.0.186) (Build 221981/1432)
Platform: sunos-5.10-sparc-server (64-bit)

Windows Service Options

Several command options are used only with Windows operating systems to configure the server to run as a service. If you chose to install Helix Server or Helix Proxy as a service, you typically do not need to use these options because the service starts automatically with the operating system.

If you did not install Helix Server or Helix Proxy as a service, you can set up a service using the command options. Optionally, you can import the server configuration file into a Windows registry key. The service can then read the configuration from that key instead of from the configuration file.

The following table summarizes the options that you can use on Windows when running Helix Server or Helix Proxy as a service.

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-import: key file</td>
<td>Import the specified configuration file into the named Windows registry key.</td>
</tr>
<tr>
<td>-install: ServiceName &quot;parameters&quot;</td>
<td>Install the server as a Windows Service called ServiceName, passing the parameter list &quot;Params&quot; upon startup.</td>
</tr>
<tr>
<td>-N</td>
<td>Run as a service.</td>
</tr>
</tbody>
</table>
Importing a Configuration File into the Registry

Importing the configuration file into the Windows registry is optional. The service can also read the configuration variables from the file at startup. Note that if you change the configuration file, you must re-import the file into the registry and, in some cases, restart the server.

To import the configuration file into a specific key in the Windows registry, open a Windows command prompt from the Start menu. Navigate to the application’s Bin directory, and enter the following:

`executable -import:key file`

This command includes the following elements:

**Registry Import Values**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>executable</code></td>
<td>Name of the Helix Server or Helix Proxy executable program, either <code>rmserver.exe</code> or <code>rmproxy.exe</code>.</td>
</tr>
<tr>
<td><code>key</code></td>
<td>The Windows registry key name to use. If the name contains spaces, enclose the name in double quotation marks. If you omit this, the default name <code>Config</code> is used.</td>
</tr>
<tr>
<td></td>
<td>Note that you can run multiple Helix Proxy service instances on a single machine. To do this, load the configuration file for each instance into a separate registry key.</td>
</tr>
<tr>
<td><code>file</code></td>
<td>The file name of the configuration file to import, including the full or relative path. If the path contains spaces, enclose the entire string in double quotation marks.</td>
</tr>
<tr>
<td></td>
<td>The configuration file itself must provide absolute paths for variables such as <code>BasePath</code>. The server does not recognize relative paths while running as a service.</td>
</tr>
</tbody>
</table>
CHAPTER 4: Startup and Shutdown

Helix Server Configuration Import Example

The following sample command imports all of the Helix Server configuration values held in rmserver.cfg:

```
rmserver.exe -import:"Server 1" "C:\Program Files\Real\Helix Server\rmserver.cfg"
```

The configuration is imported into the following key of the Windows registry:

```
HKEY_CLASSES_ROOT\Software\RealNetworks\Helix Server\11.1\Server 1
```

Helix Proxy Configuration Import Example

The following is a sample command for importing the Helix Proxy configuration into a registry key:

```
rmproxy.exe -import:"Proxy 1" "C:\Program Files\Real\Helix Proxy\rmproxy.cfg"
```

This command imports the configuration into the following key:

```
HKEY_CLASSES_ROOT\Software\RealNetworks\Helix Proxy\11.1\Proxy 1
```

Starting the Application Using the Registry Configuration

After you import a configuration, you can start the Helix Server or Helix Proxy application using the registry configuration. Do this by entering the following command from the application Bin directory:

```
executable registry:key
```

If the registry key name contains spaces, enclose the name in double quotation marks. For Helix Server, for example:

```
rmserver.exe registry:"Server 1"
```

Or, for Helix Proxy:

```
rmproxy.exe registry:"Proxy 1"
```

Installing a Service

The following procedure explains how to install Helix Server or Helix Proxy as a Windows service. You can do this whether or not you imported the configuration file into the Windows registry.

➤ To set up Helix Server or Helix Proxy as a Windows service:

1. Stop the server if it is running.

2. From the Start menu, open a Windows command prompt and navigate to the Helix Server or Helix Proxy Bin directory. For example:

```
cd "C:\Program Files\Real\Helix Server\Bin"
```
3. Install the service by entering the following command:

```
executable -install:ServiceName "parameters"
```

This command includes the following elements:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>executable</td>
<td>Name of the Helix Server or Helix Proxy executable program, either rmserver.exe or rmproxy.exe.</td>
</tr>
<tr>
<td>ServiceName</td>
<td>The name that will appear in the Services dialog box. If you omit ServiceName, the default product name is used. If the name contains spaces, enclose the name in double quotation marks.</td>
</tr>
<tr>
<td>parameters</td>
<td>Either the full path and name of the configuration file, or the Windows registry and key name in the form &quot;registry:key&quot;. You can include any command options along with the key value.</td>
</tr>
</tbody>
</table>

**Note:** The quotation marks surrounding the parameters values are required. Within the parameters string, enclose any element that includes spaces in escaped, double quotation marks (\""). See the examples below for more information.

4. Optionally, choose Start > Run and open the Windows Registry Editor by entering the following command:

```
regedit
```

5. Navigate to the following registry key and verify that the StartupParams values are correct:

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\ServiceName
```

6. In the Services control panel, select the service, right-click, and choose Start.

**Tip:** To start the service automatically when the operating system boots up, ensure that Automatic is set for Startup type in the service’s Properties dialog.

**Service Installation Examples**

The following examples show how to install the Helix Server or Helix Proxy service using a variety of methods. It is important to use quotation marks correctly in the installation command. If you specify parameters improperly, the service may install but fail to start.
Example 1: Registry-Store Configuration, No Spaces in Names
The following example installs a Helix Server service named Helix, using the configuration variables stored in the registry key Config. Because the registry entry is a parameter, it is enclosed in double quotation marks:

```
rmserver.exe -install:Helix "registry:Config"
```

For Helix Proxy, the command uses the different executable name:

```
rmproxy.exe -install:Helix "registry:Config"
```

Example 2: Registry-Store Configuration, Spaces in Names
If the service name or registry key contains spaces, you must enclose the values in quotation marks. The next example installs a Helix Server service named Helix Internet, using the configuration imported into the Server 1 registry key. Because the key name contains spaces, it is enclosed within escaped quotation marks within the quoted registry parameter:

```
rmserver.exe -install:"Helix Internet" "registry:"Server 1\"
```

For Helix Proxy, the command uses the different executable name and the appropriate key name:

```
rmproxy.exe -install:"Helix Internet" "registry:"Proxy 1\"
```

Example 3: Registry-Store Configuration and Command Options
The next example is similar to the preceding example but includes the -m command option to set the server memory maximum to 1 GB. Here the -m value is part of the overall parameter list, so it falls within the quotation marks that surround the registry entry:

```
rmserver.exe -install:"Helix Internet" "registry:"Server 1\" -m 1024"
```

For Helix Proxy, the command is similar:

```
rmproxy.exe -install:"Helix Internet" "registry:"Proxy 1\" -m 1024"
```

Example 4: File-Based Configuration
The next example installs a Helix Server service named Helix Internet using a specified configuration file. Because it is a parameter, the file name and path must be enclosed within quotation marks. However, the path also contains spaces, so the path string is enclosed within escaped quotation marks:

```
rmserver.exe -install:"Helix Internet" "\"C:\Program Files\Real\Helix Server\rmserver.cfg\"
```

For Helix Proxy, the command is the following:
Example 5: File-Based Configuration and Command Options

The last example specifies a configuration file and sets the maximum memory usage at 1 Gigabyte. Note that the -m option and the configuration file path both fall within the quotation marks that indicate the parameter list:

```
rmserver.exe -install:"Helix Internet" "\C:\Program Files\Real\Helix Server\rmserver.cfg" -m 1024"
```

Helix Proxy would take a similar command:

```
rmpoxy.exe -install:"Helix Internet" "\C:\Program Files\Real\Helix Proxy\rmproxy.cfg" -m 1024"
```

Removing a Service

The following procedure removes an installed Windows service.

> To remove a Windows service:

1. Open the **Services** control panel.
2. Right-click the service name, and choose **Stop**.
3. Open a Windows command prompt from the **Start** menu.
4. Navigate to the Helix Server or Helix Proxy Bin directory and enter the following command:
   
   ```
   executable -remove:ServiceName
   ```
   
   Here, **ServiceName** is the name that appears in the services list. If you omitted a service name when you installed the service, you can omit it here to indicate the default product name, such as Helix Server or Helix Proxy. If the service name contains spaces, enclose the name in double quotation marks, as shown in the following example:
   
   ```
rmsrver.exe -remove:"Helix Internet"
```
5. Check for the service name in the **Services** control panel to ensure that the service has been removed.

**Tip:** If the control panel is already open, click the **Refresh** button to update the services list.
Chapter 5: HELIX ADMINISTRATOR

This chapter introduces you to Helix Administrator, the Web-based tool for configuring Helix Server and Helix Proxy. It describes the basics of using this tool.

Starting Helix Administrator

To start Helix Administrator, you need to decide whether to connect using HTTP or HTTPS. The latter is recommended if administering Helix Server or Helix Proxy across the public Internet.

To start Helix Administrator:

1. Start Helix Server or Helix Proxy if it is not already running.

   For More Information: Chapter 4 explains the startup procedure.

2. On Windows, click the browser shortcut added to the desktop by the installation program. Or open one of the following locations in your Web browser. For HTTPS:

   https://address:AdminPort/admin/index.html

   For HTTP:

   http://address:SecureAdminPort/admin/index.html

   If your browser is on the same computer as the server or proxy, you can typically use the localhost address. For example:

   https://localhost:port/admin/index.html

   HTTP and HTTPS communication use separate port numbers that were set during installation. Helix Administrator does not use the standard HTTP and HTTPS ports of 80 and 443.

   Tip: The two port numbers are the values of the AdminPort and SecureAdminPort variables in the configuration file.
3. Helix Administrator responds to the URL request by asking for your user name and password. Enter the user name and password chosen during installation. The password is case sensitive.

**Tip:** The Helix Administrator password is the value of the MonitorPassword variable in the configuration file.

4. Click **OK** to start Helix Administrator.

**Tip:** You can create additional user names and passwords for accessing Helix Administrator. In the online help, see the **Helix Administrator > Security > Authentication** section.

---

**Using the Helix Administrator Interface**

Helix Administrator consists of HTML pages that you use to configure Helix Server or Helix Proxy. The left-hand frame organizes features into functional areas. Pages that display in the right-hand frame typically consist of forms that include fields and pull-down lists.

**Warning!** RealNetworks strongly recommends quitting the browser by closing all windows and tabs when you have finished using Helix Administrator.

**Control Icons**

In Helix Administrator pages that list multiple elements, you can use the control icons depicted in the following illustration.

---

**Helix Administrator Controls**

- Add New Element
- Delete Element
- Duplicate Element
- Move Element Down
- Move Element Up
Applying Changes

When you change configuration information on a Helix Administrator page, click **Apply** at the bottom of the page. If you do not click **Apply**, Helix Administrator discards changes once you navigate to a different page.

**Tip:** An arrow appears next to the **Apply** button and the page title tab to indicate that changes require saving.

Restarting the Server or Proxy

Some configuration changes you make in Helix Administrator require a server or proxy restart. The Helix Administrator interface indicates feature changes that require a Helix Server or Helix Proxy restart. It also prompts you that a change requires a server restart when you click **Apply**. Click the **Restart Server** button to restart the server or proxy.

Queuing Changes for a Later Restart

It is not necessary to restart Helix Server or Helix Proxy immediately after you make a configuration change. In this case, the **Pending Changes** flag appears in the upper-right corner of Helix Administrator. This flag reminds you that all pending changes will go into effect the next time the server or proxy is started.

Closing Open Connections

Restarting the server or proxy shuts down open connections for live events or clips streamed on demand. It’s best, therefore, to make these changes during periods of low use. Two features help you to minimize the impact of a restart:

- The redundant servers or proxies feature redirects RealPlayers to other servers or proxies to finish their media sessions. See “Redundant Servers or Proxies” on page 62.
- The delayed shutdown feature allows media players to report playback statistics before the restart commences. See “Delayed Shutdown” on page 62.

Online Help

You can view context-sensitive help for Helix Server or Helix Proxy by clicking the **Help** link at the top of any Helix Administrator page. The HTML-based help systems then opens in a new window in your browser. Use the left-hand
pane to navigate among topics, which include Helix Administrator operation, configuration file variables, and server registry properties used in reports.

**Tip:** You do not need to use Helix Administrator to access the help system. In the installation directory, navigate to the HelixAdministrator/Docs directory. Open and bookmark home.html in your browser.

**License File**

At least one valid, text-based license file must reside in the License subdirectory of the main installation directory. This file is in an XML format that you can read with a text editor. You can also display information about your licensed features by clicking the **About** link on the Helix Administrator interface. You generally do not need to do anything with the license file as long as the server or proxy reads it correctly on startup.

**Tip:** If you have multiple license files, Helix Administrator shows the values for all of the licenses at once. In this case, you need to read each file individually and calculate additive features, such as the total number of licensed streams.

**Warning!** Making any changes to a license file invalidates the license and prevents the server or proxy from starting. To resolve this, contact RealNetworks for a valid license file.

**Samples**

On Helix Server, click the **Samples** link in the upper-right corner of Helix Administrator to display a page containing links to sample clips and sample embedded players for Flash and Silverlight.

**Tip:** If your Helix Server machine does not include a supported media player, you can play a sample clip from another machine on your network by logging into Helix Administrator from that machine.
Feature Configuration

Using Helix Administrator, you can configure various features of your server or proxy. The following sections explain the major features available for Helix Server and Helix Proxy.

**Tip:** For detailed configuration instructions for each of the following features, display the feature’s administration page and click the **Help** link.

Server and Proxy Features

The following features are available in both Helix Server and Helix Proxy.

**Bookmarking**

Bookmarking allows a media player to start playback where it stopped playback in a previous session. The feature works with any player and is independent of the transport, protocol, or media type.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Helix Server:</th>
<th>Helix Proxy:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Content Management &gt; Bookmarking</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**For More Information:** Helix Proxy does not include a set-up page in Helix Administrator, but you can configure it to handle bookmarks for RTSP media players the same as Helix Server. See the Helix Proxy online help topic **Configuration File > Basic Setup Configuration > Bookmarking Configuration**.

**IP Bindings**

By default, the server or proxy binds to all IP addresses and the localhost address on the machine. The IP bindings feature allows you to bind the server or proxy to a subset of the available addresses.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Helix Server:</th>
<th>Helix Proxy:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Server Setup &gt; IP Binding</td>
<td>Proxy Setup &gt; IP Binding</td>
</tr>
</tbody>
</table>
Ports

The installation program sets the main ports used by the server or proxy. You can change these values on the Ports administration page. For Helix Server, you can perform additional functions such as restricting the UDP port range.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Server Setup &gt; Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Proxy Setup &gt; Ports</td>
</tr>
</tbody>
</table>

Bandwidth Management/Connection Control

Both the server and the proxy allow you to limit the amount of bandwidth used for streaming.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Server Setup &gt; Connection Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Proxy Setup &gt; Bandwidth Management</td>
</tr>
</tbody>
</table>

Delayed Shutdown

Delayed shutdown gives media players time to close their connections and report statistics on a server or proxy shutdown.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Server Setup &gt; Delayed Shutdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Proxy Setup &gt; Delayed Shutdown</td>
</tr>
</tbody>
</table>

Redundant Servers or Proxies

This feature instructs RealPlayers to reconnect to a different server or proxy if the current server or proxy goes offline.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Server Setup &gt; Redundant Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Proxy Setup &gt; Redundant Proxies</td>
</tr>
</tbody>
</table>

Media Delivery

This administration page configures the server-side rate control feature for various media players. Chapter 8 provides background on server-side rate control. Using Helix Administrator, you can define profiles for media players.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Server Setup &gt; Media Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Proxy Setup &gt; Media Delivery</td>
</tr>
</tbody>
</table>
CHAPTER 5: Helix Administrator

User/Group Name

This feature is available on UNIX and Linux. It shifts ownership of the server or proxy process to a user other than root after startup.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Server Setup &gt; User/Group Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Proxy Setup &gt; User/Group Name</td>
</tr>
</tbody>
</table>

Access Control

You can associate connection rules based on IP addresses to allow or deny connections to specific ports.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Security &gt; Access Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Security &gt; Access Control</td>
</tr>
</tbody>
</table>

User Name and Password Authentication

Authentication verifies user identities. On Helix Server, it also verifies processes trying to connect to the server, such as encoders delivering live streams and subscribers requesting content for caching.

| Helix Server: | Security > User Databases  
<table>
<thead>
<tr>
<th></th>
<th>Security &gt; Authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Security &gt; Commerce</td>
</tr>
<tr>
<td>Helix Proxy:</td>
<td>Security &gt; User Databases</td>
</tr>
<tr>
<td></td>
<td>Security &gt; Realms</td>
</tr>
<tr>
<td></td>
<td>Security &gt; Authentication</td>
</tr>
</tbody>
</table>

Activity Monitors

Monitors are real-time tools that display activity occurring on the server or proxy.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Logging and Monitoring &gt; Server Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Logging and Monitoring &gt; Proxy Monitor</td>
</tr>
</tbody>
</table>

Basic Logging

The basic access log records all media streaming events. Basic logging is enabled by default. You can modify the access log to change which types of information it records.

<table>
<thead>
<tr>
<th>Helix Server:</th>
<th>Logging and Monitoring &gt; Basic Logging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Proxy:</td>
<td>Logging and Monitoring &gt; Basic Logging</td>
</tr>
</tbody>
</table>
Customized Logging

The optional, customized logging features allows you to create reports about virtually any activity that occurs on the server or proxy.

<table>
<thead>
<tr>
<th>Server</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Server</td>
<td>Logging and Monitoring &gt; Advanced Logging</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>Logging and Monitoring &gt; Advanced Logging</td>
</tr>
</tbody>
</table>

SNMP

The SNMP feature allows you to monitor the server or proxy from a third-party application using Simple Network Monitoring Protocol. Chapter 6 explains SNMP set-up. Using Helix Administrator you can define the conditions that are monitored.

<table>
<thead>
<tr>
<th>Server</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Server</td>
<td>Logging and Monitoring &gt; SNMP</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>Logging and Monitoring &gt; SNMP</td>
</tr>
</tbody>
</table>

Back-Channel Multicasting

A back-channel multicast allows you to multicast a stream to RealPlayer users while receiving control and quality of service information from each player on a feedback channel.

<table>
<thead>
<tr>
<th>Server</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Server</td>
<td>Broadcast Distribution &gt; Back-Channel Multicasting</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>Proxy Setup &gt; Multicasting</td>
</tr>
</tbody>
</table>

Differentiated Services

The server and proxy can set the bits for precedence and quality of service in IPv4 packets for several streaming media protocols. This allows networks to manage media packets according to different criteria.

<table>
<thead>
<tr>
<th>Server</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helix Server</td>
<td>Transport Settings &gt; Differentiated Services</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>Transport Settings &gt; Differentiated Services</td>
</tr>
</tbody>
</table>

Server-Only Features

Using Helix Administrator, you can set up the following features on Helix Server.
On-Demand Mount Points

An on-demand mount point is a virtual path in a URL that corresponds to a content directory on your network. When you create a new content directory, you define a new mount point under **Server Setup > Mount Points**.

HTTP Delivery

By default, Helix Server does **not** allow HTTP access to content mount points. Using the HTTP delivery feature, you can define exactly which mount points support HTTP or HTTPS delivery. Go to **Server Setup > HTTP Delivery**.

MIME Types

For HTTP delivery, you can define the MIME types that Helix Server recognizes under **Server Setup > MIME Types**.

URL Aliases

This feature, defined under **Server Setup > URLAliasing**, allows you to shorten long URLs by creating aliases for directory paths or files.

Cache Directives

When your server works with proxies, the cache directives feature allows you to define which content the proxies can and cannot cache. See **Server Setup > Cache Directives**.

Broadcasting

Helix Server provides many features for delivering live broadcasts. The administration pages in the **Broadcast** section cover basic unicasting features. Use the pages under **Broadcast Distribution** to set up advanced features.

**For More Information:** See Chapter 12 for background about broadcasting features.

Content Caching

The content caching feature replicates media assets across different Helix Servers. This allows you to upload content to one server but have it available to many servers as needed. See **Content Management > Content Caching**.
Content Browsing

Using **Content Management > Content Browsing** you can define which content directories are browsable from Helix Administrator.

ISP Hosting

For Internet Service Providers, Helix Server works with existing user accounts to make media files available for streaming. Navigate to **Content Management > ISP Hosting**.

Channel Switching

Configurable under **Content Management > Channel Switching**, the channel switching feature allows Helix Server to switch the media input for a continuous stream, eliminating the need to set up a separate RTSP session for each stream.

**For More Information:** Chapter 14 provides background information about channel switching.

Server-Side Playlist

Server-side playlists comprise a sequence of clips or broadcasts that a media player receives in a single RTSP stream. You enable server-side playlist management in the **Content Management > Server-Side Playlist** page.

**For More Information:** Chapter 9 explains the set-up procedures for playlist management.

Media Segmentation

The media segmentation feature converts source content to MPEG-2 transport files (.ts) for delivery to iOS devices. You configure this feature in the **Content Management > Media Segmentation** section of Helix Administrator.

**For More Information:** Chapter 7 explains the delivery and set-up options for media segmentation.

View Source

The view source feature allows RealPlayer to view source file information for presentation. The Helix Administrator **Content Management > View Source** section allows you to configure this feature.
Proxy-Only Features

Using Helix Administrator, you can set up the following features on Helix Proxy.

Proxy Routing
The proxy routing feature allows Helix Proxy to request content from another proxy instead of a server. Configure this feature under Proxy Setup > Proxy Routing.

Pull Splitting
Helix Proxy attempts to conserve bandwidth on Helix Server by splitting all live broadcasts originating on the server. You can configure this feature under Proxy Setup > Splitting.

Cache
On the Proxy Setup > Cache page, you can modify the proxy cache size, change the directory used for the cache, or even turn off caching.
Using Simple Network Monitoring Protocol (SNMP), you can monitor Helix Server from an SNMP management system. This allows you to change Helix Server configuration from a third-party tool, and send notice of important events to an external program. This chapter explains how to set up the SNMP monitoring plug-in and the Helix Server master agent.

**Note:** Helix Proxy also supports SNMP as described in this chapter.

### Understanding SNMP

The following sections describe the components of the Helix Server SNMP monitoring system. Before implementing SNMP on Helix Server, be sure that you understand the basics of SNMP monitoring and know how to operate your chosen SNMP management system.

### SNMP Plug-in

Helix Server includes an SNMP plug-in that monitors its registry for configuration values and events. The plug-in communicates to the master agent using a proprietary protocol. It can send important information about Helix Server operation to the master agent, and update the Helix Server configuration as instructed by the management system. You must configure the plug-in before it can operate.

**For More Information:** The section “Configuring the SNMP Plug-In” on page 71 explains how to set up the plug-in. Refer to “License File” on page 60 for more about licensed features.
**Master Agent**

The SNMP plug-in communicates with the master agent, an executable program included with Helix Server. The master agent then communicates with the management system using the SNMP protocol. The SNMP plug-in and the management system never communicate directly. The master agent can run as an independent application or a Windows service. Once configured, the master agent generally runs without the need for user intervention.

**For More Information:** See “Configuring the Master Agent” on page 72.

**SNMP Protocol**

The master agent uses the SNMP protocol to communicate with the management system. It supports SNMP version 1 (SNMPv1), version 2c (SNMPv2c), and version 3 (SNMPv3). Versions 1 and 2 of the SNMP protocol do not encrypt messages between the two components, and are therefore recommended only when both Helix Server and the management system reside behind a firewall on the same private network.

**Note:** Helix Server does not support SNMP over Internet Protocol version 6 (IPv6). Components must use IPv4 addresses.

**SNMP Version 3 Protocols**

SNMPv3 is suitable for communications over an unprotected network. The User-based Security Model (USM) for SNMPv3 defines two authentication protocols, both of which are supported for Helix Server SNMP:

- **HMAC-MD5-96**
  This protocol is based on MD5. Operations using MD5 occur faster than those using SHA.

- **HMAC-SHA-96**
  This protocol is based on SHA-1. SHA provides a stronger security mechanism than MD5.
SNMP Version 3 Security Levels

SNMPv3 defines three levels of security. The lowest level (noAuthNoPriv) does not provide authentication or privacy, and is comparable to SNMP version 1. The second level (AuthNoPriv) provides authentication but no privacy. The third level (AuthPriv) provides authentication and encryption for all messages.

<table>
<thead>
<tr>
<th>SNMP Version</th>
<th>Authentication Mode</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>version 1 (SNMPv1)</td>
<td>noAuthNoPriv</td>
<td>Authentication is performed by matching an unencrypted community string. This method is not suitable for communication across an unsecured network.</td>
</tr>
<tr>
<td>version 2 (SNMPv2c)</td>
<td>noAuthNoPriv</td>
<td></td>
</tr>
<tr>
<td>version 3 (SNMPv3)</td>
<td>authNoPriv</td>
<td>This mode provides authentication based on the HMAC-MD5 or HMAC-SHA algorithm.</td>
</tr>
<tr>
<td></td>
<td>authPriv</td>
<td>This mode provides authentication based on the HMAC-MD5 or HMAC-SHA algorithm. It also uses encryption based on the CBC-DES (DES-56) standard.</td>
</tr>
</tbody>
</table>

Management System and Management Information Base (MIB)

You can use any third-party SNMP monitoring tool as your management system. The management information base (MIB) determines the Helix Server configuration variables that the management system monitors and controls. It also defines the event traps that the SNMP plug-in can report to the master agent. Helix Server ships with a MIB configuration file named helixserver.my, located in the main Helix Server installation directory.

For More Information: The section “Running a Management System” on page 79 explains the monitoring trees that appear in the management system.

Configuring the SNMP Plug-In

You configure the SNMP plug-in through Helix Administrator. The configuration connects the plug-in to the master agent, and defines which events to report to the management system. To configure the SNMP plug-in, navigate to Logging & Monitoring > SNMP in Helix Administrator. Here you
enable SNMP, define the master agent address, and set trap values. For details, refer to the Helix Administrator online help topic Helix Administrator > Logging and Monitoring > SNMP.

Configuring the Master Agent

The master agent is the intermediary through which the SNMP plug-in and the management system communicate. It must always run on the Helix Server machine. The following sections explain how to modify the master agent configuration file to define your system addresses, users, and security model.

Modifying the Master Agent Configuration File

You use the master.cfg file installed in the Helix Server installation directory to configure the master agent. This allows the master agent to communicate with the SNMP plug-in and the management system. It also defines the security level for each person who uses the management system. Edit this XML-formatted text file using any text, HTML, or XML editor. The following example shows the default configuration file:

```xml
<?xml version="1.0" encoding="US-ASCII"?>
<preferences version="0.5">
    <config ManagerAddress="127.0.0.1" ManagerSNMPPort="162"
            LocalSNMPPort="161" AgentXProtocolPort="705" EngineID="XXX"/>
    <security CommunityString="public"/>
    <SecurityModel ModelType="USM">
        <users UserName="xxx">
            <Authentication Type="MD5" Password="yyy"/>
            <Privacy Type="DES" Password="zzz"/>
        </users>
        <users UserName="unsecureUser">
            <Authentication Type="NONE" Password=""/>
            <Privacy Type="NONE" Password=""/>
        </users>
    </SecurityModel>
</preferences>

<SecurityToGroup SecurityModel="USM" User="unsecureUser" Group="v3Group"/>
<SecurityToGroup SecurityModel="USM" User="test" Group="testGroup"/>
<SecurityToGroup SecurityModel="v2" User="vishal" Group="v1v2group"/>
<SecurityToGroup SecurityModel="v1" User="public" Group="v1v2group"/>

<SecurityModel ModelType="VACM">
    <groups Name="v3Group" SecurityModel="USM" SecurityLevel="1"/>
Defining Master Agent Addresses and Ports

The following lines in the master agent configuration define the basic communication between the master agent, the SNMP plug-in, and the management system:

```
<config ManagerAddress="127.0.0.1" ManagerSNMPPort="162" 
LocalSNMPPort="161" AgentXProtocolPort="705" EngineID="XXX"/>
```

The following table explains the values you should set for these attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagerAddress</td>
<td>The IPv4 address of the management system. The master agent uses this address to send traps to the management system. You must specify an IP address, not a DNS name. The master agent does not support IPv6 addresses.</td>
</tr>
<tr>
<td>ManagerSNMPPort</td>
<td>The port used by the management system to listen for communication from the master agent. The default is port 162.</td>
</tr>
</tbody>
</table>
The following lines set the parameters for SNMP security. The USM security model defines the access rights for each person running the management system. The configuration file predefines two users. The first user operates with no security, which is equivalent to using SNMP version 1. The second user defines authentication and privacy, the highest security under SNMPv3.

You can modify or delete these predefined users, as well as create additional users by adding new `<users>…</users>` lists within the USM section:

```
<security CommunityString="public"/>
<SecurityModel ModelType="USM">
  <users UserName="xxx">
    <Authentication Type="MD5" Password="yyy"/>
    <Privacy Type="DES" Password="zzz"/>
  </users>
  <users UserName="unsecureUser">
    <Authentication Type="NONE" Password=""/>
    <Privacy Type="NONE" Password=""/>
  </users>
</SecurityModel>
```

### Setting Up SNMP Security

The following lines set the parameters for SNMP security. The USM security model defines the access rights for each person running the management system. The configuration file predefines two users. The first user operates with no security, which is equivalent to using SNMP version 1. The second user defines authentication and privacy, the highest security under SNMPv3.

You can modify or delete these predefined users, as well as create additional users by adding new `<users>…</users>` lists within the USM section:

```
<security CommunityString="public"/>
<SecurityModel ModelType="USM">
  <users UserName="xxx">
    <Authentication Type="MD5" Password="yyy"/>
    <Privacy Type="DES" Password="zzz"/>
  </users>
  <users UserName="unsecureUser">
    <Authentication Type="NONE" Password=""/>
    <Privacy Type="NONE" Password=""/>
  </users>
</SecurityModel>
```
The following table defines the master agent configuration attributes that define the SNMP security level and permissions.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommunityString</td>
<td>Password used with SNMP version 1 or 2. You can ignore the &lt;security/&gt; tag if you are using SNMP version 3. If you are using version 1 or 2, you can ignore the USM settings.</td>
</tr>
<tr>
<td>UserName</td>
<td>Name of the user as defined in the management system.</td>
</tr>
<tr>
<td>Authentication Type</td>
<td>Type of authentication used with SNMPv3. Valid values are MD5 for the HMAC-MD5 algorithm, or SHA for the HMAC-SHA algorithm. A value of NONE indicates an unsecured user.</td>
</tr>
<tr>
<td>Privacy Type</td>
<td>For privacy type, you can enter NONE for no privacy or DES for CBC-DES encryption.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for authentication or privacy. SNMPv3 uses separate passwords for authentication and privacy. You do not need to define a certain password, however, if you used NONE as the authentication or privacy type.</td>
</tr>
</tbody>
</table>

Defining a View Access Control Model

The view access control model (VACM) available through SNMPv3 allows you to define precisely which Helix Server SNMP objects each viewer can see and control. VACM is optional, and you should be familiar with how it works within your SNMP management system before you define view privileges through the master agent configuration file. The following sections provide an example of how to set up view access for a specific user.

Assigning a User to a Group

Each person who uses VACM must be defined in the <SecurityModel> list as an SNMPv3 user. The following example shows a user defined to use SNMPv3 with authentication but no privacy:

```
<SecurityModel ModelType="USM">
    <users UserName="Maria">
        <Authentication Type="MD5" Password="tl73jkl98"/>
        <Privacy Type="NONE" Password=" "/>
    </users>
    ...other users defined here...
</SecurityModel>
```
Using a `<SecurityToGroup/>` tag, you assign each user to a group name that you create. In the following example, the user Maria is assigned to a group named v3Group:

```xml
<SecurityToGroup SecurityModel="USM" User="Maria" Group="v3Group"/>
```

The following table explains the `<SecurityToGroup/>` tag attributes.

<table>
<thead>
<tr>
<th>VACM <code>&lt;SecurityToGroup/&gt;</code> Tag Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
</tr>
<tr>
<td>-----------</td>
</tr>
</tbody>
</table>
| SecurityModel | Security model for this user. Choose one of the following:  
- v1 for SNMP version 1  
- v2c for SNMP version 2c  
- USM for SNMP version 3 |
| User | The user's name. |
| Group | The group to which the user is assigned. Groups are defined with `<groups/>` tags. |

Creating Groups

Within the `<SecurityModel>` list, a `<groups/>` tag defines each group. A group has three views, indicating which parts of Helix Server the user can monitor and control. In the following example, v3group is assigned the fullView view for receiving traps and reading Helix Server variables. It is part of the noView view for writing configuration changes to the Helix Server registry:

```xml
<SecurityModel ModelType="VACM">  
  <groups Name="v3Group" SecurityModel="USM" SecurityLevel="1"  
    Context="" Notify_View="fullView" Read_View="fullView"  
    Write_View="noView"/>

  ...more groups and views defined here...
</SecurityModel>
```

The following table explains the `<groups/>` tag attributes.

<table>
<thead>
<tr>
<th>VACM <code>&lt;groups/&gt;</code> Tag Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Name</td>
</tr>
</tbody>
</table>
| SecurityModel | Security model for this group. Choose one of the following:  
- v1 for SNMP version 1  
- v2c for SNMP version 2c  
- USM for SNMP version 3 |
VACM <groups/> Tag Attributes (continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecurityLevel</td>
<td>Security level for this group. Choose one of the following:</td>
</tr>
<tr>
<td></td>
<td>- 0 for noAuthNoPriv</td>
</tr>
<tr>
<td></td>
<td>- 1 for authNoPriv</td>
</tr>
<tr>
<td></td>
<td>- 2 for authPriv</td>
</tr>
<tr>
<td>Context</td>
<td>An optional, named subset of object instances in the management</td>
</tr>
<tr>
<td></td>
<td>information base.</td>
</tr>
<tr>
<td>Notify_View</td>
<td>The name of the view assigned to the group for receiving traps.</td>
</tr>
<tr>
<td>Read_View</td>
<td>The name of the view assigned to the group for reading SNMP objects</td>
</tr>
<tr>
<td></td>
<td>values corresponding to Helix Server registry values.</td>
</tr>
<tr>
<td>Write_View</td>
<td>The name of the view assigned to the group for writing changes to</td>
</tr>
<tr>
<td></td>
<td>SNMP object values and thereby changing Helix Server</td>
</tr>
<tr>
<td></td>
<td>configuration values.</td>
</tr>
</tbody>
</table>

Defining Views

Within the <SecurityModel> list, a <views/> tag defines each view. The view identifies a group of objects by an OID from the management information base (MIB). All objects that fall under that OID are included in the view. In the following example, the fullView view is included while the noView view is excluded, allowing no access:

```xml
<SecurityModel ModelType="VACM">
    ...groups defined here...
    <views Name="fullView" OID="1.3" Mask="" Included="1"/>
    <views Name="noView" OID="1.3" Mask="" Included="0"/>
    ...more views defined here...
</SecurityModel>
```

The following table explains the <views/> tag attributes.

VACM <views/> Tag Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The view name. Groups are assigned up to three views (Notify_View,</td>
</tr>
<tr>
<td></td>
<td>Read_View, and Write_View) in each &lt;groups/&gt; tag.</td>
</tr>
<tr>
<td>OID</td>
<td>Object ID of a node. All objects that fall under that node in a tree</td>
</tr>
</tbody>
</table>
Running the Master Agent on Windows

On Windows, you can run the master agent as a service or as an application. The following sections explain how to start the master agent in either mode.

Restarting the Master Agent Service

If you installed the master agent as a Windows Service, the agent starts up automatically. If you change the master agent configuration, restart the agent service by locating the master agent service with Settings>Control Panel>Administrative Tools>Services. In the Services dialog, right-click on SNMP Master Agent and choose Restart.

Tip: Right-click SNMP Master Agent and choose Properties to change the master agent operation. Using this dialog, for example, you can disable automatic start-up or restart the service automatically if it fails.

Starting the Master Agent as an Application

The following procedure explains how to start the master agent as a Windows application. Do this only if the master agent has not been installed as a service, or you have disabled the service through the Services dialog.

➤ Starting the master agent as a Windows program:

1. Open a command prompt using Start>Program>Accessories>Command Prompt.
2. Navigate to the Helix Server installation directory. For example:
   cd “C:\Program Files\Real\Helix Server”
3. The master agent executable, master.exe, resides in the Bin subdirectory. Start it by entering the following:
   Bin\master.exe master.cfg
4. Start Helix Server as described in “Starting the Server or Proxy on
Windows” on page 39.

Starting the Master Agent on UNIX

The following procedure explains how to start the master agent as a UNIX
background process.

➤ To start the master agent on UNIX:

1. If Helix Server is running, shut it down as described in “Stopping the
Server or Proxy” on page 44.

2. Log in as root.

3. From the command line, navigate to the Helix Server installation
directory. For example:
   
   # cd /usr/local/Real/HelixServer

4. The master agent executable, master, resides in the Bin subdirectory. Start
   it as a background process:
   
   # ./Bin/master master.cfg &

5. Start Helix Server as described in “Starting the Server or Proxy on UNIX
or Linux” on page 42.

Running a Management System

Once you have the SNMP plug-in and master agent configured and running,
you can use your management system to monitor and control Helix Server.
From the management system, locate the MIB file, which is named
helixserver.my and resides in the Helix Server installation directory. Compile
the MIB file if necessary for your management system. You then connect the
management system to the master agent using the Helix Server IP address and
port defined in the master agent configuration file.

For More Information: For information about compiling the
MIB file and connecting your management system to the
master agent, refer to your SNMP manager documentation.
The section “Defining Master Agent Addresses and Ports” on
page 73 explains the master agent port usage.
Monitor Tree

The MIB file produces monitoring trees that allow you to monitor Helix Server operation and control the settings of certain variables. The $hs$ monitor tree contains objects related to Helix Server monitoring. (On Helix Proxy, it is the $hp$ monitor tree.) These objects, described in the following table, cannot be changed by the management system.

<table>
<thead>
<tr>
<th>Object</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsClients</td>
<td>Number of media players currently connected. Covers all supported players and communications protocols.</td>
</tr>
<tr>
<td>hsRTSPClients</td>
<td>Number of media players currently communicating through the RTSP control protocol.</td>
</tr>
<tr>
<td>hsRTMPClients</td>
<td>Number of media players currently communicating through the RTMP control protocol.</td>
</tr>
<tr>
<td>hsHTTPClients</td>
<td>Number of media players currently communicating through the HTTP control protocol.</td>
</tr>
<tr>
<td>hsSecureClients</td>
<td>Number of media players currently communicating through the HTTPS or RTMPS control protocol.</td>
</tr>
<tr>
<td>hsUDPTransports</td>
<td>Number of media players currently using User Datagram Protocol (UDP).</td>
</tr>
<tr>
<td>hsTCPTransports</td>
<td>Number of media players currently using Transmission Control Protocol (TCP).</td>
</tr>
<tr>
<td>hsMulticastTransports</td>
<td>Number of media players connected on multicast.</td>
</tr>
<tr>
<td>hsBandwidthUsage</td>
<td>Total outgoing bandwidth in Kilobits per second (Kbps) used by Helix Server.</td>
</tr>
<tr>
<td>hsPercentCPUUsage</td>
<td>Percentage of CPU used by Helix Server processes.</td>
</tr>
<tr>
<td>hsMemoryUsage</td>
<td>Amount of memory used by Helix Server in bytes.</td>
</tr>
<tr>
<td>hsPlatform</td>
<td>Helix Server operating system.</td>
</tr>
<tr>
<td>hsVersion</td>
<td>Helix Server software version.</td>
</tr>
<tr>
<td>hsAccumulatedBandwidth</td>
<td>Network bandwidth use in Kilobits per second</td>
</tr>
<tr>
<td>hsEncoderCount</td>
<td>Number of media encoders currently delivering live streams.</td>
</tr>
<tr>
<td>hsUptime</td>
<td>Time since last Helix Server restart in seconds.</td>
</tr>
<tr>
<td>hsMMSClientCount</td>
<td>Number of media players currently communicating through the MMS control protocol.</td>
</tr>
</tbody>
</table>
Configuration Tree on Helix Server

The hsConfig tree contains objects that map to Helix Server configuration variables, such as Helix Server ports and the various traps that can be set. You can monitor these objects using any version of SNMP.

<table>
<thead>
<tr>
<th>Object</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsPorts</td>
<td>This subtree allows you to change the main Helix Server communications ports.</td>
</tr>
<tr>
<td>hsTrap</td>
<td>The objects in this subtree correspond to the event traps defined for the SNMP plug-in. For more information, refer to “Configuring the SNMP Plug-In” on page 71.</td>
</tr>
</tbody>
</table>

**Tip:** A configuration change occurs immediately without the need to restart Helix Server. However, to make the change permanent, you need to write the change to the configuration file using the Control Tree.

Configuration Tree on Helix Proxy

The hpConfig tree contains objects that map to Helix Proxy configuration variables. You can monitor these objects using any version of SNMP.

<table>
<thead>
<tr>
<th>Object</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>hpMaxConnections</td>
<td>Maximum number of clients that Helix Proxy will accept.</td>
</tr>
<tr>
<td>hpMaxGatewayBandwidth</td>
<td>Maximum gateway bandwidth allowed by Helix Proxy.</td>
</tr>
<tr>
<td>hpMaxProxyBandwidth</td>
<td>Maximum Helix Proxy bandwidth allowed.</td>
</tr>
</tbody>
</table>
Control Tree

The hsControl tree contains objects that you can use to control certain Helix Server operations. By setting hsRestartServer to true, for example, you can restart Helix Server remotely.

<table>
<thead>
<tr>
<th>Object</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsWriteConfigFile</td>
<td>Writes configuration changes made by the management system to the configuration file when set to true or 1. This ensures that the configuration changes remain in effect after a restart. For more on the configuration file, refer to the Helix Server online help.</td>
</tr>
<tr>
<td>hsStopServer</td>
<td>Stops Helix Server when set to true or 1. For information on shutdown options, refer to the Helix Server online help.</td>
</tr>
<tr>
<td>hsRestartServer</td>
<td>Restarts Helix Server when set to true or 1.</td>
</tr>
</tbody>
</table>
In this section, you’ll learn about various features for configuring Helix Server and Helix Proxy to deliver content to specific types of clients. For information about supported clients and media formats, refer to the Helix Administrator online help.
For Apple devices running iOS 3.0 or later, Helix Server can segment and deliver streaming content that is encoded with supported audio and video codecs. This chapter describes how this process works, explains optional features, and guides you in setting up media delivery to iOS devices.


### Segment Delivery Overview

An iOS device receives a streaming clip or broadcast as a series of MPEG-2 transport stream files (.ts). Each .ts file encodes a portion of the stream, typically enough data to provide 10 seconds of playback. An iOS presentation thereby consists of a series of .ts files delivered using HTTP. The process of creating the .ts files is called *segmentation*.

When requesting the individual transport stream files, the iOS device refers to a text-based playlist file (.m3u8) that lists the segments in their playback order. Data delivery is entirely client-driven. That is, the iOS device determines when to request each transport file in the playlist based on its current buffering and playback state.

**Tip:** To support content segmentation for iOS delivery, your Helix Server should have at least 512 MB of application memory available. The section “Memory Allocation” on page 47 explains application memory use.
iOS Source Media Requirements

An iOS device supports the following codecs for source content that is segmented into transport files:

<table>
<thead>
<tr>
<th>Codec</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>H.264 baseline profile, level 3 or below. Helix Server can stream H.264 content at up to 3 Mbps. Refer to the Apple documentation for their recommendations of streaming speeds for various iOS devices.</td>
</tr>
<tr>
<td>Audio</td>
<td>AAC or AAC+ up to 700 Kbps. Enhanced AAC+ is not supported. The sampling rate can be up to 48 kHz. Stereo content is recommended.</td>
</tr>
</tbody>
</table>

Because content for iOS devices is converted to the .ts format, codec-compatible source content can be in any file format that Helix Server can read. The common formats are MPEG-4 (.mp4), 3GPP (.3gp), F4V (.f4v), and QuickTime (.mov).

**Tip:** Hinting is not required. If the source content is hinted, Helix Server ignores the hint track during segmentation.

Multi-Rate Streams

An iOS device can shift between the different rates of a clip or live broadcast that is encoded for multiple bandwidths. For multi-rate content, Helix Server automatically creates the segment and playlist files required to deliver the content at different bandwidths. The iOS device requests files for different bandwidth options based on its ongoing assessment of available bandwidth.

**Tip:** In the Apple documentation, this feature is called stream alternates.

Low-Bandwidth Audio Stream

Apple requires that iOS media applications have available at least one stream deliverable at 64 Kbps, which includes the overhead for the HTTP connection and the segment packaging. For live and on-demand video content encoded at multiple rates, Helix Server automatically segments an audio-only stream if all stream combinations require a delivery rate higher than 64 Kbps.

To create these segments, Helix Server selects from the multi-rate content the audio stream closest to, but not higher than, 48 Kbps. This limit ensures that the entire delivery bandwidth of media and overhead does not exceed 64 Kbps. For this reason, RealNetworks recommends that source files always include an audience that contains an audio stream encoded at 48 Kbps or lower.
Multi-rate Audience Groups

With multi-rate source clips or broadcasts, Helix Server may create playlist combinations that differ from the source audiences. For example, suppose that you use Helix Producer to encode an H.264/AAC clip using the four audiences shown in the following table.

**Source Clip Tracks and Bandwidths**

<table>
<thead>
<tr>
<th>Audience</th>
<th>Video Speed</th>
<th>Audio Speed</th>
<th>Media Streaming Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 Kbps</td>
<td>48 Kbps</td>
<td>148 Kbps</td>
</tr>
<tr>
<td>2</td>
<td>200 Kbps</td>
<td>96 Kbps</td>
<td>296 Kbps</td>
</tr>
<tr>
<td>3</td>
<td>500 Kbps</td>
<td>48 Kbps</td>
<td>548 Kbps</td>
</tr>
<tr>
<td>4</td>
<td>700 Kbps</td>
<td>96 Kbps</td>
<td>796 Kbps</td>
</tr>
</tbody>
</table>

When it segments this source clip for iOS devices, Helix Server pairs the 96 Kbps audio track with the fastest video track and the 48 Kbps audio track with the other video tracks. This means that the 200 Kbps video stream does **not** use the 96 Kbps stream of the original clip. Helix Server also creates a 48 Kbps audio-only track.

**Segmented Presentation Tracks and Bandwidths**

<table>
<thead>
<tr>
<th>Playlist</th>
<th>Video Speed</th>
<th>Audio Speed</th>
<th>Media Streaming Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100 Kbps</td>
<td>48 Kbps</td>
<td>148 Kbps</td>
</tr>
<tr>
<td>1</td>
<td>200 Kbps</td>
<td><strong>48 Kbps</strong></td>
<td><strong>248 Kbps</strong></td>
</tr>
<tr>
<td>2</td>
<td>500 Kbps</td>
<td>48 Kbps</td>
<td>548 Kbps</td>
</tr>
<tr>
<td>3</td>
<td>700 Kbps</td>
<td>96 Kbps</td>
<td>796 Kbps</td>
</tr>
<tr>
<td>4</td>
<td>—</td>
<td>48 Kbps</td>
<td>48 Kbps</td>
</tr>
</tbody>
</table>

**For More Information:** See also “Selecting the Initial Rate” on page 92.

**Gzip Support for Playlists**

If an iOS device indicates that it supports the gzip format, Helix Server delivers playlists compressed with gzip. This saves bandwidth when lengthy playlists are needed, such as with the playlist management feature described in Chapter 9. The compression requires no configuration and is transparent to both the iOS device user and the Helix Server administrator.
Compatibility with Other Features

The following table explains the availability of other Helix system features for iOS devices.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Compatibility Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>content caching</td>
<td>The content distribution feature is not supported for delivering segment files from a publisher to a subscriber. However, you can use it to distribute source clips to Helix Server subscribers. Each subscriber then segments the content separately when the source clip is requested. Refer to the Helix Administrator online help for information about content caching.</td>
</tr>
<tr>
<td>splitting</td>
<td>You can use the splitting feature described in Chapter 8 to deliver live streams to multiple Helix Servers in push or pull mode. Each Helix Server receiver segments the content separately.</td>
</tr>
<tr>
<td>server-side rate control</td>
<td>Content delivery to iOS devices is entirely client-driven using the HTTP protocol. The server-side rate control feature described in Chapter 8 is not used.</td>
</tr>
<tr>
<td>playlist management</td>
<td>Many of the server-side playlist management features described in Chapter 9 function with iOS devices.</td>
</tr>
<tr>
<td>channel switching</td>
<td>Channel switching, described in Chapter 14, cannot be used with live broadcasts to iOS devices.</td>
</tr>
<tr>
<td>SLTA</td>
<td>You can use the SLTA feature described in Chapter 13 to deliver simulated live broadcasts to iOS devices if the source clips are encoded using the appropriate codecs and file formats.</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>All media delivered to the iOS device is sent over HTTP. As a result, Helix Proxy cannot cache or split iOS content. All requests are handled by the origin Helix Server in pass-through mode.</td>
</tr>
</tbody>
</table>

On-Demand Segmentation Steps

The following figure illustrates the basic steps required to make on-demand content available to iOS devices.
Steps for Segmenting On-Demand Content

➤ To make on-demand content available for iOS devices:

1. An encoder such as Helix Producer creates a clip at a single rate or multiple rates using the supported codecs. It writes the content to a supported file format such as MPEG-4. The finished clip is placed under a source mount point or path. For example, the clip might be available for RTSP streaming under the predefined mount point /iPhone-src/.

2. The iOS device places an HTTP or HTTPS request for the clip using the /m3ugen/ mount point. For example:

   http://helixserver.example.com/m3ugen/iPhone-src/news.mp4

   **For More Information:** See “Using the m3ugen Mount Point” on page 91.

3. Helix Server begins to convert the requested clip into transport segments (.ts) and generates the playlist file. The files are written to the base directory of a segment mount point associated with the source mount point. For example, clips residing in the /iPhone-src/ mount point may be segmented under the predefined /iPhone/ mount point.

4. In response to the iOS device’s initial request, Helix Server returns the playlist file (.m3u8).

5. The iOS device uses the playlist file to make HTTP requests for the transport segment files located under the segment mount point (/iPhone/). The access log records a separate entry for each segment request.

   **Tip:** Although HTTPS is supported for transport segments, HTTP is recommended because of the lower overhead. To
ensure security, segments can be encrypted and key files delivered using HTTPS.

Broadcast Segmentation Steps

The following figure illustrates the general steps required to make broadcast content available to iOS devices. The process for segmenting broadcast content is similar to that for on-demand content, except that the playlist file is continually updated as new segments are generated.

Steps for Segmenting Broadcast Content

➤ To make broadcast content available for iOS devices:

1. An encoder generates a live stream at a single rate or multiple rates using the supported codecs. As with an on-demand clip, the live stream can specify a file format such as MPEG-4 or F4V. The encoder delivers the stream to the mount point used with the broadcast method (/broadcast/, /rtpencoder/, or /rtmplive/).

   Tip: Helix Producer can deliver the live stream using any push method, such as Helix Push, Helix Advanced Push, or Helix Multicast Push. Chapter 10 explains broadcast methods.

2. Helix Server immediately begins to generate transport segments (.ts) in the segment mount point associated with the broadcast mount point (/iPhone/ in this example). The server continually deletes older segments and updates the playlist file so that viewers joining the broadcast start with the latest segment.
3. Optionally, Helix Server can create archive files for the broadcast under the segment mount point or a separate archive mount point. The preceding figure illustrates an archive created under a user-defined mount point named /iPhoneArchive/.

4. The iOS device places an HTTP request for the broadcast stream. The request URL includes the /m3ugen/ and the appropriate broadcast mount points. For example:
   http://helixserver.example.com/m3ugen/broadcast/news.mp4

5. In response to the iOS device’s initial request, Helix Server returns the current playlist file (.m3u8).

6. The iOS device uses the playlist file to begin making HTTP requests for the transport segment files and updated playlist files created under the segment mount point. The access log records a separate entry for each segment request.

   **Note:** Because iOS broadcasting delivers a series of short clips, the playback point for each viewer differs depending on when the viewer requested the broadcast. If each transport segment is 10 seconds long, for example, one viewer may be up to 10 seconds ahead or behind another viewer in the broadcast timeline.

**Using the m3ugen Mount Point**

To receive segmented media, the iOS device requests a clip or live stream using the predefined /m3ugen/ mount point. This mount point appears directly after the URL address, preceding any other mount point, alias, or path. For example, the following URL requests a clip under the /iPhone-src/ mount point:

http://helixserver.example.com/m3ugen/iPhone-src/news.mp4

The following example is for a broadcast stream generated by Helix Producer:

http://helixserver.example.com/m3ugen/broadcast/live.mp4

When the viewer clicks the HTTP or HTTPS link, Helix Server segments the requested stream, writing the output as multiple .ts files.

   **Note:** The segmentation process occurs only when needed. If segments exist for an on-demand clip, Helix Server delivers the
existing segments. If the source clip is removed, however, Helix Server returns an HTTP 404 error to the client even if the segments still exist.

**Requesting Playlist Files Directly**

For presentations that have already been segmented, including archives for live broadcasts, the iOS device can request the playlist file directly. The request specifies the segment mount point but does not include the /m3ugen/ mount point. For example:

http://helixserver.example.com/iPhone/news-20120428-145721.m3u8

**Tip:** If segments have been purged, the request for the playlist file fails. If the original source was an on-demand clip, though, requesting the clip again using the /m3ugen/ mount point causes Helix Server to generate new segments for the content.

**Selecting the Initial Rate**

In the master playlist for multi-rate live or on-demand content, choices are ordered from lowest to highest bandwidth. An iOS device requests the lowest bandwidth first, shifting to higher rates when possible. Because of this default behavior, the start of the presentation may have lower quality than necessary for the device’s connection speed. You can modify this using two methods:

- Set the value of the LowestRateFirst variable in the configuration file (rmserver.cfg) to 0 to reorder the options in the master playlist from highest to lowest bandwidth.

  **Warning!** If the highest bandwidth choice exceeds the iOS device’s connection bandwidth, the presentation may stutter until the device can shift to a lower-speed option.

- Use the ir parameter to request a specific starting bandwidth. This parameter is compatible with either setting of the LowestRateFirst variable.

  **For More Information:** In the Helix Administrator online help, refer to the topic Configuration File > Content Management Configuration > MPEG-2 Transport Stream Configuration.
Query String Parameter for Initial Bandwidth Choice

Adding the query string parameter \textit{ir=streaming\_rate} to the request URL specifies the initial media rate to use in bits per second:

http://helixserver.example.com/m3ugen/iPhone-src/news.mp4?ir=600000

\textbf{Tip:} The \textit{ir} parameter cannot be used to select an audio-only stream. It has no effect on single-rate media sources.

\textbf{Note:} The \textit{ir} parameter specifies the maximum bandwidth for video and audio combined. The actual delivery bandwidth will be higher because of network overhead for HTTP and TCP.

Reordered Master Playlist

The \textit{ir} parameter causes Helix Server to reorder the playlist so that the first choice is the bandwidth option that most closely matches the parameter value without exceeding the value. For example, the value \textit{ir=600000} might result in the following order of choices within the main playlist.

<table>
<thead>
<tr>
<th>Order</th>
<th>Video Speed</th>
<th>Audio Speed</th>
<th>Media Streaming Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>500 Kbps</td>
<td>48 Kbps</td>
<td>548 Kbps</td>
</tr>
<tr>
<td>1</td>
<td>100 Kbps</td>
<td>48 Kbps</td>
<td>148 Kbps</td>
</tr>
<tr>
<td>2</td>
<td>200 Kbps</td>
<td>48 Kbps</td>
<td>248 Kbps</td>
</tr>
<tr>
<td>3</td>
<td>700 Kbps</td>
<td>96 Kbps</td>
<td>796 Kbps</td>
</tr>
<tr>
<td>4</td>
<td>—</td>
<td>48 Kbps</td>
<td>48 Kbps</td>
</tr>
</tbody>
</table>

\textbf{Note:} After the iOS device requests the first playlist option, it can shift to any other option. The order of the remaining options in the playlist does not affect its choices.

Bookmarking Presentations

The bookmarking feature adds a \textit{start\_time} query string parameter to a request URL to set the starting point for an on-demand presentation. The bookmark refers to the number of seconds into the presentation timeline where playback should start:

http://helixserver.example.com/m3ugen/iPhone-src/news.mp4?start=157
Using Third-Party Query Parameters

You can include additional query string parameters in the URL used with an /m3u8en/ request. These parameters are added to the playlist request URL returned to the iOS device. Query parameters included with the playlist file requests are added to every segment URL (request for a .ts file) in an on-demand presentation or live broadcast.

**Tip:** Query parameters are not written to playlist files stored on disk. Instead, the parameters unique to each client’s request URL are automatically added to the segment request URLs in the copy of the playlist file returned to the client.

**For More Information:** Query string parameters are recorded in the Helix Server log file along with each request URL. See the online help topic **Helix Administrator > Logging and Monitoring** for information about logging.

Defining Segmentation Mount Points

The segmentation feature matches on-demand and broadcast mount points to segment mount points. The following sections explain issues and features that affect how you set up mount points.

Predefined Mount Points for Segmentation

Helix Server predefines the following source paths for content segmentation:

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/iPhone-src/</td>
<td>This mount point is meant to hold source files (.mp4 or .3gp files, for example) that will be segmented into .ts files. Its base directory is the Content/iPhone-src/ directory of the Helix Server installation directory.</td>
</tr>
<tr>
<td>/broadcast/</td>
<td>This mount point receives live broadcast streams from Helix Producer using the Helix Push, Helix Advanced Push, or Helix Multicast Push delivery method. For details about these broadcast methods, see Chapter 10.</td>
</tr>
</tbody>
</table>
Tip: By default, segmentation is enabled for /iPhone-src/ and any new source paths you create. It must be manually enabled for broadcast mount points, however. You can enable a source path in the Content Management > Media Segmentation page of Helix Administrator.

User-Defined Mount Points for Segmentation

If you need more flexibility than the default setup provides, you can create new mount points for use with content segmentation. The sample setup in the following illustration shows one possible implementation.

Possible Source and Segment Mount Point Configurations
The preceding sample setup defines the following:

- A user-defined /media/ mount point has been added as a second source for on-demand content suitable for segmentation. When an iOS device requests a clip from this mount point, segments are created under the user-defined /iPhone2/ mount point.

- Live broadcasts delivered under the /broadcast/, /rtpencoder/, and /rtmplive/ mount points are segmented under the /iPhone/ mount point just as in the default setup. However, broadcasts segmented under /iPhone/ are archived under a separate mount point, /iPhoneArchive/.

Defining Source and Segmentation Mount Points

You can create and pair as many source and segment mount points as necessary. The following sections explain issues to keep in mind when designing your set-up.

Base Directory on Local Machine or Network Share

Every segment mount point has a base directory on the local machine or network share. For the default /iPhone/ mount point, for example, the base directory is the Content/iPhone subdirectory of the main Helix Server installation directory.

FileSize Maximum for Base Directory

For each segment mount point, you can set a size limit for files in the base directory, such as 50 GB. You can thereby use mount points to manage how much segmented content remains on hand.

For More Information: See “Purging Files from Mount Points” on page 98.

Source Paths for Segmenting Certain Content

When you enable a source mount point in Helix Administrator, you can specify the entire mount point or a source path, which is a mount point and a subdirectory structure. If you specify a source path, only the source content under the subdirectory structure can be segmented.

Specifying a source path is useful if you want to segment only certain live broadcasts. For example, you can set the source path for a live broadcast stream created by a RealNetworks encoder as the following:
/broadcast/segmented/

Here, /segmented/ indicates a virtual path defined by the encoder. When it starts the broadcast, the encoder includes this virtual path with the broadcast name. For example:

segmented/live.mp4

The presence of the virtual path in the live stream name causes Helix Server to segment the broadcast. As long as the general mount point /broadcast/ is not enabled as a source path, live streams that do not include the virtual path are not segmented.

Multiple Source Mount Points Writing to a Single Segment Mount Point

Two or more source mount points can write to the same segment mount point. Clips under both /iPhone-src/ and /media/ in the preceding example can be segmented under the single /iPhone/ mount point, for example.

Do Not Use Source Mount Points for Segments

You typically should not generate segment files under standard source mount points used with streaming protocols such as RTSP or RTMP. Segment mount points must be enabled for HTTP delivery. Doing this for source mount points allows Web browsers to cache the content held under the source mount point.

Alternate Content for Segment Mount Points

Each segment mount point can direct iOS devices to a certain resource if the segmented content is missing or the streaming connection fails. See “Redirecting Clients on Errors” on page 100 for details.

Using Archive Mount Points

By default, segments for live broadcasts are not archived. The optional archiving feature preserves the segments and a playlist file from each live broadcast. As an further option, each segment mount point can write archive files to a separate archive mount point. This is shown in the preceding sample setup with the /iPhoneArchive/ mount point.

Default Archive is Segment Mount Point

If you enable archiving but do not create a separate archive mount point, archives are written to the segment mount point. In the Helix Server default
setup, enabling archiving generates archives in the default segment mount point, /iPhone/.

**Archive Mount Point Tied to Segment Mount Point**

An archive mount point is associated with the segment mount point, not the broadcast mount point. For example, do the following if you want to create separate archive areas for broadcasts delivered on the /broadcast/, /rtpencoder/, and /rtmplive/ mount points:

- Segment the broadcast content under separate mount points. For example, /broadcast/ streams can be segmented under /iPhone/ while /rtpencoder/ and /rtmplive/ streams go under /iPhone2/.

- Assign a separate archive mount point to each segment mount point. For example /iPhone/ segments can be archived under /iPhoneArchive/ and /iPhone2/ segments under /iPhoneArchive2/.

Tip: An archive mount point inherits the segment duration setting (10 seconds by default) from the segment mount point configuration.

**Disabling HTTP for Archive Mount Points**

If you do not enable HTTP delivery for an archive mount point, iOS devices cannot access the archive files. Clients can access any archives stored under a segment mount point, however.

**Full Path for URLs for Archive Files**

The URLs in archived playlist files include the full directory path to the content but do not include the protocol, server, and port information (such as http://helixserver.example.com).

**No Maximum File Size on Archive Mount Points**

Archive mount points are not subject to a file size maximum. For example, you might limit the base directory of a segment mount point to 50 Gigabytes but allow archives files under an archive mount point to reach the disk maximum.

**Purging Files from Mount Points**

Optionally, you can purge segments on each Helix Server restart as well as define a size limit for files stored under a segment mount point.
Purging Segments on Restart

By default, Helix Server automatically purges segments and playlists for live broadcasts on each restart. Optionally, you can purge any other segmented content on a restart as well, including segments for on-demand clips and archived broadcasts.

A startup purge applies to source mount points rather than segment mount points. Suppose that segments and playlists for the on-demand source mount points /iPhone-src/ and /media/ are stored under the segment mount point /iPhone/. If /iPhone-src/ uses a startup purge but /media/ does not, only the segments and playlists that correspond to source content residing under /iPhone-src/ are deleted from the /iPhone/ mount point on a restart.

Limiting Segment Mount Point Size

You can set a size limit on the directories where segments are stored. For example, you can set a limit of 50 Gigabytes for files stored under the base directory of the /iPhone/ mount point. The following sections explain the rules for how Helix Server purges files when the maximum size is reached.

Archive Files Included in Size Total

If broadcast archives are store under the segment mount point, the archive files count toward the size maximum. However, broadcast archive files are not deleted in a purge.

**Warning!** If a segment directory fills with archive files, Helix Server will not be able to generate segments for new broadcasts or on-demand requests. For this reason, it is a good idea to store broadcast archives in a separate archive directory.

Broadcast Segments Continually Purged

Helix Server continually deletes older segments for a live broadcast. An active broadcast typically stores about four segments at any time. For this reason, a live broadcast generally does not initiate a purge of the segment mount point as long as archiving is turned off or archives are written to a different mount point.

Least Used Content Purged First

When it purges files, Helix Server deletes the least recently requested content first. When it purges an on-demand presentation, the segment files, the playlist file, and the directory structure are deleted.
Active Content Not Purged
If on-demand content is being delivered when a purge action takes place, Helix Server does not include the segment files for the content in use in the file purge. This on-demand content may be removed in the next file purge, however.

Redirecting Clients on Errors
Helix Server has two optional features to redirect iOS devices to alternate content if the requested playlist file for on-demand or live content is not available. It can redirect clients to an alternate Helix Server that hosts the same content or provide clients a URL to an error message.

Note: You can define only one of these features for each segment mount point. Each mount point can be configured differently, however.

Directing Clients to an Alternate Server
Helix Server can write the address of an alternate server into each playlist file generated under a segment mount point. If the iOS device’s HTTP requests to the current Helix Server fail because the server goes offline, for example, the client begins to request the content from the alternate server.

Tip: For this feature to work, duplicate content on the alternate server must exist in the same path as on the original server. The content distribution feature is not supported for distributing segmented content between servers. You must duplicate the content by some other method.

Note: This feature is available only to iOS devices running iOS 3.1 or later.

Supplying a Alternate URL on Errors
The alternate URL option directs the iOS device to a different content URL (on-demand clip or HTML page) if requests for playlist files fail. The primary purpose of this feature is to deliver a standby or error message. To remain valid if the server goes offline, the URL should point to content on a separate Helix Server or Web server.
Setting Up Mount Points in Helix Administrator

When you are ready to define your segmentation setup, log into Helix Administrator and perform the tasks outlined in the following sections.

For More Information: For detailed instructions on each of the following tasks, refer to the Helix Administrator online help.

Creating Mount Points

To create a source or a segment mount point, log into Helix Administrator and click Server Setup > Mount Points. The mount points section allows you to define each new mount point and set the base directory where the mount point files are stored.

Tip: For each segment mount point, leave Allow File Creation and Allow File Deletion set to No. These values do not affect content segmentation.

Setting Up HTTP Delivery

For each mount point that will hold transport segments or key files that can be requested, you must enable the use of HTTP or HTTPS. In Helix Administrator, click Server Setup > HTTP/S Delivery to display the configuration page.

Verifying MIME Types

Verify that the following MIME types, which are required to deliver transport segments and playlists, are configured in the MPEG2_Transport_Stream list of the Helix Server configuration file (rmserver.cfg):

<table>
<thead>
<tr>
<th>MIME Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>application/vnd.apple.mpegurl</td>
<td>m3u8</td>
</tr>
<tr>
<td>video/MP2T</td>
<td>ts</td>
</tr>
</tbody>
</table>

Note: These MIME types cannot be changed in the Helix Administrator Server Setup > MIME Types page, which allows you to set other MIME types.

Defining Segmentation Features

Click Content Control > Media Segmentation in Helix Administrator when you are ready to configure mount points for segmentation. On this page you enable the segmentation feature and select source mount points for which
content segmentation is allowed. For each source mount point, you set the corresponding segment mount point and define its features.

### Features Configurable in Helix Administrator for Segment Mount Points

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Default</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment Duration</td>
<td>Duration of content encoded in each segment.</td>
<td>10 seconds</td>
<td>n/a</td>
</tr>
<tr>
<td>Minimum Playlist Segments</td>
<td>Minimum number of segments for each stream.</td>
<td>3</td>
<td>n/a</td>
</tr>
<tr>
<td>Enable Live Archive</td>
<td>Whether to archive live broadcasts written to this mount point.</td>
<td>no archiving</td>
<td>page 97</td>
</tr>
<tr>
<td>Archive Mount Point</td>
<td>Mount point to use for archives if archiving is enabled.</td>
<td>none</td>
<td>page 97</td>
</tr>
<tr>
<td>Enable Segment Encryption</td>
<td>Encrypt segments and generate a key file.</td>
<td>no encryption</td>
<td>page 106</td>
</tr>
<tr>
<td>Key File Interval</td>
<td>Interval in minutes at which to change the encryption algorithm.</td>
<td>none</td>
<td>page 107</td>
</tr>
<tr>
<td>Key File Mount Point</td>
<td>Mount point where key file is saved.</td>
<td>segment mount point</td>
<td>page 108</td>
</tr>
<tr>
<td>Key File URL</td>
<td>Host that delivers the key file.</td>
<td>none</td>
<td>page 108</td>
</tr>
<tr>
<td>Alternate URL</td>
<td>URL for content to display on error.</td>
<td>none</td>
<td>page 100</td>
</tr>
<tr>
<td>Alternate Server</td>
<td>URL for server to contact on error.</td>
<td>none</td>
<td>page 100</td>
</tr>
<tr>
<td>Segment Purge Size</td>
<td>Maximum size for all segment files, including archive files.</td>
<td>5 Gigabytes</td>
<td>page 98</td>
</tr>
<tr>
<td>Use Absolute Paths for Segments</td>
<td>Include absolute paths in URLs to segments within playlist files.</td>
<td>no (use relative paths)</td>
<td>page 103</td>
</tr>
</tbody>
</table>

### File Names and Directory Structures

The playlist and segment files created under a segment or archive mount point correspond to the file name of the requested content. This section explains file and directory names used for on-demand content, live broadcasts, and broadcast archives.
Output Directories and File Names for Segmented On-Demand Content

The preceding figure illustrates the content directories and files created when an iOS device requests an on-demand clip named `news.mp4` that is stored under the `/iPhone-src/` source mount point.

**Subdirectory Creation**

On the first request for on-demand or broadcast content, Helix Server creates a subdirectory under the segment mount point’s base directory. This prevents file name collision in case multiple source mount points share the same segment mount point.

The subdirectory name corresponds to the source mount point. For example, requesting a clip from the `/iPhone-src/` mount point creates an `iPhone-src/` subdirectory under the `/iPhone/` mount point’s base directory. For archived broadcasts, Helix Server creates a subdirectory such as `broadcast/` under the archive mount point.

**Playlist File**

The playlist file uses the same file name (including the file extension) as the requested content. For on-demand content, the name includes a timestamp. For archived content, the name includes a timestamp and the `-archive` flag.
The file name ends with the extension .m3u8. For example, the stream news.mp4 could generate the following playlist files:

<table>
<thead>
<tr>
<th>Type</th>
<th>Playlist File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>on-demand clip</td>
<td>news.mp4-20120428-145732.m3u8</td>
</tr>
<tr>
<td>live broadcast</td>
<td>news.mp4.m3u8</td>
</tr>
<tr>
<td>broadcast archive</td>
<td>news.mp4-20120428-145732-archive.m3u8</td>
</tr>
</tbody>
</table>

**Tip:** Links to segment files within a playlist file are relative to the playlist file location. Optionally, you can write full HTTP URLs to each segment. See the Helix Administrator online help topic [Helix Administrator > Content Management > Media Segmentation > Defining Segment Mount Points](#) for details.

**Note:** The -archive flag is configurable using the Helix Server ArchiveExtension configuration attribute. For details, refer to the Helix Administrator online help topic [Configuration File > Basic Setup Configuration > File System Configuration > On-Demand Mount Point Configuration](#).

**Timestamp Format**

The timestamp for files and directories reflects the file generation time of the requested clip or the stream origination time of the live broadcast. The format is the following:

```
/YYYYMMDD-HHMMSS
```

These values mean the following:

<table>
<thead>
<tr>
<th>YYYY</th>
<th>four digits for the year, such as 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM</td>
<td>two digits for the month, such as 04 for April</td>
</tr>
<tr>
<td>DD</td>
<td>two digits for the date, as in 12</td>
</tr>
<tr>
<td>HH</td>
<td>hour in 24-hour format, as in 16 for 4 p.m.</td>
</tr>
<tr>
<td>MM</td>
<td>minutes</td>
</tr>
<tr>
<td>SS</td>
<td>seconds</td>
</tr>
</tbody>
</table>

**Tip:** When it receives a request for a source clip that has been segmented, Helix Server compares the generation time of the clip with the segment timestamps. If the source clip has been updated, Helix Server segments the source again. This ensures that segments are regenerated if source clips are modified.
Top Segment Directory

At the same directory level as the playlist file, Helix Server creates a top segment directory to hold segment files for single-rate content. The directory name uses the same name as the playlist file minus the `.m3u8` file extension. For example:

<table>
<thead>
<tr>
<th>Type</th>
<th>Directory Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>on-demand clip</td>
<td>/news.mp4-20120428-145732</td>
</tr>
<tr>
<td>live broadcast</td>
<td>/news.mp4</td>
</tr>
<tr>
<td>broadcast archive</td>
<td>/news.mp4-20120428-145732-archive</td>
</tr>
</tbody>
</table>

Multi-rate Playlist Files

Segmenting a multi-rate clip generates a secondary playlist file for each bit rate. These files reside in the top segment directory. Suppose that the requested `news.mp4` clip is encoded for 80 Kbps and 200 Kbps. In this case, the following secondary playlist files are created:

<table>
<thead>
<tr>
<th>Type</th>
<th>Directory Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>on-demand clip</td>
<td>news.mp4-20120428-145732-mr80k.m3u8</td>
</tr>
<tr>
<td></td>
<td>news.mp4-20120428-145732-mr200k.m3u8</td>
</tr>
<tr>
<td>live broadcast</td>
<td>news.mp4-mr80k.m3u8</td>
</tr>
<tr>
<td></td>
<td>news.mp4-mr200k.m3u8</td>
</tr>
<tr>
<td>broadcast archive</td>
<td>news.mp4-20120428-145732-mr80k-archive.m3u8</td>
</tr>
<tr>
<td></td>
<td>news.mp4-20120428-145732-mr200k-archive.m3u8</td>
</tr>
</tbody>
</table>

Note: For multi-rate content, the main playlist file contains only rate information and links to each of the secondary playlist files. The iOS device then requests secondary playlist files based on its determination of the available bandwidth.

Multi-rate Subdirectories

For a multi-rate stream, Helix Server creates additional subdirectories under the top segment directory to hold the different sets of segment files. For a stream encoded at 80 Kbps and 200 Kbps, Helix Server creates the following subdirectories:

<table>
<thead>
<tr>
<th>Type</th>
<th>Directory Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>on-demand clip</td>
<td>/news.mp4-20120428-145732-mr80k</td>
</tr>
<tr>
<td></td>
<td>/news.mp4-20120428-145732-mr200k</td>
</tr>
</tbody>
</table>
Here, the subdirectory with the -mr80k suffix holds segment files for the 80 Kbps encoding. The subdirectory ending in -mr120 holds segment files for the 200 Kbps second rate.

### Segment File Names

Segmented transport files are named \( n-xxxxx.ts \) in which \( n \) is a sequential integer starting with 1. This integer is followed by a hyphen and a randomly generated, five-digit number. For example, the first file might be 1-02875.ts, the second file could be 2-21487.ts, and so on. The last file includes an -END flag, as in 486-47214-END.ts.

**Tip:** You can omit the random digits in file names by setting the SegmentNameRandomization value to 0 for the source mount point in the Helix Server configuration file. Refer to the Helix Administrator online help topic **Configuration File > Basic Setup Configuration > File System Configuration > On-Demand Mount Point Configuration** for details.

### Omitted Characters

If a file or directory name includes one of the following special characters, the character is replaced with an underscore in file names and subdirectories:

/ ? < > \
: * | "

### Encrypting Segments

Optionally, Helix Server can encrypt segment files for on-demand or live streams. If you enable encryption for a segment mount point, all segments created under that mount point require a key file to play.

**Tip:** RealNetworks recommends delivering key files using HTTPS. Encrypted segment files can be delivered safely using HTTP as long as security is maintained on the key file.
Note: Encryption follows the Apple specification and uses the Advanced Encryption Standard (AES_128), which implements a 128-bit key and PKCS7 padding. Each time it segments on-demand or broadcast content, Helix Server changes the hash phrase it uses with the encryption algorithm.

Key File

When encryption is enabled, Helix Server generates a key file that allows the iOS device to decrypt the segment files. It automatically adds a URL for the key file to the playlist file.

Periodic Key File Updating

By default, each piece of segmented content uses a single key file. Optionally, Helix Server can periodically change the hash phrase it uses during each segmentation process. This results in multiple key files generated for each piece of segmented content.

Note: If multiple key files are generated for a single piece of content, Helix Server updates playlist files accordingly as each new key file is generated.

Key File Names

Note the following about key file names:

- Key files follow the naming convention used with playlist files and end with the file extension .key.
- Key files also include the five random integers included with segmented transport files (see “Segment File Names” on page 106).
- For multi-rate clips, key file names include the streaming rate designation.
- When periodic key file updating is used, each new key file includes a flag in the form -kfn in which n is an integer beginning with 1.
Key File Name Example

Segmenting the multi-rate news.mp4 stream and periodically updating the encryption hash phrase results in multiple key files like the following:

| on-demand clip: | news.mp4-20120428-145732-mr80k-kf1-21496.key |
|                | news.mp4-20120428-145732-mr80k-kf2-23983.key |
|                | ...additional key files...                   |
| live broadcast: | news.mp4-mr80k-kf1-02628.key                 |
|                | news.mp4-mr80k-kf2-32579.key                 |
|                | ...additional key files...                   |
| broadcast archive: | news.mp4-20120428-145732-mr80k-kf1-32098-archive.key |
|                | news.mp4-20120428-145732-mr80k-kf2-43376-archive.key |
|                | ...additional key files...                   |

Writing Key Files to Alternate Locations

By default, key files are written to the same directory that holds the segment files. You can write the key file to an alternate location if you wish to do any of the following:

- Host the key file on an alternate Helix Server.
- Host the key file on a Web server with optional user name and password authentication.

Key File Write Location

When you define a segment mount point in Helix Administrator, you specify the alternate mount point where the key file is written. This might be a Helix Server mount point on a network disk shared with a Web server, for example. The Web server can then enforce user authentication for the key file.

**Note:** Helix Server replicates the directory structure of the segmented content under the key file mount point. The directories contain only key files, however.

Key File URL

If you write the key file to a separate location, you must also provide a URL to the location. In the playlist file, Helix Server concatenates the URL with the key file location under the key file mount point, allowing the iOS device to locate the key file.
Progressive Download Alternative

As an alternative to segmenting content, you can allow the iOS device to download and play an entire QuickTime or MPEG clip. With progressive download, each file begins to play shortly after the download starts. This delivery method allows the iOS device to cache the entire file, however.

**Note:** Refer to the following subsections only if you are creating progressive download content.

File Formats for Progressive Download

Using a product such as Helix Producer, encode content in one of the following formats using one of the supported audio and video codecs:

- 3GPP (.3gp)
- MPEG-4 (.mp4, .m4v)
- MP3 (.mp3)
- QuickTime (.mov)

**Note:** Content must be encoded using the progressive download option.

Codecs for Progressive Download

The iOS device supports a wider range of codecs for downloadable content than for segmented content:

<table>
<thead>
<tr>
<th>Video</th>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG-4, H.264</td>
<td>AAC, AAC+, MP3</td>
</tr>
</tbody>
</table>

Hosting Downloadable Content

You can host downloadable content on any HTTP-enabled mount point. Link to the content using a standard HTTP or HTTPS URL. For example:

http://helixserver.example.com/iPhone/sports.mp4
This chapter explains server-side rate control. Using rate control, Helix Server can vary the streaming rate of a multi-rate clip to compensate for network congestion. This helps to minimize rebuffering by adjusting the streaming rate for the network’s current conditions.

**Note:** Helix Proxy provides server-side rate control identical to that found in Helix Server.

**Server-Side Rate Control Basics**

Using server-side rate control, Helix Server can adjust the streaming bit rate of an on-demand clip or live broadcast based on the player’s buffer status. This allows Helix Server to compensate for fluctuating bandwidth that may occur due to network congestion or hand-off between cellular access points.

**Rate Control Basic Requirements**

Use of the server-side rate control feature depends on three general conditions:

- **multi-rate clip or broadcast**
  Helix Server can shift streaming rates only if the requested clip or broadcast is encoded in a multi-rate format. The section “Media Formats Used with Server-Side Rate Control” on page 114 lists the acceptable formats.

- **buffer information**
  Helix Server must have information about the media player’s buffering characteristics. As described in “Buffer Modeling” on page 127, Helix Server adjusts streaming rates based on a model of the player’s buffer state.
• status reports

Helix Server must receive periodic status reports from the media player to
gauge network congestion and ensure that its buffering model is accurate.
For information, refer to “Receiver and Sender Reports” on page 124.

Types of Rate Control

When Helix Server streams a multi-rate clip or broadcast, it uses one of two
types of server-side rate control, or the client-side SureStream rate control.
The following sections explain these different streaming methods.

Note: If a media player does not meet the conditions required
for any type of rate control, Helix Server can still stream a
multi-rate clip or broadcast to the player. However, it does not
shift the streaming rate during the presentation. Network
congestion is therefore more likely to cause the media player to
rebuffer the presentation.

3GPP Rate Adaptation

3GPP rate adaptation is a standards-based method of server-side rate control
available to some 3GPP clients. It uses the standards-based RTP packet
format. Helix Server can use 3GPP rate adaptation for the following clients:

• 3GPP Release 6 players that support 3GPP rate adaptation and NADU
  reports for RTP streaming. These players must send RTCP receiver reports
  as expected by b=RR. Helix Server can provide more efficient rate
  adaptation to players that also utilize the RTSP 3GPP-Link-Char header.

• 3GPP Release 5, PSS-compliant media players that implement the Annex
  G video buffer model. These players must send RTCP receiver reports as
  expected by b=RR.

• Mobile RealPlayers or Helix DNA-based players that support 3GPP
  Release 5 or Release 6 media types, as well as the RTP packet format.

Helix Rate Adaptation

Helix rate adaptation is available for some RealNetworks media players, as well
as players based on Helix DNA. Helix Server supports Helix Adaptation for the
following media players:

• Desktop RealPlayer 11
For More Information: See “Enabling Rate Control for Desktop RealPlayers” on page 122.

- Mobile RealPlayers or Helix DNA-based players that implement Helix rate adaptation with RDT version 3 for RDT streaming, and provide RDT feedback reports.

Tip: Some of these players can switch to 3GPP rate adaptation if you turn off Helix rate adaptation support or disable the use of the RDT packet format.

SureStream Rate Adaptation

SureStream rate adaptation is the older, client-side method of rate adaptation used by desktop RealPlayers that predate RealPlayer 11. It requires the use of a multi-rate SureStream clip (RealAudio or RealVideo) and the RDT packet format. Helix Server switches to this type of rate control automatically for media players that support SureStream but not server-side rate control.

Note: Because the client makes rate shifting decisions when using SureStream rate control, none of the information contained in this chapter applies to this method of rate adaptation.

Live Broadcasts and Rate Control

Rate control works for live broadcasts streamed to any media player that supports Helix or 3GPP rate adaptation. Media players using 3GPP rate adaptation must also send the 3GPP-Link-Char header to support upshifting. Players that do not support either of these adaptation methods receive a single-rate stream for the duration of the broadcast.

Live Rate Control and Reduced Start-Up Delay

Live rate adaptation is compatible with the reduced start-up delay feature. With this feature, Helix Server buffers video streams so that media players receive a video key frame when the stream begins. This reduces the time required for the player to begin to render the stream.

Live Rate Control and Splitting

Live rate control is compatible with most broadcast splitting arrangements, in which a Helix Server transmitter sends the source streams to Helix Server
receivers in either push or pull mode. The receiver that handles the session for a particular media player makes all rate control adjustments for that player.

**Note:** Low-latency broadcasts and live rate adaptation are not compatible. Running a low-latency broadcast automatically disables Helix Server from using live rate control for any media player sessions.

**For More Information:** Chapter 12 explains splitting arrangements, and how to configure transmitters and receivers. The Helix Administrator online help explains low-latency broadcasts.

**Rate Control with Simulated Live Broadcasts**

Live rate control is available for simulated-live, RealMedia streams delivered by the SLTA utility. It is not used with any other media formats streamed by SLTA, however.

**For More Information:** Chapter 13 explains simulated-live broadcasts.

**Media Encoders Used with Live Rate Control**

To enable live rate control, a media encoder must deliver a broadcast that encodes multiple streams at different bit rates. The following media encoders can be used with Helix Server:

- Helix Producer broadcasting with SureStream. This enables rate control for RealNetworks media players using Helix rate adaptation.
- Helix Producer broadcasting with a 3GPP Release 6 format.
- Envivio 4Caster series.


**Media Formats Used with Server-Side Rate Control**

For rate control to function, Helix Server must deliver a clip that includes multiple streams encoded at different bit rates. For 3GPP players, you can use the supported 3GPP Release 6 codecs or a multi-rate container file. For RealNetworks media players, you can use SureStream.
Stream Adaptation for 3GPP Release 6 Players

Media players that support 3GPP Release 6 can use 3GPP rate adaptation. See the Helix Administrator online help topic Streaming Basics > Media Types > Standards-Based Formats for information about the codecs that Helix Server supports.

Multi-rate Container Files for 3GPP Release 5 or Release 6 Players

An .mrc file can contain multiple tracks encoded with a 3GPP Release 5-compliant codec. An .mrc file can contain only one audio track, so rate shifting affects only the video track.

SureStream Clips for RealNetworks Players

A SureStream clip encodes multiple streams at different bit rates. The mechanism used to control the shifting of bit rates during playback varies according to the media player version:

- For desktop RealNetworks media players earlier than RealPlayer 11, Helix Server uses SureStream rate adaptation. This causes RealPlayer to request the upshifts and downshifts.
- For RealPlayer 11 and later, Helix Server manages stream shifting based on the Helix rate adaptation model. This model allows delivery of all existing SureStream clips.
- For RealPlayer mobile players and many Helix-based players, Helix Server manages stream shifting based on its Helix rate adaptation model. For these players, however, rate shifting affects only the video track. If the SureStream clip contains multiple audio tracks, the initial audio track is used throughout the presentation.

Media Encoding Guidelines

When you encode a multi-rate clip or broadcast for server-side rate control, Helix Server allows upshifting to the next higher stream rate only if that rate is no more than a certain percentage above the speed of the current rate. This is called the maximum oversend rate.

Suppose that the maximum oversend rate is 200 percent. For a clip encoded with 32, 64, 128, and 256 Kbps streams, the media player can upshift through all streams because no rate is more than 200 percent of the preceding rate. If
the clip contains just 64 and 256 Kbps streams, however, the media player cannot upshift from the 64 Kbps stream.

For More Information: The Helix Administrator online help topic Helix Administrator > Server Setup > Media Delivery (Rate Control) > Default Profile and Preset Values lists the maximum oversend rate used when you choose a default streaming profile using Helix Administrator. You can also set a value manually in a user agent profile with the MaxOversendRate variable.

Recommended Encoding Rates for Mobile Networks

The following are recommended media rates for multi-rate content intended to stream on mobile networks. These rate distributions ensure that no media rate is more than 140 percent of the next lower media rate.

• For GPRS rates, encode at 15, 20, 25, and 30 kbps.
• For UMTS rates, encode at 75, 85, 95, 105, and 115 kbps.
• For use with both GPRS and UMTS environments, encode at 15, 20, 25, 30, 55, 75, 85, 95, 105, and 115 kbps.

User Agent Profiles

You configure rate control by defining a series of profiles. Each profile is closely tied to the streaming capabilities of a media player or group of players. For instance, you might define one profile for desktop RealPlayers and one for Helix-based clients. Each profile reflects the player’s streaming environment, such as its typical network speed and its media buffer size.

UAS Profiles

Helix Server can write each profile to a separate user agent settings (UAS) file. This is an XML-formatted text file that uses the extension .uas. Helix Server stores the .uas files in the ClientProfiles subdirectory of its main installation directory.

When you use Helix Administrator to create a profile, the file name follows the profile name. For example, if you create a profile called Helix DNA Player, the UAS file is named Helix DNA Player.uas. The UAS file can use any unique file name, however.
CHAPTER 8: Rate Control

Configuration File Profiles

Creating separate UAS files is recommended because of its convenience. However, you can also store profile information in the Helix Server configuration file. Each profile is created within the configuration file’s UserAgentSettings list.

For More Information: For details about the profile syntax, refer to the Helix Administrator online help.

RDF Files

A media player may have an associated Resource Description Framework (RDF) file. When the media player contacts Helix Server, it indicates the Internet location of its RDF file, which typically resides on a Web server hosted by the media player’s manufacturer.

Like the UAS files generated by Helix Server, the RDF file describes the player’s streaming capabilities. Helix Server provides the option to use the RDF file or ignore it. If you choose to use it, Helix Server downloads the RDF file and integrates some of the RDF values into the client’s UAS file.

Note: A media player can send Helix Server the RDF file location in the x-wap-profile header of an HTTP request or any RTSP message.

Tip: To ensure that Helix Server can retrieve RDF files, configure your firewall to allow Helix Server to contact external HTTP servers on TCP port 80. For more about firewalls, see Chapter 2.

Profile Value Inheritance

A rate control profile (a child profile) can inherit values from another profile (a parent or base profile). For example, a profile might inherit values from the Default profile stored in the Helix Server configuration file. To these values, it adds its own values for variable not defined within the Default profile. A profile stored in the configuration file can inherit values from a profile defined within a UAS file, and vice versa.
Inheritance Chains

Inheritance can extend across any number of profiles. For example, profile A can inherit values from profile B, which inherits values from profile C, and so on. A single profile can have only one base profile, however. If a profile and its base profile set conflicting values for the same variable, Helix Server uses the value from the child profile.

Profile Configuration Errors

In most situations, Helix Server compensates for profile configuration problems:

• missing configuration values

It is OK for a configuration variable not to be defined in a child profile or any profile from which the child inherits values. In this case, Helix Server applies the default value for the undefined variable.

For More Information: For information about rate control default values, refer to the Helix Administrator online help topic Helix Administrator > Server Setup > Media Delivery (Rate Control) > Default Profile and Preset Values.

• missing base profile

If a profile inherits values from a base profile that is missing or misconfigured, these values revert to their defaults. Helix Server writes an error to its error log warning about a missing base profile.

For More Information: See the Helix Server online help topic Helix Administrator > Logging and Monitoring > Basic Logging > Error Log for information about the error log.

• looping inheritance

An example of an inheritance loop is when profile A inherits values from profile B, which inherits values from profile A. In this case, if profile A is used, Helix Server treats profile B as missing and writes an error message about an inheritance loop.

User Agent Strings

Helix Server uses a media player’s user-agent string to determine which rate control profile to use for the player. Formats for user-agent strings vary widely,
and you should know the user-agent strings sent by individual media players when defining profiles. Typically, the user-agent string identifies the application, version, and compatibility:

\[\text{application/version (compatibility)}\]

For example, the following user-agent string is for a Helix-based media player:

\[\text{HelixDNAClient/10.0.0.0 (symbian)}\]

The user-agent string can also identify the player’s capabilities, such as which stream formats it can render. When the media player connects to Helix Server, it passes its user-agent string in the RTSP connection headers. For each rate control profile, you define one or more user-agent strings that the profile matches. Matches can be partial or exact, depending on your needs.

**Exact Profile Matching**

You can set a profile to match a media player’s user-agent string exactly by adding an equals sign (=) to the beginning of the string:

\[=\text{HelixDNAClient/10.0.0.0 (symbian)}\]

In this case, the user-agent string listed in the profile must match the user-agent string sent by the client character-for-character in its entirety. The match is also case sensitive.

**Partial String Matches**

A profile might include a partial user-agent string. Such a string, for instance, might match all media players of a certain brand. It does this by identifying a common element of the user-agent string sent by all models of this brand. For example, the following user-agent string matches all version 10 Helix DNA clients:

\[\text{HelixDNAClient/10}\]

The preceding string matches the user-agent strings sent by all Helix DNA version 10 clients (including versions 10.1, 10.2, and so on) running on any operating system. Unlike exact user-agent string matching, partial matching is **not** case sensitive.

**Multiple Profile Sorting**

Because of partial matching, more than one rate control profile may match a user-agent string sent by a media player. In this case, Helix Server selects the profile to use by applying the following criteria in order:
1. Exact match.
   If a profile provides an exact match to the user-agent string, Helix Server uses that profile and ignores profiles that provide partial matches.

2. Most complete match.
   If multiple profiles are partial matches, Helix Server chooses the profile that provides the greatest number of exactly matched characters.

3. Last child match.
   If profiles inherit values, the child profile furthest from the parent is used. For example, suppose that profile B inherits values from profile A, and profile C inherits values from profile B. If profiles B and C equally match a user-agent string, Helix Server chooses profile C.

4. First found.
   If all of the preceding criteria still result in multiple, matching profiles, Helix Server uses the first of these profiles it found as it searched for the matches.

Profiles Shipped with Helix Server

Helix Server includes a number of predefined profiles, as well as a default profile.

Default Profile

The Helix Server configuration file defines a profile named Default. The server uses this profile if the user-agent string sent by the client does not match any other profiles defined within the configuration file or the separate UAS files. You can use the Default profile as the base profile when defining additional streaming profiles.

For More Information: The Helix Administrator online help topic Helix Administrator > Server Setup > Media Delivery (Rate Control) > Default Profile and Preset Values describes the values set by the Default profile.

Predefined User-Agent Files

The following are the user-agent profiles included as separate .uas files in the ClientProfiles subdirectory of the Helix Server installation directory. These
profiles inherit values from the Default profile, then set additional values specific to the client type:

- DesktopPlayers.uas – QuickTime and VLC Desktop Clients
  - QuickTime
  - VLC media player
- Motorola.uas – Motorola Mobile Clients
  - MOT-A820/00.00.00 MIB/2.1
  - MOT-MOTORAZRV9
- NokiaR1M.uas – RealNetworks Mobile Clients
  - RealOnePlayer
  - RealMedia Player/3
  - RealMedia Player/s
  - RealMedia Player (epoc_
  - RealMedia Player HelixDNA/Client/10.0.0.6839 (S60 HX cvs_cays221_2008
  - RealMedia Player HelixDNA/Client/10.0.0.6839 (S60 HX cvs_cays221_2009
- PVPlayer.uas – PacketVideo Clients
  - PVPlayer 3.0 PVPlayer/3.1
  - PVPlayer/3.4
  - PVPlayer 3.4
- QtvPlayer.uas – LG, Samsung, and Qualcomm QtvPlayer
  - QtvPlayer
  - RealMedia Player (HelixDNA/Client)/10.0.0.0 (Qualcomm Inc.)
  - QtvPlayer (org=SS; model=13; terversion=11
  - SAMSUNG-SGH-A707
- RealPlayer.uas – RealNetworks Desktop Clients
  - RealMedia Player (HelixDNA/Client)
  - RealMedia Player HelixDNA/Client
  - RealMedia Player (HelixDNA/Client)/10.0.0.7217 (win32)
  - RealMedia Player HelixDNA/Client/10.0.1.260 (win32)
Enabling Rate Control for Desktop RealPlayers

Desktop RealPlayers use a version of client-side rate control. However, RealPlayer 10.5 and later support server-side rate control if you enable it. To do this, open the RealPlayer.uas file in the ClientProfiles subdirectory. In the profiles for RealMedia Player versions 10.0.0.7217 and higher, change the values of UseServerSideRateAdaptation and UseServerSideRateAdaptationForLive from never to clientselect:

```xml
<Var UseServerSideRateAdaptation="clientselect"/>
<Var UseServerSideRateAdaptationForLive="clientselect"/>
```

**Note:** RealNetworks recommends that you do this only in consultation with RealNetworks support staff.

Streaming Rate Selection

The following sections explain the user agent settings that affect how much bandwidth Helix Server may use for a client connection, and how Helix Server selects the initial streaming rate for a multi-rate clip.

**Maximum Sending Rate**

Using Helix Administrator, you can set a maximum bandwidth for a profile. This is optional, and the default maximum is 10 million bits per second. It is useful to lower the maximum, however, for networks that limit the bandwidth of each client connection. In this case, Helix Server considers any stream that exceeds the maximum sending rate as unstreamable. It excludes these streams when choosing the initial streaming rate or upshifting to higher rates.

**For More Information:** In the user agent settings, the MaxSendRate variable controls this sending rate. For details, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Rate Control Configuration.
Initial Rate Selection

When a media player requests a multi-rate clip, Helix Server evaluates several user agent values to determine which encoding to stream. Once it has established the initial rate criterion to use, Helix Server chooses the encoding that is closest to the specified bandwidth without exceeding it. The following are the use agent values, listed in order from highest to lowest priority, that Helix Server evaluates to select the initial streaming rate:

1. Helix Server first evaluates query string parameters appended to the request URL. The optional \texttt{ir=rate} parameter indicates the preferred bit rate in bits per second. The following example shows the query parameter in an RTSP request:

   \begin{verbatim}
   rtsp://helixserver.example.com/video1.3gp?ir=28000
   \end{verbatim}

   The next example shows the \texttt{ir} parameter in an SDPgen request:

   \begin{verbatim}
   http://helixserver.example.com/sdpgen/video1.3gp?ir=28000
   \end{verbatim}

2. If no \texttt{ir} parameter is present in the request URL, Helix Server next determines the initial streaming rate based on the value of the \texttt{Subscribe} header sent as part of the RTSP \texttt{SET_PARAMETER} request. This value is sent only by RealNetworks and Helix-based media players.

3. If \texttt{UseRel6InitialRateSelection} is set to 1 \texttt{(true)} in the media player’s user agent settings, the next evaluated criterion is the stream choice presented in the \texttt{stream.id} of the RTSP \texttt{SETUP} request.

   \begin{quote}
   \textbf{For More Information}: This variable must be set directly in the user agent settings. For more on the \texttt{UseRel6InitialRateSelection} variable, refer to the Helix Administrator online help topic \texttt{Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration}.
   \end{quote}

4. For 3GPP clients, Helix Server uses the guaranteed bandwidth value (\texttt{GBW}) returned by the client in the \texttt{3GPP-Link-Char} header. Not all 3GPP clients provide this header, however.

5. Helix Server evaluates the \texttt{Bandwidth} header sent by the client in an RTSP \texttt{OPTIONS}, \texttt{DESCRIBE}, \texttt{SETUP}, \texttt{SET_PARAMETER}, or \texttt{PLAY} request. This header is optional, and is used to indicate the estimated bandwidth available to the client in bits per second.
6. If the preceding options do not produce a viable initial rate, Helix Server uses the default media rate specified in the client’s user-agent settings. The variable DefaultMediaRate in the rate adaptation section provides this value.

**For More Information:** You cannot set DefaultMediaRate using Helix Administrator. You must set it in the user agent profile manually. Refer to the Helix Administrator online help topic [Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration](#).

7. If UseRel6InitialRateSelection is set to 0 (false) in the media player’s user agent settings, the last evaluated criterion is the stream.id in the RTSP SETUP request.

### Initial Oversend Rate

Once Helix Server selects the initial streaming rate, it may begin to stream the clip faster than the initial rate. The purpose is to fill the media player’s buffer quickly so that the player can start to render the stream as soon as possible. By default, Helix Server may transmit the initial stream at 100 to 400 percent of the actual streaming speed, depending on factors such as the media player capabilities, the stream bandwidth, the maximum sending rate, and so on.

**For More Information:** The Helix Administrator online help explains the maximum, initial oversend rate selected when you choose a default streaming profile using Helix Administrator. You can also set the InitialOversendRate variable directly in the user agent settings. For more on this, refer to the Helix Administrator online help.

### Receiver and Sender Reports

For server-side rate control to function accurately, Helix Server must receive periodic updates from the media player. These updates, or **receiver reports**, allow Helix Server to synchronize its buffer model with the actual buffer state experienced by the client. Using this information, Helix Server changes the streaming rate as necessary to compensate for network congestion.

**Note:** The Helix rate adaptation method uses a proprietary feedback mechanism implemented in the RDT packet format.
Information in the following sections pertains only to the 3GPP rate adaptation method, which is based on the RTP packet format.


Report Frequency

For rate control to function, a media player must send RTCP receiver reports to Helix Server. The report frequency is typically less than one second but may be up to five seconds. The shorter the reporting interval, the greater Helix Server’s ability to adjust streaming rates in a timely manner.

Receiver Report Rate

To send receiver reports at an appropriate interval, the media player should recognize the b=RR directive set in the SDP file. This directive determines the frequency of receiver reports. The default action is to request the client to reserve the amount of bandwidth equal to two percent (2%) of the initial streaming rate. Helix Server can also request a set amount of bandwidth regardless of the initial streaming rate.

Suppose that the initial streaming rate is 32 Kbps and the receiver report rate is two percent. Helix Server requests the media player to use 0.64 Kbps \((32 \times 0.02)\) of the connection bandwidth for receiver reports. This gives the player the ability to transmit 80 bytes of feedback per second. If a receiver report is 160 bytes in size, the receiver report frequency is one feedback report every two seconds.

When receiver report rates are defined as percentages of initial bandwidth, the bandwidth reserved for the reports increases along with the initial streaming rate. This means that higher-bandwidth clips report client feedback more frequently. This action is appropriate because, at higher bandwidths, the media player consumes data from its buffer faster, requiring that Helix Server compensate for network congestion more quickly.

Tip: After the bandwidth reserved for receiver reports has been set, it does not change if the streaming rate shifts up or down during the session.

For More Information: As explained in the online help, Helix Administrator allows you to select a streaming profile that sets
a receiver report bandwidth. You can also set a value manually by changing the \texttt{RTCPRRrate} variable in the user agent settings.

**Sender Report Rate**

Helix Server also sends the media player sender reports. The report rate is set by the \texttt{b=RS} directive set in the SDP file. The default is to reserve for sender reports the amount of bandwidth equal to one percent (1\%) of the initial streaming rate. Helix Server can also request a set amount of bandwidth regardless of the initial streaming rate. As with receiver reports, the reserved bandwidth does not change during the session.

*For More Information:* The online help lists the sender report bandwidths that you can select using Helix Administrator. You can also set a value manually by changing the \texttt{RTCPRSrate} variable in the user agent settings.

**NADU Feedback**

Helix Server supports the inclusion of NADU (Next Application Data Unit) information in receiver reports. The NADU signaling method provides feedback on the client buffer state. Helix Server uses this information to keep its server-side model of the player’s buffer accurate. By default, Helix Server requests media players that support NADU to include NADU information in each receiver report.

*For More Information:* You can decrease the NADU reporting frequency by changing the \texttt{NADUReportFreq} value in the Helix Server configuration file. This variable sets a global value that affects all media players. For details, refer to the Helix Administrator online help topic \texttt{Configuration File > Server-Side Rate Control Configuration > Stream Adaptation Configuration}.

**Receiver Report Timeout**

If ten seconds elapse and a new receiver report has not arrived, Helix Server disables rate control for the media player. It immediately shifts to the lowest streaming rate, and does not resume rate control even if a new receiver report arrives. The downshift to the lowest rate occurs even if you have configured downshifting to occur in stages, as described in the section “Stepwise Downshifting” on page 133.
You cannot change the timeout behavior using Helix Administrator. However, you can modify it using variables in the media player’s user agent settings:

- The `FeedbackTimeout` variable sets the feedback timeout period in milliseconds.
- You can disable downshifting to the lowest media rate by setting the value of the `DownshiftOnFeedbackTimeout` variable to 0. In this case, Helix Server continues to stream at the current rate when a receiver report timeout occurs.

For More Information: Refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Rate Control Configuration.

Buffer Modeling

To implement server-side rate control, Helix Server maintains a model of each player’s data buffer. This allows it to track throughout the streaming session the amount of data the player has available to render. Using receiver reports, it constantly updates its model to synchronize it with the player’s actual state. Helix Server can then upshift or downshift the streaming rates as needed. The following sections explain the basics of how Helix Server models a media player’s buffer.

Data Size Versus Playback Time

The amount of data in a media player’s buffer is called its buffer depth. There are two ways to measure buffer depth: bits of data or milliseconds of playback time. Correspondingly, Helix Server allows you to set parameters that affect buffer modeling in terms of the amount of data in the buffer, or the length of the clip playback time that’s buffered.

Because a media player’s data buffer has a fixed size, it is natural to think of buffering values in bits and bytes. However, this may not be advantageous for certain settings, such as a downshift depth (see “Downshifting” on page 131). Suppose that you set the buffer depth at which to downshift as 1/16 of a buffer capacity of 256 Kilobytes (KB). This means that a downshift occurs when the buffer drops to 16 KB of data.

For a 128 Kilobit-per-second (Kbps) media stream, 16 KB represents one second of buffered playback time. In this case, it is a good idea to downshift to
a slower streaming speed. For a 32 Kbps stream, however, 16 KB is four seconds of buffered playback time. At a relatively slow streaming rate of 32 Kbps with four seconds of buffered playback, downshifting is not necessary. Alternatively, you could set the downshift point at 2000 milliseconds. This means that the presentation downshifts when the buffer contains no more than two seconds worth of media. For a 128 Kbps stream, this is 32 KB of data. For a 32 Kbps stream, this is 8 KB of data.

Using millisecond values for certain buffer depths such as the upshift point may cause the inverse problem. Suppose that you set the upshift point at 8 seconds. For a 256 Kbps stream, the upshift occurs only when the 256 KB buffer fills to its maximum capacity. Although the upshift will still occur, the quality of playback will be degraded. Setting an upshift point as a percentage value makes more sense in this case.

**Buffer Size**

Each media player has a maximum buffer size typically defined in its RDF file. Or, you can use Helix Administrator to set this size directly in the UAS file as the value for **Video Predecode Buffer Size**. By default, Helix Server sets a limit on buffer usage as 98 percent of the maximum capacity. It automatically limits some other buffer variables to this usage limit. For example, if the max advance level exceeds the buffer usage limit, Helix Server automatically resets the max advance level to be equal to the usage limit.

**For More Information:** You can change the usage limit value by setting the BufferUsageLimit variable in the user agent settings. For details, refer to the Helix Administrator online help topic **Configuration File > Server-Side Rate Control Configuration > Client Capabilities Configuration.**

**Preroll**

Before a media player begins to render a stream, it buffers a certain amount of the stream data. This initial buffering, called **preroll**, may vary from clip to clip. The following figure illustrates a media player’s buffer filling with data until the necessary preroll depth is reached. At this point, the player begins to render the stream.
Preroll

When modeling the player’s buffer, Helix Server assumes that playback begins once the preroll has been met. However, as the following sections explain, certain media players may start playback at a point other than the clip’s standard preroll value.

Encoded Preroll Value

The most commonly observed preroll is a value encoded directly into the clip. The codec used to create the clip sets this preroll value, which is usually measured in milliseconds. Many media players respect the preroll value, buffering data until the preroll level is reached. Because preroll can vary from clip to clip, the initial buffering period may change for a media player depending on the content it plays.

Minimum Preroll In SDP

A media player’s user agent settings can include a MinPreroll variable that specifies a preroll value in milliseconds. If this variable is present, Helix Server advertises this minimum preroll value in the SDP file delivered to the player. If the player respects the SDP preroll value, it uses the larger of the two preroll values. That is, if the minimum preroll value advertised in the SDP file is larger than the encoded preroll value, the SDP value is used, and vice versa.

Note: Not all media players observe the preroll value in the SDP. RealNetworks and Helix-based players respect the SDP-derived preroll. Many 3GPP players do not, however. Be sure that you know the player characteristics before setting this variable.
For More Information: The Helix Administrator online help topic Helix Administrator > Server Setup > Media Delivery (Rate Control) > Default Profile and Preset Values explains the minimum preroll values set when you choose a default streaming profile using Helix Administrator. You can also set the MinPreroll value manually in the user agent settings.

Client-Specified Buffer Period

Some media players buffer a specific number of milliseconds for each stream as the preroll, regardless of the preroll value encoded in the stream or advertised in the SDP. In this case, you can set the ClientBufferPeriod variable in the user agent settings to match the player’s behavior. Helix Server then uses this value in its buffer model to estimate when the player begins media playback.

For More Information: You must set the ClientBufferPeriod value manually in the user agent settings. For instructions, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Client Capabilities Configuration.

Target Time

Media players that support Helix rate adaptation or 3GPP rate adaptation provide Helix Server with a Target-Time value. This is the number of milliseconds of stream playback time that the media player prefers to have buffered until playback ends. The target time can differ for each media player. It may be above or even below a clip’s encoded preroll time. The following figure illustrates Helix Server maintaining a player’s buffer at a target time level that is higher than the clip’s preroll requirement.
No user agent variables directly affect the target time, which is determined entirely by each media player. Various user agent variables, however, can set the stream upshift and downshift points relative to the reported target time. If a media player does not report a target time, Helix Server attempts to maintain the buffer at the preroll level or higher. It then bases the upshift and downshift points on the preroll value.

**Tip:** Because the client-specific Target-Time value reflects the media player’s buffering capability better than a stream-specific preroll value, stream shifting based on Target-Time is likely to be more accurate than that based on media preroll.

### Downshifting

When a clip is encoded at multiple bandwidths, Helix Server can *downshift* to a stream encoded at a lower bandwidth (less data per second). This occurs when the media player’s buffer begins to drain. A low buffer indicates that the player is consuming packets faster than they are being delivered by the network.

Downshifting causes the player to consume data from its buffer at a slower pace, thereby allowing the buffer to refill by bringing the rate of packet consumption in line with the rate of packet delivery. This, in turn, helps prevent rebuffering, an undesirable condition in which the player must halt the presentation to refill an empty buffer.

If the network conditions that necessitated the downshift clear, the player’s buffer may fill to a level that allows Helix Server to upshift back to the original streaming rate, or even to a higher streaming rate. The following figure
illustrates network congestion that causes packets to be delivered more slowly than they are rendered. Once the player buffer reaches its downshift depth, Helix Server switches the stream to a slower encoding.

**Downshift Depth**

![Diagram showing network congestion and downshift depth](image)

**Downshift Depth Settings**

Helix Administrator provides two separate fields in which you can specify the downshift depth. The two values affect different types of media players.

**Tip:** It is OK to define values for both **Downshift Depth** and **Target-Time Downshift Depth** in the same profile. Helix Server automatically chooses which value to use based on the media player’s capabilities.

**Downshift Depth**

The **Downshift Depth** value works with 3GPP Release 5 clients, RealPlayer 10 and earlier, and older Helix-based media players. It defines a downshift depth value based on the media preroll, which is encoded in the stream and may vary for each stream. The default depth is 80 percent of the preroll value. You can also set the downshift depth as a certain number of milliseconds of buffered media playback, such as 1000 milliseconds. In this case, the clip’s preroll value has no effect on downshifting.

**Note:** You can set this downshift depth using Helix Administrator. Or, you can change the value of the DownshiftDepth variable manually in the player’s user agent settings as described in the Helix Administrator online help.
Target-Time Downshift Depth

The Target-Time Downshift Depth value applies to 3GPP Release 6 clients, RealPlayer 11 and later, and newer Helix-based media players. It sets the downshift depth as a percentage of the Target-Time value reported by the media player. The default value is 80 percent of Target-Time. You can also set this downshift depth to a certain number of milliseconds of media buffered, such as 1000 milliseconds. In this case, the reported Target-Time value has no effect on downshifting.

Note: You can set this downshift depth using Helix Administrator. Or, you can change the value of the TargetTimeDownshiftDepth variable manually in the player’s user agent settings as described in the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Client Adaptation Configuration.

Stepwise Downshifting

Stepwise downshifting allows Helix Server to step down through each encoded rate in the clip. This feature is disabled by default, which causes Helix Server to downshift immediately to the most appropriate bit rate, bypassing any intermediate streaming speeds. The default behavior sacrifices the higher quality of any intermediate bandwidths in an attempt to ensure that the media player’s buffer does not drain.

You may want to enable stepwise downshifting if your network provides relatively stable bandwidth for each client connection. In this case, Helix Server may go through a longer cycle of downshifting to reach the highest sustainable speed. This keeps the quality of the rendered media as high as possible on each downshift. However, this also increases the chance that the media player may need to rebuffer the presentation if the available bandwidth drops significantly.

For More Information: To enable stepwise downshifting, change the value of the EnableStepwiseDownshift variable in the media player’s user agent settings to 1. For more information, refer to the Helix Administrator online help topic Configuration File >
Server-Side Rate Control Configuration > Rate Adaptation Configuration.

**Note:** Helix Server waits a certain time between successive rate shifts. The Helix Administrator online help topic Helix Administrator > Server Setup > Media Delivery (Rate Control) > Default Profile and Preset Values explains the stream switch delay selected when you choose a default streaming profile. You can also modify the value of the StreamSwitchDelay variable in the player’s user agent settings.

**Client-Initiated Downshift**

If a media player supports the RTSP 3GPP-Link-Char or Bandwidth header, it can send information about its network link characteristics to Helix Server over the RTSP channel. By default, Helix Server allows the media player to initiate a downshift in the streaming rate by using one of these protocol headers. This allows downshifting to occur immediately, without Helix Server having to wait for the next client receiver report.

**For More Information:** You can turn off support for this feature by setting the EnableClientReportedDownshift variable in the player’s user agent settings to 0. For more information, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Client Adaptation Configuration.

**Streaming Below the Lowest Media Rate**

By default, the lowest rate at which Helix Server transmits data to the media player is the lowest streaming rate encoded in the clip. However, you can set Helix Server to deliver the slowest stream at a rate even slower than its encoded speed. This allows Helix Server to transmit data to the player during periods of extremely high network congestion. Although the player will most likely rebuffer the stream once the congestion clears, it may have received enough data to shorten the rebuffering period.

**For More Information:** To enable streaming below the lowest encoded rate, set the value of the DeliverBelowLowestMediaRate variable in the media player’s user agent settings to 1. For details, refer to the Helix Administrator online help topic.
Upshifting

*Upshifting* means that Helix Server switches to a media stream that is encoded at a higher bandwidth (more data per second). This occurs when the media player’s buffer begins to fill. A nearly full buffer indicates that the player is consuming packets from its buffer slower than they are being delivered.

Upshifting causes the player to consume data from its buffer at a faster pace. This helps to synchronize packet consumption from the buffer with packet delivery by the network. By upshifting to a higher data rate, Helix Server delivers a higher-quality user experience, and prevents data packets from being lost because the media player’s buffer was too full to accept them.

**Upshift Depth**

**Upshift Depth Settings**

Helix Administrator provides two separate fields in which you can specify the upshift depth. The two values affect different types of media players.

*Tip:* It is OK to define values for both **Upshift Depth** and **Target-Time Upshift Depth** in the same profile. Helix Server automatically chooses which values to use based on the media player capabilities.
Upshift Depth

The **Upshift Depth** value works with 3GPP Release 5 clients, RealPlayer 10 and earlier, and older Helix-based media players. It defines an upshift depth based on the media preroll, which is encoded in the stream and may vary for each stream. The default depth is 400 percent of the preroll value. You can also set the upshift depth as a certain number of milliseconds of buffered media playback, such as 4000 milliseconds. In this case, the clip preroll has no effect on upshifting.

**Note:** You can set this upshift depth using Helix Administrator. Or, you can change the value of the `UpshiftDepth` variable manually in the player’s user agent settings.

Target-Time Upshift Depth

The **Target-Time Upshift Depth** value applies to 3GPP Release 6 clients, RealPlayer 11 and later, and newer Helix-based media players. It sets the upshift depth as a percentage of the Target-Time value reported by the media player. The default value is 400 percent of Target-Time. You can also set the upshift depth as a certain number of milliseconds of buffered media playback, such as 4000 milliseconds. In this case, the reported Target-Time value has no effect on upshifting.

**Note:** You can set this upshift depth using Helix Administrator. Or, you can change the value of the `TargetTimeUpshiftDepth` variable manually in the player’s user agent settings.

Stepwise Upshifting

Stepwise upshifting causes Helix Server to step up in sequence through each encoded rate in the clip until it finds the maximum sustainable rate. This feature is enabled by default, causing Helix Server to upshift more gradually through the intermediate streaming speeds. The default behavior produces conservative upshifting that attempts to ensure that each upshift is sustainable.

**For More Information:** You can disable stepwise upshifting by changing the value of the `EnableStepwiseUpshift` variable in the player’s user agent settings to 0. For details, refer to the Helix
CHAPTER 8: Rate Control

Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration.

Note: Helix Server waits a certain time between successive rate shifts. The Helix Administrator online help explains the stream switch delay selected when you choose a default streaming profile. You can also set the value of the StreamSwitchDelay variable in the player’s user agent settings as described in the online help topic Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration.

Client-Initiated Upshift

Media players that support the RTSP 3GPP-Link-Char or Bandwidth header can send information about network link characteristics to Helix Server over the RTSP channel. By default, Helix Server allows the media player to initiate an upshift in the streaming rate by using one of these protocol headers. This allows upshifting to occur immediately, without Helix Server having to wait for two consecutive receiver reports.

For More Information: You can turn off support for client-initiated upshifting by setting the EnableClientReportedUpshift variable in the player’s user agent settings to 0. For details, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Client Adaptation Configuration.

Slowdown Mode

Helix Server can use a feature called slowdown mode to help ensure that a media player’s buffer does not overflow on an upshift. The user agent settings that control slowdown mode typically do not need to be changed from their default values. However, you need to enable slowdown mode, and may want to change the max advance level if a media player frequently experiences packet loss during upshifting.

For More Information: You enable slowdown mode by setting the value of the RateManagerType variable in the player’s user agent settings to unblocking. For details, refer to the Helix
Slowdown Mode During Upshifting

During upshifting, Helix Server attempts not to overload a media player’s buffer. For example, suppose that a media player is receiving a 32 Kbps stream. Network conditions are good and the player’s buffer has filled to its upshift depth, which equals six seconds of playback time. Once the upshift depth is reached, Helix Server switches to a 64 Kbps encoding of the stream. This upshift may overload the player’s buffer, however.

Suppose that the buffer limit is 10 seconds of data at 32 Kbps. The buffer already contains six seconds of the 32 Kbps stream yet to play. This leaves four seconds at 32 Kbps, or only two seconds of available buffer space at 64 Kbps. At the faster data rate, the buffer may overflow with new 64 Kbps data before the player has been able to drain the existing 32 Kbps data. If this happens, some 64 Kbps packets may be lost to the network, lessening the quality of the playback when the player starts to render the 64 Kbps stream.

Max Advance

The max advance value defines a buffer level that is usually above the upshift depth but lower than the buffer limit. By default, it is the buffer usage limit. If Helix Server determines that the buffer has reached the max advance level, it slows the amount of data it transmits to allow the buffer to drain. The following figure illustrates a player reaching its max advance state and entering slowdown mode.

Max Advance Level
Warning! The max advance level should always be higher than clip preroll or the player’s Target-Time value. The max advance level must also be higher than the upshift depth for upshifting to occur.

For More Information: You can change the max advance point by setting the MaxAdvance variable in the user agent settings. For details, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration.

Limiting Stream Buffering with Max Advance

Helix Server always enters slowdown mode when a media player’s buffer reaches its max advance level. This occurs even if upshifting is not possible because the highest-rate stream is already playing, or the clip is encoded at a single rate. Because you set a max advance value in milliseconds, you can use max advance to cause a media player to buffer only a certain amount of media playback time.

As an example of why you would do this, consider the channel switching feature described in Chapter 14. This feature allows you to switch to a stream from a different clip within a single RTSP session, thereby eliminating buffering during the switchover. If you set a player’s max advance level to 3000 milliseconds, for example, you cause the media player to buffer approximately three seconds of its current stream. This allows the new channel stream to start playing three seconds after the switch command is given.

Slowdown Mode Exit

When it enters slowdown mode, Helix Server drops the data transmission rate to approximately 70 percent of the initial stream’s natural streaming rate. This increases empty buffer space by allowing the buffer to drain to a level defined as the slowdown mode exit. Once the exit point is reached, data transmission resumes at the upshifted streaming rate.

By default, the exit point is the Target-Time value or the content preroll, whichever the media player uses. You can change the slowdown mode exit point to any value, however. Typically, the exit point falls between the values for the upshift depth and the downshift depth. The following figure illustrates reduced data transmission until the buffer reaches the slowdown mode exit point.
Slowdown Mode Exit

**For More Information:** You can change the exit point by setting the `SlowdownModeExit` variable in the media player’s user agent settings. For details, refer to the Helix Administrator online help topic [Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration](#).

**Tip:** Once it leaves slowdown mode, Helix Server waits for two consecutive receiver reports that indicate an upshift is possible. You can enforce a more conservative upshift behavior by increasing the value of the `RequiredUpshiftConfirmations` variable in the media player’s user agent settings. For details, refer to the Helix Administrator online help topic [Configuration File > Server-Side Rate Control Configuration > Rate Adaptation Configuration](#).

### Congestion Control

Helix Server can employ different methods of congestion control. These methods allow it to deliver a steady amount of data to a media player by compensating for various fluctuations in available bandwidth. Rate shifting between different streams in a clip occurs only on a persistent change in network bandwidth that exceeds the abilities of the congestion control mechanism. The default congestion control mechanism is TFRC.

**For More Information:** You can change the congestion control mechanism using the `UDPCongestionControlType` variable in the media player’s user agent settings. The Helix Administrator
online help topic Helix Administrator > Server Setup > Media Delivery (Rate Control) > BCC explains how to set BCC as a profile’s congestion control mechanism.

TFRC

TCP Friendly Rate Control (TFRC) is the default means of congestion control when the UDP transport is used. It poses a problem, however, for some networks that contain large network buffers (around one Megabyte or larger). On such networks, TFRC may cause large round trip times, negatively affecting throughput.

For More Information: For information about TFRC variables, refer to the TFRC section in Helix Administrator online help. For more about TFRC, refer to IETF RFC 3448 at http://www.ietf.org/rfc/rfc3448.txt.

BCC

Binomial Congestion Control (BCC) can be used on networks with large buffers along the link path. It insures that correct congestion control response is maintained while not causing the round-trip time to increase to excessive levels. You should not choose BCC without first performing diagnostics on the network to determine if this type of congestion control is appropriate. BCC works only with the UDP transport.

For More Information: For information about BCC variables, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > BCC Configuration. You can learn more about BCC at http://nms.lcs.mit.edu/papers/binomial-infocom01.html.

TCP

If a media stream is delivered using TCP rather than UDP, Helix Server relies on the native TCP congestion control mechanism, which is built into the TCP/IP network layer. The TCP transport is available to any media player that prefers it.

Note: If the user agent variable PreferTCP is set to 1, Helix Server uses TCP for media players that advertise support for the TCP transport.
For More Information: For information about TCP variables, refer to the Helix Administrator online help topic Configuration File > Server-Side Rate Control Configuration > TFRC Configuration.

Defining Rate Control Profiles

Using Helix Administrator, you can configure rate control profiles for various media players. You can create the profile in the Helix Server configuration file, or store it as a separate user agent file. For each profile, you can set the major rate control variables, such as those that control the upshift and downshift depths.

In Helix Administrator, click Server Setup > Media Delivery (on Helix Proxy, click Proxy Setup > Media Delivery). Refer to the online help for information about performing the following tasks:

- Enable advertisement of the Helix rate adaptation and 3GPP rate adaptation methods to all media players.
- Create a new profile for a media client
- Assign the profile to user-agent strings.
- Set the rate control method used by the client profile for on-demand clips and live broadcasts.
- Choose a preset value (Conservative, Moderate, or Aggressive) for how the media player responds to bandwidth changes.
- Set the client’s upshift depth and downshift depth.
- Set the top streaming bit rate for this profile.
- Enable TFRC or BCC for the profile.
This chapter explains server-side playlist management, which streams a sequence of clips or broadcasts to RTSP-based or iOS media players. For certain types of playlists sent to RTSP media players, the viewer can instruct Helix Server to skip to different playlist entries.

Understanding Playlist Management

To create a playlist, a viewer interacts with a third-party Web portal to define a sequence of on-demand clips, live broadcasts, or simulated-live broadcasts. The portal creates a text file (extension .hpl) that uses the XML-based SMIL markup to define the playlist features. It then uploads the playlist to Helix Server. When the viewer requests the playlist, the media selections play sequentially.

Playlist of Four Clips Streaming Sequentially

The preceding figure illustrates a set of four on-demand clips lasting three minutes apiece. The entire presentation appears to the media player as a single, 12-minute session. RTSP media players receive the content in a single RTSP session while iOS devices maintain a persistent HTTP connection. As
the playlist streams, transitions between the clips occur seamlessly, without the need for rebuffering.

**Types of Playlist Sessions**

Helix Server supports the following types of playlist management sessions:

- **externally controlled (RTSP media players only)**
  
  In this type of session, the viewer can skip to various parts of the playlist by sending HTTP directives to Helix Server outside of the RTSP connection.

- **internally controlled (RTSP media players and iOS devices)**
  
  An internally controlled playlist appears to the viewer as a single on-demand clip. The viewer can seek through the playlist using the media player’s seek commands.

- **noncontrolled (RTSP media players and iOS devices)**
  
  A noncontrolled playlist session appears to be a single live stream. The viewer cannot skip or seek through the contents.

**Externally Controlled Session**

Available with RTSP media players, an externally controlled session appears to be a single live stream even if the playlist contains no live broadcasts. The media player’s timeline controls don’t function, and RTSP seeking is disabled. However, the viewer can issue seek and skip commands outside of the RTSP channel by clicking HTTP links that are proxied to Helix Server by a third-party Web server. Playlist control is therefore external to the RTSP channel.

To support an externally controlled session, a playlist must include certain features such as chapter markers that designate the allowed skip points. The RTSP URL used to request the playlist must also contain specific query string parameters, such as an ID value that allows Helix Server to identify the RTSP session referred to in HTTP-based skip directives.

**For More Information:** See “Externally Controlled Playlist Sessions” on page 154.

**Internally Controlled Session**

In an internally controlled playlist session, a viewer using an RTSP media player or an iOS device can seek through the playlist by issuing seek
commands directly to Helix Server. Hence, playlist control is **internal** to the media stream. External, HTTP-based directives sent by a third-party Web server are not supported. For a playlist to be internally controllable, it must consist entirely of on-demand media streams that have defined durations. This gives the playlist an overall timeline and allows it to function like a single clip.

**Tip:** Playlists for internally controlled sessions do not require features such as chapter markers. However, a playlist designed for externally controlled sessions will work for internally controlled or noncontrolled sessions. Helix Server sets the session type based on the playlist contents and the elements in the request URL. It ignores any playlist components not supported by the session type.

**For More Information:** See “Internally Controlled and Noncontrolled Playlist Sessions” on page 151.

**Noncontrolled Session**

In a noncontrolled playlist session, an RTSP media player or iOS device treats the entire session as if it were a single, live stream. No seek commands are allowed. The viewer does not have control over the timeline other than to start and stop the playlist session.

The playlist components determine if the session is internally controlled or noncontrolled. A noncontrolled session is used whenever Helix Server cannot calculate the overall playlist timeline, which is necessary for it to manage seeking through a playlist. Including any of the following in the playlist results in a noncontrolled session:

- **live or simulated-live broadcast**

  A playlist that includes at least one live or simulated-live broadcast functions as a noncontrolled session because the indeterminate timeline of a broadcast prevents Helix Server from calculating an overall playlist timeline.

- **undefined clip duration**

  If one or more on-demand clips do not have durations defined in the playlist, Helix Server cannot calculate seek times across clips. It therefore streams the playlist contents in a noncontrolled session.
• non-skippable clip

If at least one clip in the playlist has been declared non-skippable (see page 170), Helix Server streams the playlist contents in a noncontrolled session. This default action can be modified, however.

Web Portal Requirements

The Web portal is a third-party application that gives users access to playlists. It may provide standard playlists for all users or allow each user to create a customized playlist. The portal submits playlists to Helix Server according to the workflow described in the section “Initial Playlist Selection” on page 154.

Note: The user builds the playlist and issues skip directives by interacting with the Web portal. The playlist management feature does not set any specific browser requirements.

Basic Portal Requirements

Minimally, the Web portal must be able to do the following:

• Process any necessary registration and billing information required from the viewer.

• Configure playlists in the XML-based SMIL format described as the section “Playlist Format” on page 163.

• Upload playlists to the required Helix Server mount point as described in the section “Uploading Playlists” on page 174.

Requirements for Managing Externally Controlled Playlists

To implement externally controlled playlist sessions, the portal must also do the following:

• Issue HTTP or HTTPS directives to the Helix Server controller port as described in the section “Externally Controlled Playlist Sessions” on page 154.

• Add required parameters to an SDP file returned by Helix Server as explained in the section “Initial Playlist Selection” on page 154.

Stream Requirements for RTSP Players

Any media player that is compliant with the RTSP standard and that supports the chosen streaming format should be able to render a playlist session.
Because an RTSP player cannot switch its decoding method within an RTSP session, all streams within a single playlist must be encoded using the same streaming rates and the same codec (same profile and level).

**Tip:** Because stream quality can differ across encoders, creating content using the same hardware or software encoder helps to prevent streaming errors.

**Supported Codecs**

Playlist management for RTSP players works with the following codecs.

**Video Codecs**
- H.264 (recommended)
- H.263
- MPEG-4
- RealVideo

**Audio Codecs**
- AAC or AAC+ (recommended)
- Enhanced AAC+
- AMR-NB or AMR-WB
- RealAudio

**Supported File Formats**

Supported file formats for RTSP Players are the following:
- MPEG-4 (.mp4 and variants)
- 3GPP (.3gp)
- RealMedia (.ra, .rv, .rm)
- QuickTime (.mov)
- F4V (.f4v)

**Tip:** Playlist management does not function with Flash clients, which use the RTMP protocol. However, you can use the F4V format when streaming to RTSP clients that support the H.264 and AAC codecs used to compress the media.
Unsupported Formats

The following formats are **not** supported:

- Windows Media and the MMS streaming protocol
- Flash FLV clips (.flv)
- *any* format that use digital rights management (DRM) protection

Protocols and Transports

Playlist management for RTSP players works with RTSP and RTSP streams cloaked as HTTP. The network transport can be either UDP or TCP. Playlist sessions support both standards-based RTP and the RealNetworks RDT data packet format, which is used with RealPlayer and Helix-based media players.

Segment Handling with iOS Devices

Playlist sessions for iOS devices can use the same .hpl playlist used with RTSP media players as long as the content is codec-compatible across all devices. As noted in the following sections, though, iOS devices handle live broadcasts and rate changes between clips differently than RTSP players. This can affect how playlists are structured.

**Note:** Playlist management supports iPhone, iPod Touch, and iPad devices running iOS 3.0 or later. Other devices that support HTTP live streaming in accordance with Apple specifications may also be able to use playlists.

**For More Information:** See Chapter 7 for information about how Helix Server generates transport segments and supports iOS devices. Media segmentation must be configured before you can deliver playlist sessions to iOS devices.

/m3ugen/ Mount Point

An iOS device **must** request the .hpl playlist using the /m3ugen/ mount point, which causes Helix Server to segment the resource and return an .m3u8 playlist.

**Note:** Only the request URL for the .hpl playlist uses the /m3ugen/ mount point. Within the .hpl playlist, media resources are designated by relative URLs that do **not** include the /m3ugen/ mount point.
For More Information: For .hpl playlist examples, see “Specifying the Media Source” on page 167. The section “Using the m3ugen Mount Point” on page 91 explains how to use /m3ugen/ with iOS device requests.

Codecs and Rate Handling

Content for iOS devices must be encoded as H.264 video and AAC audio. Unlike RTSP players, iOS devices can switch streaming rates within a playlist. For example, an iOS device can play one stream encoded at 100 Kbps and then play another stream at 200 Kbps. In an .m3u8 playlist, Helix Server indicates where bandwidth options change, allowing the iOS device to choose streaming options appropriately.

Tip: To make a single playlist accessible to both RTSP media players and iOS devices, use H.264/AAC encoding at the same bandwidth (or set of bandwidths), profile, and level for all media sources.

For More Information: The section “iOS Source Media Requirements” on page 86 provides more information about the encoding requirements for iOS devices.

Broadcasts

The .hpl playlist for an iOS device may contain one (and only one) link to a live or simulated live broadcast. Because the cessation of a broadcast stream closes the device’s persistent HTTP connection, the broadcast URL must be the last media resource in the playlist.

Note: When a live broadcast is used, the request URL must include a GUID that enables Helix Server to keep track of the session. See “Session ID” on page 161 for details about GUID values.

Segment Creation

Requesting a clip or live broadcast through an .hpl playlist does not alter the location or naming of segment files. If segments for song100.mp4 already exist on Helix Server, for example, those segments are used if song100.mp4 is requested as part of an .hpl playlist. If Helix Server segments song100.mp4 because of an .hpl request, those segments are used if an iOS device later requests song100.mp4 outside of a playlist management session.
For More Information: The section “File Names and Directory Structures” on page 102 explains the default location and naming conventions for segments and playlists.

Session Playlists

When an iOS device requests an .hpl file, Helix Server returns to the device a playlist or set of playlists in the .m3u8 format:

• An internally controlled session has an overall length calculated by Helix Server. The .m3u8 playlist returned to the iOS device lists all segments for all media sources, noting where encoding rates change.

For More Information: For an internally controlled session, Helix Server and iOS device interaction is similar to that described in “On-Demand Segmentation Steps” on page 88.

• A noncontrolled session does not have a known, overall length. Helix Server treats it as a live broadcast and sends the iOS device a series of .m3u8 playlists. Each playlist provides the URLs for three segments and indicates where encoding rates change.

For More Information: For a noncontrolled session, Helix Server and iOS device interaction is similar to that described in “Broadcast Segmentation Steps” on page 90.

The .m3u8 playlist or playlists generated from the .hpl playlist use the location and naming conventions described in the section “Playlist File” on page 103. As noted in the section “Segment Creation” on page 149, the media segments are not stored along with the master session playlist.

Compatibility with Other Features

The following table summarizes the compatibility of playlist management with other Helix Server features.

<table>
<thead>
<tr>
<th>Other Feature</th>
<th>Compatibility Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookmarking</td>
<td>Bookmarking works with internally controlled sessions that appear to be a single on-demand clip. For more on bookmarking, refer to the Helix Administrator online help.</td>
</tr>
</tbody>
</table>

(Table Page 1 of 2)
Internally Controlled and Noncontrolled Playlist Sessions

The set-up and management for internally controlled and noncontrolled sessions are virtually identical. The following sections explain the workflow for managing an internally controlled or noncontrolled playlist session for an RTSP player or iOS device. These types of sessions do not accept HTTP requests proxied through a third-party Web server.

Initial Playlist Selection

The following is the sequence of events that occurs when a user defines and requests a new playlist.
Requesting a Playlist (Internally Controlled or Noncontrolled Session)

To request a playlist in an internally controlled or noncontrolled session:

1. The user constructs the playlist using the Web portal. The portal handles all necessary authentication, financial, and data transactions.

   **Note:** If the .hpl playlist already exists on Helix Server, the next action is Step 5.

2. The Web portal formats the playlist as described in the section “Playlist Format” on page 163. It passes the playlist to Helix Server using an authenticated HTTP or HTTPS POST, which the section “Uploading Playlists” on page 174 describes.

3. Helix Server writes the .hpl playlist to disk.

4. On a successful upload, Helix Server responds to the portal with an HTTP 200 message. Otherwise, it issues an HTTP error as described in the section “Error Conditions” on page 176.

5. Using a browser, the viewer requests a playlist:
   - The user of an RTSP media player requests the .hpl file using the SDPgen utility of Helix Server.
• The HTTP request of an iOS device includes the /m3ugen/ mount point to segment the content and generate the .m3u8 playlists.

**For More Information:** For background on SDPgen, refer to the online help topic [Streaming Basics > Predefined Mount Points > Client Mount Points](#). The section “Using the m3ugen Mount Point” on page 91 explains how iOS devices request content.

6. Helix Server verifies that the playlist has not expired. It then returns an .sdp file to the RTSP player or an .m3u8 playlist to the iOS device. Optionally, the Web portal may proxy the .sdp file or .m3u8 playlist to the client and update the playlist request URL to include the following parameter:

   guid=ID  Sets a user-defined session ID that enables logging of the individual clips in the playlist. This is optional for RTSP players using an internally controlled session. It is required for iOS devices if the playlist contains a live broadcast. For details, refer to “Session ID” on page 161 and “Logging and Error Codes” on page 176.

   For example, Helix Server may return the following request URL in the SDP file sent to an RTSP player:

   rtsp://helixserver.example.com/playlists/songs.hpl

   The Web portal updates the URL to look like the following:

   rtsp://helixserver.example.com/playlists/songs.hpl?guid=e624acc1-fa91-11e0-f953-6b6ee78b776b

   **For More Information:** The playlist expiration time is defined within the playlist. See “Playlist Expiration” on page 164.

7. The user’s media player initiates the playlist session using the returned playlist URL. For RTSP devices, clips are delivered in sequence within the RTSP session. An iOS device receives a segment playlist (.m3u8) or a series of playlists.

8. Helix Server begins to stream the media resources listed in the playlist.

9. As the playlist resources stream, Helix Server generates separate entries for each clip or broadcast in its access log.
**Playlist Seeking**

The following figure illustrates playlist seeking in an internally controlled session, which defines an overall playlist length. While watching the first clip, the viewer uses the media player’s seek controls to seek ahead several minutes, skipping over the second clip entirely and resuming playback in the middle of the third clip.

**Seeking Through Clips in a Playlist**

![Seeking Through Clips in a Playlist](image)

**Tip:** Playlist seeking is disabled in noncontrolled sessions, which appear to the user to be live broadcasts with indeterminate playback lengths.

**Note:** A seek action cannot identify a chapter ID defined in the playlist or provide an external skip directive such as `seek=next`. These features are available only with externally controlled sessions.

**Externally Controlled Playlist Sessions**

The following sections explain the workflow in externally controlled playlist sessions available for RTSP media players. These sessions associate an ID with the RTSP session and can accept HTTP directives to modify the session by skipping or seeking to a different part of the playlist timeline.

**Initial Playlist Selection**

The following events occur when a user defines and requests a new playlist.
To request a playlist in an externally controlled session:

1. The user constructs the playlist using the Web portal. The portal handles all necessary authentication, financial, and data transactions.

   **Note:** If the .hpl playlist already exists on Helix Server, the next action is Step 5.

2. The Web portal formats the playlist as described in the section “Playlist Format” on page 163. It passes the playlist to Helix Server using an authenticated HTTP or HTTPS POST, which the section “Uploading Playlists” on page 174 describes.

3. Helix Server writes the playlist to disk.

4. On a successful upload, Helix Server responds to the portal with an HTTP 200 message. Otherwise, it issues an HTTP error as described in the section “Error Conditions” on page 176.

5. The Web portal requests the playlist as an SDP file (.sdp) using the Helix Server SDPgen utility.

**For More Information:** For background on SDPgen, refer to the online help topic Streaming Basics > Predefined Mount Points > Client Mount Points.
6. Helix Server verifies that the playlist has not expired. It then returns a standard SDP file to the Web portal.

**For More Information:** The expiration time is defined within the playlist. See “Playlist Expiration” on page 164.

7. The Web portal updates the playlist request URL within the SDP file to include the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hpl=1</td>
<td>Indicates that the RTSP session is subject to playlist control. See “Playlist Request Marker” on page 160.</td>
</tr>
<tr>
<td>guid=ID</td>
<td>Sets a user-defined session ID. Subsequent requests from the Web portal to modify the playlist session include this ID. Refer to “Session ID” on page 161.</td>
</tr>
<tr>
<td>mdp=1</td>
<td>Optional parameter that allows Helix Server to use its server-side rate control mechanism when streaming clips. See “Rate Control Flag” on page 162.</td>
</tr>
</tbody>
</table>

For example, Helix Server may return the following request URL in the SDP file:

```
rtsp://helixserver.example.com/playlists/songs.hpl
```

The Web portal updates the URL to look like the following:

```
rtsp://helixserver.example.com/playlists/songs.hpl?hpl=1&guid=e624acc1-fa91-11e0-f953-6b6ee78b776b
```

8. The Web portal delivers the SDP file to the media player.

9. The media player initiates the RTSP session using the playlist URL contained in the SDP file.

10. Helix Server begins to stream the media resources listed in the playlist. It identifies this RTSP session using the guid value from the request URL.

11. As the playlist streams, Helix Server generates entries in its access log.

**For More Information:** As explained in the section “Logging Styles and Variables” on page 176, log entries include the user-defined ID and list the sequence of each clip within the playlist.
Chapter Skipping

To support chapter skipping in an externally controlled session, the playlist organizes content into *chapters*. Each chapter is typically two or more clips that function as a group. For example, an advertisement and a movie preview may make up a chapter. If the viewer issues an HTTP-based skip command while watching the advertisement, Helix Server skips to the next chapter rather than directly to the movie preview.

**Tip:** The playlist may also designate clips as *non-skippable*. Directives to skip through these clips are not allowed. See “Making a Clip Non-Skippable” on page 170.

**For More Information:** The section “Setting Up Chapters” on page 170 explains how to organize clips into chapters and assign IDs used with skip directives.

Chapter Skipping Example

In the following figure, the playlist is organized into two chapters. Each chapter groups together two clips that play in sequence. While watching the first clip in chapter 1, the viewer issues a command to skip to the next chapter. This takes the viewer to the start of the third clip.

*Skipping Chapters in a Playlist*

Chapter Skipping Directives

Signalling for a chapter skip occurs outside of the RTSP session, typically through an HTTP or HTTPS GET requested directed toward the Web portal.
Process for Skipping to a Different Chapter

To skip to a different position in the playlist:

1. Using a browser, the session viewer requests a jump to a different position in the playlist.

   Tip: As depicted above, the HTTP or HTTPS command can go through the Web portal, which then forwards the commands to Helix Server. Alternatively, the link commands may go directly to Helix Server.

   Note: The media player does not receive the playlist from Helix Server. To allow skipping, the Web portal must construct an interactive Web page from the playlist values and allow the user to indicate the desired skip points.

2. The session viewer’s skip request is translated into an HTTP or HTTPS GET command sent to the Helix Server playlist control port. To the original request URL, the HTTP or HTTPS directive adds the following required query string parameters:

   | hpl=1 | Identifies the request as a playlist session directive. See “Playlist Request Marker” on page 160. |
   | guid=ID | Provides the user-defined ID assigned to the session when the playlist session was initialized. See “Session ID” on page 161. |
The request also includes one or more of the following parameters, which indicate where to jump in the playlist:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seek=point</td>
<td>Indicates a skip point, such as the next chapter. See “Seek” on page 161.</td>
</tr>
<tr>
<td>plref=ID</td>
<td>Identifies a chapter ID within the playlist as the skip point. See “Playlist Reference” on page 162.</td>
</tr>
<tr>
<td>time=ms</td>
<td>Provides a timing offset for the skip. See “Time” on page 162.</td>
</tr>
</tbody>
</table>

For example, an HTTP request directed toward the Helix Server playlist control port may look like the following:

http://helixserver.example.com:8009/playlists/songs.hpl?hpl=1&guid=e624acc1-fa91-11e0-f953-6b6ee78b776b&seek=next

3. Helix Server returns an HTTP response indicating success or failure.

   **For More Information:** See “Error Conditions” on page 176.

4. On a success, Helix Server skips to the designated position in the playlist timeline.

   **Note:** The skip typically causes rebuffering in the viewer’s player as the new data streams. Once this preroll requirement is fulfilled, however, streaming continues without rebuffering during normal network conditions.

5. Helix Server records the action in its log file.

   **For More Information:** See “Logging Styles and Variables” on page 176.

**Playlist Seeking**

In an externally controlled session, the playlist always appears to be a live broadcast, preventing the media player from seeking through the timeline using RTSP commands. However, the viewer can interact with the Web portal to indicate a seek request. The Web portal then issues an HTTP or HTTPS directive to Helix Server. The directive includes a time parameter to specify the desired position in the overall playlist timeline. For example:

http://helixserver.example.com:8009/playlists/songs.hpl?hpl=1&guid=e624acc1-fa91-11e0-f953-6b6ee78b776b&time=321400

   **For More Information:** See “Time” on page 162.
Playlist Control Parameters

For externally controlled playlists, requests to start the playlist or jump to a different position include query string parameters that are summarized in the following table. The last two table columns indicate if the parameter is required in the initial RTSP request and any HTTP control directives.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
<th>Example</th>
<th>RTSP Request</th>
<th>HTTP Directive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playlist Request Marker</td>
<td>Indicates that the request pertains to a playlist.</td>
<td>hpl=1</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>Session ID</td>
<td>Identifies the playlist RTSP session.</td>
<td>guid=ID</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>Seek</td>
<td>Specifies a different chapter to play:</td>
<td>seek=next</td>
<td>not used</td>
<td>optional</td>
</tr>
<tr>
<td></td>
<td>next – next chapter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>prev– previous chapter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>first – first chapter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Playlist Reference</td>
<td>Provides the ID of a playlist chapter to play.</td>
<td>plref=341</td>
<td>not used</td>
<td>optional</td>
</tr>
<tr>
<td>Time</td>
<td>Indicates the time in milliseconds at which to start playback.</td>
<td>time=213400</td>
<td>not used</td>
<td>optional</td>
</tr>
<tr>
<td>Rate Control Flag</td>
<td>Enables server-side rate control:</td>
<td>mdp=0</td>
<td>optional</td>
<td>not used</td>
</tr>
<tr>
<td></td>
<td>0 – rate control disabled</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 – rate control enabled (default)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tip: The order of query string parameters in a request URL does not matter.

For More Information: You can change the parameter names, such as hpl, through the Helix Server configuration file. See the Helix Administrator online help topic [Configuration File > Content Management Configuration > Server-Side Playlist Configuration](#).

Playlist Request Marker

The parameter hpl=1 indicates that the HTTP or RTSP request is for a playlist session. It must be included in all requests to start or modify an externally controlled playlist session.
Session ID

The guid parameter provides a user-defined value that identifies the playlist session. The Web portal is responsible for generating a unique value for each user. It must include this parameter and value pair in each playlist request to enable Helix Server to handle the session. The guid value should be 8 to 32 characters in length, and may contain any of the following characters:

- a-z
- A-Z
- 0-9
- - (hyphen)

Seek

You can include the seek parameter with HTTP-based skip commands to instruct Helix Server to skip to a different clip within a playlist. Clips do not need to have IDs explicitly defined in the playlist for the seek parameter to work. Valid values are the following:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>first</td>
<td>Skip to the first clip in the playlist.</td>
</tr>
<tr>
<td>next</td>
<td>Skip to the next clip or defined skip-point in the playlist.</td>
</tr>
<tr>
<td>prev</td>
<td>Skip to the previous clip or the beginning of the current clip. (See below.)</td>
</tr>
</tbody>
</table>

Note: If either the plref or time parameter is also present in the URL, Helix Server ignores the seek value.

For More Information: See the section “Setting the Skip Point” on page 169 for more about skip-point markers.

Seeking to the Previous Clip

By default, seek=prev skips back to the start of the current clip if that clip is playing at greater than the 10-second mark in its timeline. If the clip is playing at less than its 10-second mark when Helix Server receives the seek=prev directive, Helix Server skips to the start of the preceding clip.

To summarize:

- Current clip timeline < 10 seconds = skip to the previous clip
- Current clip timeline > 10 seconds = skip to the start of the current clip

Tip: You can change the default value from 10 seconds by editing the PreviousSeekTimeout variable in the Helix Server
configuration file. See the Helix Administrator online help topic **Configuration File > Content Management Configuration > Server-Side Playlist Configuration**.

**Playlist Reference**

The `plref=ID` value instructs Helix Server to skip to a different clip or chapter in the playlist. The parameter value matches the case-sensitive ID assigned in the clip tag or the sequence tag. You can pair this parameter with a `time=ms` parameter to fast-forward to a specific position in the new chapter’s timeline.

**For More Information:** For more on chapter IDs, see “Defining a Chapter or Clip ID” on page 171.

**Time**

The `time=ms` parameter indicates the number of milliseconds into the timeline at which to start playing. When the HTTP or HTTPS directive also contains a `plref=ID` value, the time parameter indicates the amount of time into the specified chapter to start playback. If the directive does not contain a `plref=ID` value, the time parameter sets an absolute time from the start of the playlist.

**Tip:** The playlist `clipBegin` attribute can set a clip’s start position at somewhere other than the clip’s normal beginning. In this case, the `clipBegin` value in the playlist and the `time` offset in the HTTP command are cumulative. See “Defining the Clip Beginning” on page 167.

**Rate Control Flag**

For the rate control flag, the default value is `mdp=1`. This allows server-side rate control to modify the streaming rate and accommodate fluctuating network conditions. The value `mdp=0` prevents server-side rate shifting. This means that Helix Server delivers each clip or broadcast at a single streaming rate unless the media player uses a client-side rate control method such as SureStream.

**Note:** Once the session starts, the rate control state (enabled or disabled) stays in effect for the entire RTSP session.

**For More Information:** Server-side rate control must be enabled and configured for each type of client. For details about rate control, refer to Chapter 8.
Playlist Format

The Helix Server playlist format utilizes SMIL 2.0, a case-sensitive, XML-based language for creating multimedia presentations. Typically, a media player interprets the SMIL mark-up. With playlists, however, only Helix Server reads the playlist and interprets the mark-up. This makes playlists compatible with media players that do not support the SMIL 2.0 standard.

A playlist file uses an .hpl file extension. It must contain a <body> section, and it typically includes the optional <head> section. The opening <smil> tag must declare the SMIL 2.0 namespace, as shown in the following example:

```xml
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    ...header section for defining metadata...
  </head>
  <body>
    ...body section for defining clips...
  </body>
</smil>
```

**Tip:** If the .hpl file extension causes an error for a media player, you can use the extension .hpl.rm, .hpl.mp4, or .hpl.3gp.

**Note:** Playlists support the use of HTML-style comments within the file. For example: <!-- This is a comment -->

**For More Information:** Pronounced *smile*, SMIL stands for Synchronized Multimedia Integration Language. You can find the official SMIL 2.0 specification at the W3C Web site, [http://www.w3.org/TR/smil20/](http://www.w3.org/TR/smil20/).

SMIL Timing Values

Playlist attributes that specify relative timing values use the following syntax:

```
hh:mm:ss.xy
```

Where:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>hh</td>
<td>hours</td>
</tr>
<tr>
<td>mm</td>
<td>minutes</td>
</tr>
<tr>
<td>ss</td>
<td>seconds</td>
</tr>
</tbody>
</table>
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. Note that all of the following values are equivalent to 90 minutes:

clipBegin="1:30:00.0"
clipBegin="90:00"
clipBegin="5400"

Setting Metadata Values

The playlist <head> section defines overall metadata for the presentation:

```xml
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="Expires" content="time"/>
    <meta name="title" content="title"/>
    <meta name="author" content="author"/>
    <meta name="copyright" content="copyright"/>
    <meta name="chapter-skip" content="0|1"/>
  </head>
  <body>...
  </body>
</smil>
```

Presentation Title, Author, and Copyright

Most RTSP media players display metadata values in the user interface. The values are ignored by iOS devices. The metadata attributes of title, author, and copyright allow you to assign title, author, and copyright values to the entire playlist. For example:

```xml
<meta name="title" content="Year’s Best Music"/>
<meta name="author" content="Exciting Media, Inc."/>
<meta name="copyright" content="(c)2012"/>
```

Playlist Expiration

The name="Expires" element functions like an HTTP header that provides an absolute time after which the playlist is no longer valid. To set an expiration time, add the following metadata tag to the playlist:
CHAPTER 9: Playlist Management

The time format is `dd-Mon-yyyy hh:mm:ss`. All time fields are required:

<table>
<thead>
<tr>
<th>dd</th>
<th>day numeric value, as in 04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>first three letters of the month, as in Sep</td>
</tr>
<tr>
<td>yyyy</td>
<td>year, as in 2012</td>
</tr>
<tr>
<td>hh</td>
<td>hour in 24-hour format, as in 14 for 2 p.m.</td>
</tr>
<tr>
<td>mm</td>
<td>minute (00 to 59)</td>
</tr>
<tr>
<td>ss</td>
<td>second (00 to 59)</td>
</tr>
</tbody>
</table>

For example, the following element causes the playlist to expire on July 24, 2012 at 4:56.13 p.m:

```html
<meta name="Expires" content="24-Jul-2012 16:56:13"/>
```

The `Expires` attribute is useful for time-sensitive material such as a news show that is updated hourly. When a media player requests a playlist, Helix Server verifies the expiration timestamp against its system clock. It returns a 404 Not Found error if a media player requests an expired playlist.

**Note:** If a playlist expires while a media player is playing the content, Helix Server continues to stream the content until the session stops.

**Chapter Skipping**

Chapter skipping works with externally controlled playlists streamed to RTSP media players. If `chapter-skip` is set to 1, a user-issued skip command causes Helix Server to jump to the next defined chapter in the playlist:

```html
<meta name="chapter-skip" content="1"/>
```

**For More Information:** Chapters are groups of clips, as explained in the section “Setting Up Chapters” on page 170.

**Chapter Skipping Enabled Example**

As an example of chapter skipping, suppose that the playlist defines a sequence of three clips as Chapter 1. Three additional clips make up Chapter 2. To skip between chapters, the session viewer uses a browser-based interface that contacts Helix Server with a `seek=next` or `seek=prev` directive (see “Seek” on page 161).

If Helix Server receives a `seek=next` directive while it is streaming any of the three clips in Chapter 1, it skips to the beginning of Chapter 2 (the fourth
clip). Conversely, if Helix Server receives a seek=prev directive while the viewer is watching any of the three clips in Chapter 2, Helix Server jumps back to the beginning of Chapter 2 or skips back to the start of Chapter 1.

For More Information: See “Seeking to the Previous Clip” on page 161. The section “Chapter Skipping Example” on page 157 illustrates the process of skipping chapters.

Chapter Skipping Disabled Example
The default value for chapter-skip is 0, which disables chapter skipping. In this case, if Helix Server receives a seek=next while streaming the second clip in Chapter 1, the skip takes the viewer to the start of the third clip in Chapter 1. In other words, skip requests step the viewer through the clips in the order in which they are listed in the playlist, not in the order of defined chapters.

Adding Clips to the Playlist
Within the playlist <body> section, individual <video> elements identify each clip or broadcast. If the <video> element contains only simple attributes, it can be a unary tag ending with a closing slash. For example:

```html
<video src="/media/clip_1.mp4 dur="20000"/>
```

You can also write the <video> element as a binary tag. This is required if the element uses complex attributes expressed as param values. In this case, the <video> element does not include a closing slash. Instead, a </video> tag follows the param values. For example:

```html
<video src="/media/clip_1.mp4 dur="20000">
  <param name="skip-point" value="sequence_6"/>
</video>
```

Tip: Because the <video> element generically identifies a media source rather than a media type, you can use it for audio clips as well. Helix Server identifies the actual media type using the file extension or MIME type of the media stream.
Clip Attribute Summary

The following table summarizes the attributes that you can add to a `<video/>` element. Only the `src` attribute is required.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clipBegin</code></td>
<td>time</td>
<td>Sets the point where the clip starts playback. Can be used with all types of sessions and media clients.</td>
<td>page 167</td>
</tr>
<tr>
<td><code>dur</code></td>
<td>time</td>
<td>Determines how long the clip plays. Can be used with all types of media clients.</td>
<td>page 168</td>
</tr>
<tr>
<td><code>id</code></td>
<td>string</td>
<td>Sets an ID used with skipping. This is useful only with externally controlled playlists.</td>
<td>page 171</td>
</tr>
<tr>
<td><code>skippable</code></td>
<td>0</td>
<td>1</td>
<td>Indicates if the viewer can seek through the clip or skip over the clip.</td>
</tr>
<tr>
<td><code>skip-point</code></td>
<td>ID</td>
<td>Sets a target for a <code>seek=next</code> action. This is supported only with externally controlled playlists.</td>
<td>page 169</td>
</tr>
<tr>
<td><code>src</code></td>
<td>path</td>
<td>Provides the relative path to the clip or playlist.</td>
<td>page 167</td>
</tr>
<tr>
<td><code>src-dur</code></td>
<td>time</td>
<td>Gives a clip duration for playlist timing purposes.</td>
<td>page 168</td>
</tr>
</tbody>
</table>

Specifying the Media Source

The `src` attribute provides the relative URL to the location of the clip or live broadcast on Helix Server. The value starts with the content or broadcast mount point under which the media resides. For example, the following clip is stored under the Helix Server default mount point for content (`/`):

```xml
<video src="/clip_1.mp4" .../>
```

The next clip is stored under a user-defined mount point named `/media/`:

```xml
<video src="/media/clip_2.mp4" .../>
```

Defining the Clip Beginning

The optional `clipBegin` attribute defines the amount of time into the clip’s internal timeline at which playback begins. If you omit it, the clip starts to play at its encoded beginning. The following example starts the clip at its 20-second mark:

```xml
<video...clipBegin="20"/>
```
A time value that falls outside of the clip’s timeline is ignored. For example, setting \texttt{clipBegin=\textasciitilde60} on a clip that lasts 30 seconds has no effect. The \texttt{clipBegin} attribute is ignored for live broadcasts.

\textbf{Note:} If the \texttt{clipBegin} value used with a video clip does not fall on a keyframe, Helix Server starts playback at the preceding keyframe. If the first preceding keyframe occurs five seconds before the \texttt{clipBegin} value, for example, the actual clip timeline will be five seconds longer than calculated by Helix Server and will be reported to the media player when the playlist initializes.

\textbf{Defining the Clip Duration}

The optional \texttt{dur} attribute controls the amount of time that a clip plays once it starts to stream. The following example ends the clip after 85 seconds, regardless of the length of the clip’s internal timeline:

\texttt{<video...dur=\textasciitilde85/>}

If you omit the \texttt{dur} value, the clip plays to its encoded end point. If the specified duration exceeds the actual clip length, Helix Server ignores the \texttt{dur} value. The \texttt{dur} attribute has no effect on live broadcasts.

Helix Server takes the \texttt{clipBegin} value into account when calculating the clip’s end point. For example, if the \texttt{<video/>} element specifies both \texttt{clipBegin=\textasciitilde20} and \texttt{dur=\textasciitilde40}, the clip stops playing at the 60-second mark of its internal timeline. When it stops, it will have played for 40 seconds.

\textbf{Specifying the Clip Duration for Timing Purposes}

The optional \texttt{src-dur} attribute indicates the total length of the clip. Helix Server uses this value only to calculate playlist lengths, not to shorten a clip’s playback duration. If a \texttt{dur} attribute is also present in the video tag, Helix Server uses the \texttt{dur} value and ignores the \texttt{src-dur} value. You specify \texttt{src-dur} as a \texttt{param} element embedded in a binary \texttt{<video>} element:

\texttt{<video...>}
    \texttt{<param name=\textasciitilde\texttt{src-dur} value=\textasciitilde122.3/>}
\texttt{</video>}

\textbf{Warning!} The \texttt{src-dur} value should always reflect the actual clip length to an accuracy of at least one-tenth of a second. Because Helix Server uses \texttt{src-dur} values to calculate the overall playlist duration, inaccurate \texttt{src-dur} values may lower the quality of
playback, adversely affect the timing of seek actions, or cause a media player to end the presentation before all media sources have fully played.

Clip Durations and Presentation Seeking
Helix Server can calculate an overall presentation timeline only if each clip contains either a dur or a src-dur value. If an on-demand clip lacks either value, Helix Server streams the entire playlist as if it were a live broadcast. This has the following ramifications:

• In a noncontrolled playlist session, the viewer cannot fast-forward or rewind through the playlist.

  **Note:** An internally controlled session automatically fails over to a noncontrolled session when clip durations are not defined.

• In an externally controlled playlist session, an HTTP or HTTPS directive using the time parameter is ignored (see “Time” on page 162). However, directives to seek to other chapters within the playlist are honored.

Setting the Skip Point
The skip-point element lists as its value an existing chapter ID in the primary playlist or any secondary playlists. If the user issues a skip=next directive, Helix Server skips to the skip-point target, rather than the next chapter. If the skip-point does not exist in the playlist, Helix Server skips to the next clip or chapter in the defined sequence.

You specify skip-point as a param element embedded in a binary <video> element:

```html
<video...>
  <param name="skip-point" value="sequence_6"/>
</video>
```

**Note:** The skip-point value is ignored if chapter-skip is enabled in the playlist metadata (see “Chapter Skipping” on page 165), or if the skippable value for the current clip is false.

**For More Information:** See “Defining a Chapter or Clip ID” on page 171.
Making a Clip Non-Skippable

The skippable parameter with the default value of 1 allows the user to skip the clip or seek through it. You can disable skipping or seeking for a clip by setting the skippable attribute to 0 within a param element embedded in a binary <video> element:

```xml
<video>
  <param name="skippable" value="0"/>
</video>
```

The configuration variable AllowInbandforNonskippablePL determines if seeking is available in an internally controlled playlist that contains a non-skippable clip. By default, making a clip non-skippable automatically turns an internally controlled playlist into a non-controlled playlist. That is, it disables seeking, making the entire playlist appear to be a live broadcast.

**For More Information:** See the Helix Administrator online help topic Configuration File > Content Management Configuration > Server-Side Playlist Configuration.

Setting Up Chapters

An externally controlled session plays media resources in the sequence defined by the playlist. Within the overall playlist sequence, however, you use the SMIL <seq> element to combine clips into chapters. Consider the following example:

```xml
...earlier chapters...
  <seq id="chapter_3">
    <video id="clip_8" src="clip_8.mp4"/>
    <video id="clip_9" src="clip_9.mp4"/>
    <video id="clip_10" src="clip_10.mp4"/>
  </seq>
...later chapters...
```

Because the three clips are enclosed within <seq> and </seq> tags, the media player treats the clips as a single chapter. Skipping to the next chapter (if allowed) takes the viewer to the clip or sequence that starts after chapter_3.

**Tip:** If a clip is not contained with a <seq> group, the clip functions as a single chapter. For example, if a playlist has no defined chapters or <seq> groups, each skip=next directive takes the viewer to the next clip defined in the playlist.
Defining a Chapter or Clip ID

The optional id attribute identifies a chapter or any target for a skip directive. You can add an id value to a <seq> tag or to individual <video/> or <ref/> tags (see “Specifying the Media Source” on page 167). The ID provides a unique, user-defined name:

```xml
<video id="clip5" .../>
```

The ID is used with plref=ID directives (see “Playlist Reference” on page 162), as well as to define skip-point locations (see “Setting the Skip Point” on page 169) in externally controlled playlist sessions. The following rules apply to the use of IDs:

- Each ID must be unique within a single playlist.
- If you use two or more words for an ID, combine the words, or separate the words with an underscore or hyphen, as in clip1, clip-1, or clip_1.
- The first character for an ID can be a letter, a colon, or an underscore. It cannot be a number or a special character such as an ampersand (&). You can use numbers and special characters after the first character, however. For example, you can use id="video3" as an ID but not id="3video".
- There is no minimum or maximum length for IDs.
- IDs are case-sensitive.

Streaming a Secondary Playlist

A playlist can refer to another playlist. When Helix Server reaches the reference to the secondary playlist, it begins to stream that playlist’s media resources. When that playlist finishes, Helix Server returns to the original playlist.

To set up a secondary playlist, add a <ref> element to the primary playlist. This element functions like a source clip <video> element. For the <ref> element’s src attribute, indicate the secondary playlist located on Helix Server:

```xml
<ref src="/playlists/playlist2.hpl"/>
```

Notes on Streaming Secondary Playlists

Note the following about secondary playlists:

- Secondary playlists function only with RTSP media clients. Do not include them in playlists used with iOS devices.
• Clips in all secondary playlists must be compatible with the first clip streamed from the primary playlist.

For More Information: See “Stream Requirements for RTSP Players” on page 146.

• Most metadata values listed in the <head> section of a secondary playlist are ignored. Metadata values for title, author, and copyright in the first playlist stay in effect through the presentation.

• The chapter-skip value of the original playlist is observed throughout the secondary playlist session.

• Helix Server honors the Expires date for a secondary playlist. If the expiration time has passed, Helix Server ignores the playlist, issues an error message, and continues to the next clip in the primary playlist.

For More Information: See “Playlist Expiration” on page 164.

• A skip-point in the primary playlist can target a chapter in a secondary playlist. As well, a plref directive can skip to a chapter in the secondary playlist from the primary playlist. For these functions to work, all chapters within the primary and secondary playlists must have unique IDs.

For More Information: For background on these functions, refer to “Setting Up Chapters” on page 170, “Setting the Skip Point” on page 169, and “Playlist Reference” on page 162.

Playlist Examples

The following sections provide examples of playlist features.

Simple Playlist

The following is a simple playlist of three clips that play in sequence:

<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="title" content="Coming Attractions"/>
    <meta name="author" content="Exciting Media, Inc."/>
    <meta name="copyright" content="(c)2012"/>
  </head>
  <body>
    <video id="clip_1" src="/media/clip1.mp4">
      <param name="src-dur" value="185.3"/>
    </video>
  </body>
</smil>
Note the following about this playlist:

• The body consists of a sequence of clips without a <seq> grouping. In this case, each clip functions as a separate chapter.

• Each clip includes an id value, which allows the viewer to skip to different clips using an HTTP or HTTPS request in an externally controlled playlist session.

• All clips include src-dur values that indicate the clip’s playing time. This allows Helix Server to calculate the session’s total playing time, as well as to seek forward or backward across the three clips in either an externally controlled or internally controlled playlist session.

Advertisements Preceding Video Clips

The following sample playlist contains three video clips. Each clip is preceded by a short advertisement. In the header, the chapter-skip attribute is enabled. In this case, a user-initiated skip command takes the viewer to the next defined sequence. This effectively groups each ad clip with its subsequent content clip:

```xml
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="title" content="Coming Attractions"/>
    <meta name="chapter-skip" content="1"/>
  </head>
  <body>
    <seq id="chapter_1">
      <video id="ad_1" src="/ads/ad1.mp4">
        <param name="src-dur" value="31.3"/>
      </video>
      <video id="preview_1" src="/media/preview1.mp4">
        <param name="src-dur" value="185.4"/>
      </video>
    </seq>
  </body>
</smil>
```
Uploading Playlists

After the Web portal creates a playlist, it uploads it to Helix Server. If the portal has write access to the Helix Server file system, it can copy playlists to the proper directory. Otherwise, the portal uses the HTTP methods described in the following sections.

**Note:** For the Web portal to upload a playlist, a mount point must have a Yes value set for its **Allow File Creation** option. A delete action requires a Yes value for the **Allow File Deletion** option. For an update action, both permissions must be set to Yes. For details, refer to the Helix Server online help topic Helix Administrator > Server Setup > Mount Points > Creating an On-Demand Mount Point.

Adding a Playlist

To upload a playlist, the Web portal issues an HTTP or HTTPS XML POST command directed at the Helix Server file system control port. The request URL includes an add command along with a path query string parameter that indicates the intended mount point and playlist name. For example:

http://helixserver.example.com:8010/add?path=playlists/songs.hpl
The preceding example writes the posted playlist data to the file songs.hpl under the Helix Server mount point /playlists/. Note that a forward slash preceding the mount point name is not used in the actual command.

**Tip:** The path can also include subdirectories under the mount point, such as playlists/music/songs.hpl. If the specified subdirectory does not exist already, Helix Server creates it, assigning it the same permissions as the parent directory.

**Warning!** Do not enclose the mount point and playlist string in single or double quotation marks. If the string includes spaces, URL-encode each space character as %20.

**For More Information:** If authentication is enabled, Helix Server prompts for a user name and password. For more about authentication, refer to “Secure the Playlist Management System” on page 179.

### Updating a Playlist

To update a playlist, the Web portal adds a new version of the file to the same playlist mount point, overwriting the existing playlist. If viewers are playing the older playlist when the newer playlist arrives, they continue to receive the older content until their session finishes. Requesting the playlist again provides the updated playlist.

**Warning!** The Web portal must ensure unique naming of playlists to avoid inadvertently overwriting one playlist with another.

### Deleting a Playlist

To delete a playlist, use an HTTP or HTTPS GET command directed at the file system control port. The URL includes a delete command and a path query string parameter:

http://helixserver.example.com:8010/delete?path=playlist/songs.hpl
Logging and Error Codes

The following sections explain how Helix Server logs clips streamed during a playlist session, as well as how it reports HTTP errors to the Web portal or user browser.

Logging Styles and Variables

Helix Server adds a log record for each clip or broadcast when the content streams. To record information about playlist sessions, set the logging style to 8. This logging style includes three fields that contain playlist information for each logged clip. These fields provide the guid value, list the order that clips played, and indicate the last clip in the playlist.

Custom Logging Variables

You can also define custom logging reports. The following are the Helix Server registry variables related to playlist logging:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%Client.<em>.Session.</em>.SessionIDName%</td>
<td>Session ID assigned by the guid parameter. See “Session ID” on page 161.</td>
</tr>
<tr>
<td>%Client.<em>.Session.</em>.Clip.*.ClipCount%</td>
<td>Counter that starts at 1 and increments by 1 for each successive clip.</td>
</tr>
<tr>
<td>%Client.<em>.Session.</em>.Clip.*.FinalClip%</td>
<td>Counter set to 0 for all clips in the playlist, except the final clip, which has a value of 1.</td>
</tr>
</tbody>
</table>

For More Information: Refer to the Helix Administrator online help topic Registry Properties > Client Session and Clip Properties for details about logging variables.

Error Conditions

Helix Server does not verify the contents of a playlist before streaming the resources. If a clip is incompatible with the preceding clip in a playlist streamed to an RTSP player, Helix Server omits that clip and streams the subsequent clip. Any of the following changes in format or encoding may cause a clip to be omitted:

- Switching from one file format to another, such as from MPEG-4 (.mp4) to RealMedia (.rm).
- Changes in the codec used to encode the streams, such as from H.264 to H.263.
• Switching the media bit rate when streaming to RTSP players. For multi-rate clips, all bit rate encodings must be identical to support playback with RTSP players.

• Changes in maximum video frame rate or audio sampling rate when streaming to RTSP players.

• Switching from the RDT packet format to RTP (or vice versa) when streaming to RealNetworks media players.

• Streaming content located on a different server.

Timeline Error from Omitted Clips

In an internally controlled playlist, an omitted clip can result in a timeline error. With this type of playlist, Helix Server adds the clip duration values to calculate an overall playback time that it sends to the media player. If Helix Server must omit one or more clips, the actual playback time will be shorter than the total playback time reported to the media player.

**Note:** Some media players do not tear down the RTSP session once the last clip has played. Instead, these players wait for the advertised playback time to elapse, or for the viewer to stop the stream. This results in blank playback equal to the playing time of the omitted clips.

HTTP Status Codes

The following are the HTTP status codes that Helix Server may return for a playlist request.

200 OK

Action successful.

204 No Content

Playlist switch request, such as a chapter skip, was successful.

273 No Session

Requested change to the playlist session failed because the session identified by the guid value is no longer active. See “Session ID” on page 161.
274 Content Not Compatible

Clip is not compatible with previous clips in the playlist.

403 Forbidden

1. Requested playlist session uses a guid value already in use. See “Session ID” on page 161.

2. Request to add or delete the playlist failed because of one of the following reasons:
   - Helix Server file permissions do not allow the requested action.
   - Control port value in the URL is incorrect.
   - The Allow File Creation or Allow File Deletion option is set to False for the playlist mount point.

404 Not Found

1. Requested playlist does not exist under the specified mount point. See “Adding a Playlist” on page 174.

2. Next clip in the playlist was not found.

3. Playlist expiration time has been exceeded. See “Playlist Expiration” on page 164.

4. Request to modify the playlist session failed because the a guid value could not be found. See “Session ID” on page 161.

Configuring Helix Server for Playlist Control

The following sections explain the components of Helix Server that must be configured to enable server-side playlist management.

Set Playlist HTTP Ports

The server-side playlist system uses two Helix Server ports for HTTP or HTTPS communications with the Web portal. These ports are assigned during system installation. Navigate to the Ports page in Helix Administrator (Server Setup>Ports) to ensure that the ports are set to the values you want.

Tip: The setting for the Enable Control Port Security field on the Helix Administrator ports page determines if only HTTP
(security disabled) or HTTPS (security enabled) requests are accepted on the control port.

Note: If a firewall separates the Web portal and Helix Server, ensure that the firewall allows HTTP or HTTPS traffic on these ports.

For More Information: See “Feature Control Ports” on page 33 for more about these port values. For instructions about changing port values, refer to the Helix Administrator online help topic Helix Administrator > Server Setup > Ports > Port Assignments. The online help also explains how to set up access rules that limit the use of certain ports to specific IP addresses.

Define Playlist Mount Points

RealNetworks recommends that you store playlists under a dedicated Helix Server mount point that does not store content clips. In Helix Administrator (Server Setup > Mount Points), create a mount point such as /playlists/. Create one or more playlist mount points depending on your needs.

Note: For each playlist mount point, set Yes values for the Allow File Creation and Allow File Deletion options. This allows the Web portal to write and delete playlists as described in the section “Uploading Playlists” on page 174.

Tip: The base path that corresponds to a mount point may have subdirectories. You can create subdirectories under the mount point’s base path directly. Or, the Web portal can create a subdirectory when it uploads a playlist.

Secure the Playlist Management System

User name and password authentication is used with Web portals that add or delete playlists to Helix Server mount points. The section “Uploading Playlists” on page 174 explains the process for adding or deleting a playlist.

Warning! Failure to secure playlist mount points may allow outside processes to add content to, and delete content from, any unsecured Helix Server mount point.
Playlist Authentication

The following procedure summarizes the process for setting up authentication for playlist addition and deletion commands.

➤ To implement authentication for playlist management:

1. For the authentication realm (Security > Authentication), select the predefined realm SecurePlaylistManager. Then indicate if the Web portal credentials will be submitted using the Basic or Digest authentication protocol.

   **Warning!** Using a realm other than SecurePlaylistManager causes playlist authentication to fail. You can select a different realm only if you manually update the FileSystemControl list in the Helix Server configuration file. For details, refer to the Helix Administrator online help topic Configuration File > Content Management Configuration > Server-Side Playlist Configuration.

   **For More Information:** For background on realms as well as the Basic and Digest protocols, refer to the Helix Administrator online help topic Helix Administrator > Security > Authentication.

2. Under the SecurePlaylistManager realm, click Add a User to Realm. In the pop-up dialog, enter the user name and password that the Web portal must submit when adding or deleting a playlist.

   **Tip:** The user name and password used to access Helix Administrator is automatically added to this realm when the system is installed.

User Authentication

Securing playlist mount points against the SecurePlaylistManager realm causes Helix Server to authenticate only playlist additions and deletions. When a user requests a playlist stored under the playlist mount point, Helix Server does not require a user name and password, however.

You can implement user name and password authentication for desktop media players requesting playlist content. To do so, place the playlist clips under a secure content mount point (such as the default security mount point, /secure/). Then, set up user names and passwords for individual users within a content authentication realm, such as SecureContent.
**Warning!** When Helix Server authenticates a playlist request, it validates only the request for the first clip in the playlist. It does not require authentication for subsequent clips in the playlist. If the first clip does **not** reside under a secure mount point, **no** authentication occurs for any content in the playlist.

**Enable iOS Segmentation**

The segmentation features described in Chapter 7 must be enabled to support playlist management for iOS devices. All content included in playlists must fall under a mount point enabled for segmentation, such as `/iPhone/`. You must also enable segmentation for applicable broadcasts, such as those that are streamed under the default `/broadcast/` mount point.

**Enable Playlist Control Requests**

Once you have set up and secured the necessary mount points and HTTP ports, you must enable Helix Server to listen for playlist control requests. In Helix Administrator, navigate to **Content Management > Serverside Playlist**. In the drop-down box for **Enable Serverside Playlist**, select **Yes** and click **Apply**.

**Warning!** Setting this option to **No** prevents Helix Server from honoring requests to skip to different parts of the playlist. However, it does **not** prevent users from requesting and receiving playlist content.
This section explains how to broadcast live events. Once you understand the basics of live broadcasting, you can learn how to distribute your broadcasts as widely as possible.
Helix Server can broadcast live streams in RealMedia, Windows Media, Flash, MPEG-4, and H.264/AAC formats. This chapter explains live unicasting, which forms the basis for multicasting and splitting described in Chapter 11 and Chapter 12.

**Tip:** Chapter 13 explains how to deliver a prerecorded clip as if it were a live event. This is a good way to test broadcasting before streaming a live event.

### Understanding Unicasts

In a broadcast, an encoder such as Helix Producer delivers a live stream to Helix Server, which then replicates the stream to each media player. Viewers typically receive the broadcast by clicking a Web page link that includes one or more mount points that determine how Helix Server delivers the live stream to the media player.

### H.264/AAC Broadcasts

By encoding live streams using the H.264 video codec and the AAC or AAC+ audio codec, you can reach the widest audience of media players.

#### Supported H.264/AAC Encoders

The following encoders can create live H.264/AAC streams:

- **Helix Producer**
  
  Helix Producer can encode H.264/AAC as either an MPEG-4 or 3GPP Release 6 output type. (The output type does not matter for the live broadcast.) For the broadcast method, you can use any of the following:
  
  - **Helix Push** (see page 187)
  
  - **Helix Advanced Push** (see page 189)
- Helix Pull (see page 191)

- RTP-based live media encoder

Most RTP-based media encoders, such as MPEG-4 and 3GPP encoders, can transmit live H.264/AAC streams to Helix Server using the RTP Push broadcast method (see page 192).

- Flash Media encoder

Most encoders that produce live Flash streams can create content using H.264/AAC codecs. The Flash broadcast method (see page 199) supports connections from Flash encoders to Helix Server.

**Media Player Support for H.264/AAC**

Flash Player, iOS devices, and RTSP-based players can receive H.264/AAC live broadcasts regardless of the encoder used to generate the stream. In the request URL, mount points instruct Helix Server to deliver the content using the media player’s expected method. For example, Flash Player receives an H.264/AAC stream generated by Helix Producer as if the stream originated from a Flash encoder.

*For More Information:* See “iOS Source Media Requirements” on page 86 for specific H.264/AAC requirements for iOS devices. The Helix Administrator online help explains how to write URLs to broadcasts based on the encoder type and the media player.

**Broadcast Archives**

Helix Server can archive broadcasts using RealMedia, MP3, or H.264/AAC codecs. An H.264/AAC stream is saved in the MPEG-4 file format (.mp4). Helix Server does not archive other broadcast formats, such as Windows Media. You can stream archived files immediately after they are recorded. In Helix Administrator, navigate to the Broadcasting > Live Archiving page to set up archiving.

*For More Information:* Refer to the online help topic Helix Administrator > Broadcasting > Live Archiving for details about setting up rules for selectively archiving broadcasts.
### Broadcast Redundancy

When broadcasting any media format, you can specify one or more backup encoders. When the encoders connect to Helix Server to deliver their live event streams, they form a queue based on their connection order. To modify encoder redundancy settings, navigate to **Broadcasting > Broadcast Redundancy** in Helix Administrator.

**For More Information:** Refer to the Helix Administrator online help topic **Helix Administrator > Broadcasting > Broadcast Redundancy** for setup details.

### Standby Messages

If a live broadcast has not started or is interrupted, you can send a message that indicates general information about the broadcast, such as when it is scheduled to play and what viewers can do if the stream is interrupted. You do this by making a file that contains the message you want to display, and placing it in a subdirectory with the same name as the live mount point.

**For More Information:** For instructions about creating a standby message, refer to the Helix Administrator online help topic **Helix Administrator > Broadcasting > Standby Messages**.

### Bandwidth Constraints

In unicasting, each broadcast stream uses bandwidth, so you are also limited by Helix Server’s available, outgoing bandwidth. Unicasting from a single Helix Server is generally suited for light- to medium-volume broadcasts. For events with a large number of viewers, you can use splitting (see Chapter 12), multicasting (see Chapter 11), or a combination of the two to deliver a large number of broadcast streams or to conserve outgoing bandwidth.

### Helix Push Broadcast Method

A Helix Push broadcast (formerly called “account-based”) is the simplest method for connecting Helix Producer to Helix Server. In addition to its data
channel, the encoder maintains an accounting channel that allows it to receive broadcast feedback and reestablish a dropped connection.

<table>
<thead>
<tr>
<th>encoders</th>
<th>RealProducer (RealMedia only)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RealProducer 10 and 11 (RealMedia only)</td>
</tr>
<tr>
<td></td>
<td>Helix Producer 13 and later (RealMedia, MPEG-4, 3GPP)</td>
</tr>
<tr>
<td>server setup</td>
<td>Requires only a user name and password defined within the Helix Server authentication database.</td>
</tr>
<tr>
<td>mount point</td>
<td>/broadcast/</td>
</tr>
<tr>
<td>transport</td>
<td>UDP or TCP for data</td>
</tr>
<tr>
<td></td>
<td>TCP for monitoring connection</td>
</tr>
</tbody>
</table>

**Helix Push Broadcast Connection Steps**

The following figure illustrates the interaction between Helix Producer, Helix Server, and a media player in a Helix Push broadcast.

1. Helix Producer contacts Helix Server on its HTTP port and establishes a monitoring connection. Using this connection, Helix Producer sends a user name and password to authenticate its access attempt.

2. Helix Server validates the Helix Producer user name and password in its authentication database. It then informs Helix Producer how to make the broadcast connection, telling it which server ports to use, for example.

3. Helix Producer establishes the broadcast stream connection to Helix Server. It begins to send the encoded packets whether or not any media players have requested the broadcast yet.

4. A viewer requests the broadcast by clicking a link in a Web page.

5. Helix Server streams the broadcast data to the media player.
For More Information: For details about broadcast URLs, refer to the Helix Administrator online help topic Helix Administrator > Broadcasting > Helix Push.

Helix Push Setup

You set up Helix Push broadcasting in Helix Administrator by navigating to the Broadcasting > Helix Push page. Here you can configure the range of data ports (50001 to 50050 by default) and define a user name and password in the SecureRBSEncoder realm.

On Helix Producer, you set up a Helix Push broadcast method (“account-based” on older versions), specifying the Helix Server IP address along with the same user name and password recorded in the server database.

For More Information: For specific configuration details, refer to the Helix Administrator online help topic Helix Administrator > Broadcasting > Helix Push > Setting Up a Helix Push Broadcast.

Helix Advanced Push and Multicast Push Broadcast Methods

A Helix Advanced Push broadcast (formerly called “password-only”) provides a more robust connection method than Helix Push mode. Helix Multicast Push is identical to Helix Advanced Push except that it broadcasts to a multicast address, allowing multiple Helix Servers to receive the live stream.

| encoders: | RealProducer (RealMedia only) |
| server setup: | Requires that Helix Server be set up as a receiver in a splitting arrangement (see Chapter 12). |
| mount point: | /broadcast/ |
| transport: | Helix Advanced Push: UDP unicast or TCP |

Helix Advanced Push Broadcast Connection Steps

The following figure illustrates the interaction between Helix Producer, Helix Server, and a media player in a Helix Advanced Push broadcast.
Helix Advanced Push Broadcast

1. As soon as it starts encoding the broadcast stream, Helix Producer sends the stream packets to Helix Server. The password required by the Helix Server receiver is encoded into the data stream.

   **Note:** The stream periodically encodes metadata values that allow Helix Server to reconnect to the stream if it has been dropped.

2. A viewer requests the broadcast by clicking a link in a Web page.

3. Helix Server delivers the broadcast data to the media player.

   **For More Information:** Broadcast URLs are the same as for a Helix Push broadcast. See the online help topic *Helix Administrator > Broadcasting > Helix Push.*

**Helix Advanced Push Setup**

You set up Helix Advanced Push or Multicast Push broadcasting in Helix Administrator in the **Broadcast Distribution > Receiver** page. Here you define information such as the password used by the Helix Producer encoder, which acts as the transmitter in a splitting arrangement.

On Helix Producer, you set up a Helix Advanced Push or Helix Multicast Push broadcast method, specifying the Helix Server IP address or multicast address along with the same password recorded in the receiver configuration.

   **Tip:** The Helix Multicast Push method differs from Helix Advanced Push only in that the receiver definition specifies a multicast address for the transmitter and defines the transport as udp/multicast.

   **For More Information:** For specific configuration details, refer to the online help topic *Helix Administrator > Broadcast Distribution > Receiver > Defining a Receiver.* For background
about multicast network requirements, see “Network Configuration for Multicasts” on page 206.

**Helix Pull Broadcast Method**

In pull broadcasting, Helix Producer generates broadcast packets continuously but does not deliver the broadcast stream until Helix Server requests it. For example, an online radio station may continuously encode input but deliver the stream to a specific Helix Server only when a listener requests the stream.

**Helix Pull Broadcast Connection Steps**

The following figure illustrates the interaction between Helix Producer, Helix Server, and a media player in a Helix Pull broadcast.

**Helix Pull Broadcast**

1. Helix Producer begins encoding the stream, but the output is not immediately sent to Helix Server.

2. The first audience member requests the broadcast, typically by clicking a link in a Web page.

3. Helix Server requests the broadcast stream from Helix Producer, sending the encoder information about the server address and ports to use. Once
the connection is established, Helix Server sends keep-alive requests to Helix Producer as long as viewers are receiving the broadcast.

4. Helix Producer sends the broadcast stream to Helix Server, terminating the stream when it receives no more keep-alive messages.

5. Helix Server streams the broadcast to media players.

**Note:** The first viewer may experience a longer-than-normal delay as Helix Server contacts Helix Producer to acquire the broadcast stream. After the server acquires the stream, however, subsequent viewers experience no additional delay.

**For More Information:** URLs for Helix Pull broadcasts build on the basic URL formats used with Helix Push broadcasts. A pull URL includes information about both the receiver and the transmitter, however. For details, see the Helix Administrator online help topic *Helix Administrator > Broadcast Distribution > Splitting Basics > Links to Split Content > Pull Splitting Links*.

**Helix Pull Setup**

You set up Helix Pull broadcasting in Helix Administrator in the **Broadcast Distribution > Receiver** page. Here you set up a receiver and enable it for pull splitting. On Helix Producer, you set up a Helix Pull broadcast destination and specify the same listen port value and password defined on the receiver.

**For More Information:** For specific configuration details, refer to the Helix Administrator online help topic *Helix Administrator > Broadcast Distribution > Receiver > Enabling Pull Splitting Requests*.

**RTP Push Broadcast Method**

Helix Server is preconfigured to receive a live stream in a supported MPEG or 3GPP file format from standards-compliant, RTP-based encoders. This
broadcast method is called RTP Push. Pull broadcasting from a third-party encoder is not supported.

| encoders: | QuickTime encoders  
| RTP-based MPEG-4 or 3GPP encoders  
| Helix Mobile Producer 10 and 11 (3GPP)  
| Helix Producer 13 and later (MPEG-4, 3GPP) |
| server setup: | none required (optional configuration available) |
| mount point: | /rtpencoder/ |
| transport: | set on encoder |

**Tip:** If you are using Helix Producer 13 or later, RealNetworks recommends using one of the Helix broadcast methods rather than RTP Push.

**RTP Push Broadcast Requirements**

To deliver a broadcast stream to Helix Server, an encoder must do the following:

- Support the RTSP control protocol.
- Create output using a standards-compliant RTP packet format.
- Generate a Session Description Protocol (SDP) file (see page 195).

**RTP Push Broadcast Connection Steps**

The following figure illustrates the interaction between an RTP-based encoder, Helix Server, and a media player in an RTP Push broadcast.

**RTP Push Broadcast**

1. The RTP encoder begins to encode a stream and deliver it to the Helix Server address on a specific data port.
Note: Helix Server can receive only one encoder stream on a port. If an encoder attempts to deliver a broadcast stream to a port already in use, Helix Server rejects the stream and logs an error. To prevent this, inform each content provider of the correct port to use and ensure that firewalls allow enough open ports to support simultaneous encoder connections.

2. The RTP encoder creates an SDP file containing information about the encoded stream, such as the stream name and the Helix Server port where the packets are arriving. The file is delivered, typically by FTP, to the rtpencodersdp subdirectory of the Helix Server Content directory. For example, on Windows:
C:\Program Files\Real\Helix Server\Content\rtpencodersdp

On UNIX or Linux, for instance:
/usr/local/Real/HelixServer/Content/rtpencodersdp

Note: This directory corresponds to the /rtpencodersdp/ mount point.

3. The media player requests the live stream.

4. Helix Server delivers the replicated live stream to the media player.

Tip: Deleting the broadcast’s SDP file from the SDP directory does not stop a broadcast in progress. You stop an RTP Push broadcast by terminating the encoder stream. When a broadcast ends, you can delete the SDP file from the SDP directory.

For More Information: URLs for RTP Push broadcasts are similar to those for Helix Push broadcasts. However, RTP Push broadcasts use an /rtpencoder/ mount point in place of the /broadcast/ mount point. See the online help topic Helix Administrator > Broadcasting > QuickTime and RTP Encoding > RTP Broadcast Link Examples.

RTP Push Setup

Helix Server is configured by default to stream an RTP Push broadcast. Optionally, you can navigate to Broadcasting > QT & RTP Encoding in Helix Administrator to change default settings:
• Helix Server can scan the SDP upload directory periodically. When it finds a new SDP file, it acquires and queues the stream. The default action is to check for the SDP file when the first media player requests the broadcast.

• The main SDP directory can receive any number of SDP files as long as each file has a unique file name. If you want to split different broadcasts in different ways (see “Splitting Different Streams to Different Receivers” on page 221), you can create subdirectories for different SDP files. For example:

```
main_installation_directory/Content/rtpencodersdp/news/
```

**Note:** The subdirectory path must precede the SDP file name in the request URL.

**For More Information:** See the Helix Administrator online help topic Helix Administrator > Broadcasting > QuickTime and RTP Encoding for details about setup options.

### SDP Requirements

Helix Server can typically receive RTP-based media streams without problem if the encoder follows the SDP standard. The following sections explain the SDP information necessary for Helix Server to acquire the live stream. Use this information if you need to troubleshoot the encoder connection.

**For More Information:** The SDP specification is available at http://www.ietf.org/rfc/rfc4566.txt.

### SDP File Example

Session Description Protocol uses a text format to describe media streams. Informational fields are designated by letters followed by an equals sign (“=”). Within a field, attribute and value pairs are separated by colons (“;”). The following is an example of an SDP file generated by an encoder:

```
v=0
o=- 1331598807030348 1331598807 IN IP4 10.225.16.98
s=MediaStreaming
i=MediaStreaming
c=IN IP4 10.225.16.44
b=AS:1958
t=0 0
a=range:npt=now-
a=X-wmfversion:1.0
```
a=random_access_denied
a=control:*
a=tool:MediaStreaming
m=video 6066 RTP/AVP 96
b=AS:1864
b=TIAS:1843200
b=RR:9216
b=RS:9216
a=control:trackID=1
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f;packetization-mode=1;sprop-parameter-sets=Z0LAH9oBQBboQAAABkAAAXajxgyo,aM4PyA==
a=cliprect:0,0,720,1280
a=mpeg4-esid:201
m=audio 6068 RTP/AVP 97
b=AS:56
b=TIAS:49152
b=RR:1024
b=RS:1024
a=control:trackID=2
a=rtpmap:97 mpeg4-generic/22050/2
a=fmtp:97 streamType=5; profile-level-id=15;mode=AAC-hbr;config=1390;SizeLength=13;IndexLength=3;IndexDeltaLength=3;
a=mpeg4-esid:101

Note: Not all of the information contained in a typical SDP file is used by Helix Server. The following sections describe the necessary components.

SDP Session Requirements
The SDP file begins with session description information. The following fields are required by Helix Server:

v=0
o=- 1331598807030348 1331598807 IN IP4 10.225.16.98
c=IN IP4 10.225.16.44

Version Number
The v= field gives the SDP version, which is 0 with no minor number.

Origin Address
The o= field indicates the encoder that started the session:
o=- 1331598807030348 1331598807 IN IP4 10.225.16.98
The field may include values such as an encoder user name and a session name. The last three attribute values are required:

- **IN**— Specifies a stream available on the public Internet.
- **IP4|IP6**— Indicates the type of IP address, either IP version 4 or 6. (Only IPv4 addresses are supported by Helix Server for third-party encoder connections.)
- **address**— Provides the encoder's IP address or DNS name.

**Connection Information**
The `c=` line provides data about the encoder connection to Helix Server:

```
c=IN IP4 10.225.16.44
```

- **IN**— Specifies a stream available on the public Internet.
- **IP4|IP6**— Indicates the type of IP address, either IP version 4 or 6. (Only IPv4 addresses are supported by Helix Server for third-party encoder connections.)
- **address**— Provides the Helix Server IP address or DNS name. This may be a unicast address:
  
  10.225.16.44

Or it may be a multicast addressed followed by a time-to-live value in the range of 0-255:

```
224.2.36.42/127
```

*For More Information:* For more about TTL values, see “Packet Time to Live” on page 207.

**SDP Video Field Requirements**

An `m=` video field indicates the beginning of description fields for a video stream. The following lines are required:

```
m=video 6066 RTP/AVP 96
b=AS:1864
a=control:trackID=1
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f;packetization-mode=1;prop-parameter-sets=Z0LAH9oBQBboQAAABkAAAXajxgyo,aM4PyA==
a=cliprect:0,0,720,1280
```
Media Line for Video
The \texttt{m=} line for video indicates the media type (video) and gives the RTP port number along with the RTP dynamic payload number:
\texttt{m=video 6066 RTP/AVP 96}

Bandwidth
The \texttt{b=} line provides the video bandwidth in Kilobits per second:
\texttt{b=AS:1864}

Track Number
An \texttt{a=} control field indicates the track number of the video within the stream:
\texttt{a=control:trackID=1}

RTP Map
An \texttt{a=rtpmap} line matches the payload number of the \texttt{m=} video line (96 in this example). It provides the video MIME type and may include clock rate information:
\texttt{a=rtpmap:96 H264/90000}

FMTP
For \texttt{a=fmtp}, the SDP matches the payload number of the \texttt{m=} video line (96 in this example). It also includes information specific to video decoding:
\texttt{a=fmtp:96 profile-level-id=42c01f;packetization-mode=1;sprop-parameter-sets=Z0LAH9oBQBboQAAABkAAAXajxgyo,aM4PyA==}

Clip Size
The attribute field \texttt{a=cliprect} defines the offset and size (height and width in pixels) of the video:
\texttt{a=cliprect:0,0,720,1280}

SDP Audio Field Requirements
An \texttt{m=} audio field indicates the beginning of description fields for an audio stream. The following lines are required:
\texttt{m=audio 6068 RTP/AVP 97}
\texttt{b=AS:56}
\texttt{a=control:trackID=2}
\texttt{a=rtpmap:97 mpeg4-generic/22050/2}
\texttt{a=fmtp:97 streamType=5; profile-level-id=15;mode=AAC-hbr;config=1390;SizeLength=13;IndexLength=3;IndexDeltaLength=3;}
Media Line for Audio
The m= line for audio indicates the media type (audio) and gives the RTP port number along with the RTP dynamic payload number:

m=audio 6068 RTP/AVP 97

Bandwidth
The b= line provides the audio bandwidth in Kilobits per second:

b=AS:56

Track Number
An a=control field indicates the track number of the audio within the stream:

a=control:trackID=2

RTP Map
An a=rtpmap line matches the payload number of the m=audio line (97 in this example). It provides the audio MIME type, the audio sampling rate in kHz, and the number of audio channels, such as 2 for stereo:

a=rtpmap:97 mpeg4-generic/22050/2

FMTP
For a=fmtp, the SDP matches the payload number of the m=audio line (97 in this example). It then includes information specific to decoding:

a=fmtp:97 streamType=5; profile-level-id=15; mode=AAC-hbr; config=1390; SizeLength=13; IndexLength=3; IndexDeltaLength=3;

Flash Broadcasts
Helix Server can receive single-rate or multi-rate media streams from most Flash media encoders. Helix Server can broadcast the stream to Flash Players. If the content is H.264/AAC, Helix Server can deliver the stream to RTSP media players and iOS devices.

<table>
<thead>
<tr>
<th>encoders</th>
<th>Adobe Flash Media Live Encoder and most hardware encoders such as Media Excel HERO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>server setup</td>
<td>Helix Server defines the RTMP port and an allowable range of IP addresses for encoders</td>
</tr>
<tr>
<td>mount point</td>
<td>/rtmplive/</td>
</tr>
<tr>
<td>transport</td>
<td>RTMP</td>
</tr>
</tbody>
</table>
Flash Broadcast Connection Steps

The following figure illustrates the interaction between a Flash encoder, Helix Server, and Flash Player.

Flash Push Broadcast

1. The Flash encoder begins to stream packets to Helix Server on a designated RTMP port.
2. A viewer requests the broadcast by clicking a Flash link in a Web page.
3. Helix Server delivers the broadcast data to the Flash Player.

For More Information: For details about Flash encoder broadcast URLs, refer to the Helix Administrator online help topics for single-rate and multi-rate broadcasting under Helix Administrator > Broadcasting > Flash Media Encoding.

Flash Broadcast Setup

You set up Flash broadcasting in the Broadcasting > Flash Media Encoding page of Helix Administrator. Here you define the RTMP port on Helix Server where Flash encoders direct stream data. Helix Server can restrict the range of IP addresses for encoder that are allowed to connect. (No authentication is used to verify the encoder connections.)

For More Information: For specific configuration details, refer to the Helix Administrator online help topic Helix Administrator > Broadcasting > Flash Media Encoding > Flash Encoder Setup.

Windows Media Broadcasts

Helix Server is preconfigured to deliver live broadcast streams in the Windows Media format. It includes support for the MMS protocol, a Windows Media
broadcast mount point, and an ASXgen mount point for launching Windows Media Player.

<table>
<thead>
<tr>
<th>encoders:</th>
<th>Windows Media Encoder version 7 or later</th>
</tr>
</thead>
<tbody>
<tr>
<td>server setup:</td>
<td>Helix Server defines the pull sources or the push encoding port.</td>
</tr>
<tr>
<td>mount point:</td>
<td>/wmtencoder/</td>
</tr>
<tr>
<td>transport:</td>
<td>HTTP</td>
</tr>
</tbody>
</table>

**Windows Media Push and Pull Modes**

Broadcasting in the Windows Media format can use a push or pull mode:

- In a Windows Media pull broadcast, Helix Server pulls a stream from Windows Media Encoder version 7 or later over an HTTP connection, delivering the stream to Windows Media Players over the MMS or HTTP protocol.

- In a Windows Media push broadcast, Helix Server receives a stream from Windows Media Encoder version 9 or later on a predefined port. It then delivers the stream to Windows Media Players over the MMS or HTTP protocol.

**Note:** Multiple bit rate (MBR) encoding is supported only with pull broadcasting with Windows Media Encoder version 7. It is not supported with push or pull broadcasting using Windows Media Encoder version 9 or later.

**Windows Media Broadcast Setup**

To set up a Windows Media broadcast, navigate to **Broadcasting > Windows Media Encoding** in Helix Administrator. Refer to the Helix Administrator online help topic **Helix Administrator > Broadcasting > Windows Media Encoding** for detailed instructions about running a Windows Media broadcast.
Multicasting reduces the number of broadcast streams sent to media players. It requires a specially configured network and is more suited for intranets than Internet delivery. You can multicast RealMedia, QuickTime, Windows Media, MPEG-4, and a number of RTP-based formats.

Tip: Chapter 12 explains splitting, which allows you to transmit a live stream from one Helix Server to any number of additional Helix Servers on a multicast-enabled network.

Understanding Multicasts

Unicasting delivers a unique broadcast stream to each media player. In contrast, multicasting sends a single live stream to multiple players. The multicast can be a live event or a prerecorded clip broadcast by SLTA, which Chapter 13 explains. The players connect to the stream rather than to the Helix Server as shown in the following illustration.

Publicizing Multicasts

Optionally, you can publicize back-channel and scalable multicasts to anyone running a program that listens for the Session Announcement Protocol (SAP).
These applications, such as SDR and ICAST Guide, display a list of all multicasts currently playing. Helix Server creates the SAP file automatically. Programs that listen for SAP announcements show the title, author, and copyright information encoded into the files you multicast.

**Note:** The Windows Media multicast format does not support Session Announcement Protocol.

**Tip:** You set up Helix Server to create SAP files in the Helix Administrator page under *Broadcast Distribution > Session Announcement*.

### Multicasting Used with Other Features

The following sections summarize how multicasting works with other Helix Server features.

#### Splitting and Multicasting

Transmitters can multicast a stream to receivers as explained in Chapter 12. This does not require the setup described in this chapter, which concerns only server-to-player multicasts. If the receiver multicasts a stream to media players, however, you must configure the receiver for multicasting as described in this chapter.

#### Live Archiving and Multicasting

As with all live broadcasts, you can configure Helix Server to archive files for live multicasts in the RealMedia or MP3 format. Helix Server does not archive MPEG-4 or Windows Media multicasts.

#### Simulated Live Broadcasts with Multicasting

You can multicast a simulated live stream from on-demand clips using SLTA, which Chapter 13 describes.

#### Helix Proxy and Multicasting

Depending on how the network is configured and the streams are listed in Helix Server, clients whose requests are forwarded by a Helix Proxy may receive different results.

Helix Proxy cannot join a multicast. Instead, it will try to receive the multicast using pull splitting. If pull splitting is enabled on the source Helix Server,
Helix Proxy will use that broadcast instead of connecting to the multicast. The client will receive the broadcast in unicast mode.

**Tip:** If there is a multicast-enabled network between Helix Proxy and the client, Helix Proxy can be configured to resend its pull split stream by multicast instead.

**Firewalls and Multicasting**

Multicasts usually take place within an intranet. If a multicast passes through a firewall, the firewall must be specially configured to allow multicast traffic.

**Access Control, Authentication, and Multicasting**

As with all delivery methods, Helix Server verifies that the client requesting a broadcast is allowed to receive it. If you include an authentication mount point such as `/secure/` in the link, Helix Server verifies the viewer’s identity. You cannot authenticate scalable multicasts, however.

**Reporting and Multicasting**

The basic access log shows which method was used to transmit a stream for a multicast. Scalable multicasts can be identified by the mount point in the `GET` statement. If Helix Server is configured for requesting client statistics, the log file will also contain statistics for each client.

**Multicast Resources**

The Helix Server implementation of back-channel and scalable multicasting is based on industry standards. You may find the following resources useful.

**General Multicasting Information**


**Scalable Multicasting Information**


**Network Configuration for Multicasts**

To use multicasting, Helix Server, media players, routers, switches, and all other networking devices between them must be multicast-enabled. For this reason, multicasting is primary used on intranets. However, it is possible to deliver multicasts over the Internet where intermediary network devices have been multicast-enabled. Before multicasting, verify the following with your network administrator:

- Routers and all equipment in your network are multicast-enabled.
- The machine running Helix Server is correctly configured for multicast support.

**Tip:** RealNetworks and Microsoft media players are configured for multicast by default. Viewers can turn off multicast support in their player preferences, however.

**Multicast Addresses**

A multicast requires the use of a continuous range of multicast addresses on your network. Valid ranges are between 224.0.0.0 and 239.255.255.255. Check with your network administrator about which multicast addresses are available on your network. On the public Internet, certain ranges in the multicast address space (from 224.0.0.0 to 224.0.0.255) cannot be used.

The number of addresses required for a multicast depends on the type of multicast you use as well as the number of streams you deliver. For scalable multicasting, you also need to reserve a number of Helix Server ports for the broadcast. The topics on setting up each type of multicast in the Helix Administrator online help section **Helix Administrator > Broadcast Distribution** explain the number of addresses and ports needed.

**Note:** Helix Server does not support multicasting to IPv6 addresses.
Warning! If you use multiple types of multicasts, such as both back-channel and scalable multicasts, the address ranges you pick cannot overlap.

Packet Time to Live

All multicast broadcasts include a “time to live” feature. As a multicast data packet passes through a multicast-enabled router, its time to live decreases by 1. When the value reaches 0, the router discards the data packet. When you set up a multicast, you specify a time to live of 0 to 255. The larger the value, the greater the distance a packet can travel. The default value of 16 typically keeps multicast packets within an internal network. The following table summarizes possible values.

<table>
<thead>
<tr>
<th>TTL Value</th>
<th>Packet Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>local host</td>
</tr>
<tr>
<td>1</td>
<td>local network (subnet)</td>
</tr>
<tr>
<td>16</td>
<td>intranet</td>
</tr>
<tr>
<td>32</td>
<td>site</td>
</tr>
<tr>
<td>64</td>
<td>region</td>
</tr>
<tr>
<td>128</td>
<td>continent</td>
</tr>
<tr>
<td>255</td>
<td>world</td>
</tr>
</tbody>
</table>

Multicasts with Multiple Network Interface Cards

If your Helix Server machine has multiple network interface cards (NICs) and you want to ensure that Helix Server always uses a particular NIC for multicasts, use your operating system to set a default address. On Windows, set IP bindings as described in the Helix Server online help (Helix Administrator > Server Setup > IP Binding). On UNIX, use the route command to associate the multicast route with the appropriate NIC.

Back-Channel Multicasting

Back-channel multicasting works with all RealNetworks media players that support the RTSP protocol. In this method of multicasting, each media player maintains a control channel to Helix Server. A media player uses its channel to
send commands such as **Stop** and to deliver statistics about the quality of service to the archive log. This channel lets Helix Server receive a user name and password if authentication is used. It also enables the Server Monitor to track how many players are viewing the multicast.

**Back-Channel Multicasting**

![Diagram of Back-Channel Multicasting]

**Note:** Because each player uses a control channel, back-channel multicasting is limited to the number of client connections licensed to your Helix Server.

**Unicast Failovers**

If a RealNetworks media player is not multicast-enabled or cannot connect to the back-channel multicast, it fails over to a unicast automatically. This ensures that all players can receive the broadcast. Because each unicast stream consumes extra bandwidth and Helix Server overhead, you can choose to disable the failover feature and provide just the multicast. In this case, a player not able to participate in a multicast receives an error message when attempting to connect to the broadcast.

**Tip:** You can use back-channel multicasting with the failover feature for all broadcasts to RealNetworks media players. By default, players attempt a back-channel multicast connection first, switching to unicast if the failover feature is enabled, and the multicast is not available. Hence, enabling an automatic multicast for all broadcasts can help conserve bandwidth.
Back-Channel Multicasting Configuration

Back-channel multicasting is enabled by default. Minimally, you need to define your multicast address range. Other features are optional. To set up back-channel multicasting, navigate to Broadcast Distribution > Back-Channel Multicasting in Helix Administrator. Refer to the online help for set-up instructions and multicast address requirements.

After you configure a back-channel multicast, you start a unicast. Media players that can connect to the multicast do so. Players that cannot connect to the multicast use a unicast, as long as the failover feature is not disabled. Links to multicasts are identical to links for unicasts. This enables a single link to serve both multicast and unicast clients.

Scalable Multicasting

Using a scalable multicast, you can broadcast to an unlimited number of media players because the transmission is one way. Unlike back-channel multicasting, scalable multicasting does not use a control channel. Thus, it uses less bandwidth, administrative overhead, and system resources on Helix Server. Scalable multicasting works with RealPlayer and any RTP-based media player that complies with scalable multicasting standards, including the desktop version of Apple’s QuickTime Player.

Session Description Files

Viewers connect to scalable multicasts by clicking a link to a Session Description Protocol (SDP) file, which Helix Server automatically generates. All data is multicast on the network once, and media players do not connect to
Helix Server during the multicast. Because of this, tools such as Server Monitor do not track client connections and activity during the broadcast.

**Authentication and Statistics**

Because media players do not connect to Helix Server directly during the multicast, you cannot enforce user name and password validation in a scalable multicast. Optionally, you can have players connect to Helix Server or a Web server to deliver quality of service statistics when the broadcast ends (or they disconnect).

**Unicast Failovers**

Like back-channel multicasting, scalable multicasting has a failover feature that lets you direct RealNetworks media players to a unicast if they cannot receive the multicast. You can provide a unicast on the same Helix Server that hosts the multicast, or choose a different Helix Server. As well, you can direct players to a Web page that indicates that the multicast is not available, and provides information about broadcast alternatives.

**Scalable Multicasting Configuration**

Using Helix Administrator, you set up a live channel for delivering a scalable multicast to RealNetworks media players. You might set up just one live channel for all scalable multicasts. Alternatively, you can create different live channels to multicast broadcast streams in different ways. To run two multicasts at the same time, for example, you define two separate channels.

To create a live channel, navigate to **Broadcast Distribution > Scalable Multicasting** in Helix Administrator. For setup instructions, refer to the Helix Administrator online help, which also describes how to do the following:

- Determine the number of addresses and ports required for the multicast.
- Gather client statistics.
- Start the scalable multicast.
- Link to the broadcast stream.
Windows Media Multicasts

Helix Server can deliver scalable multicasts in the Windows Media format to Windows Media Players. Each multicast stream is delivered by Windows Media Encoder and made available to players on a single multicast channel. Players click a Web page link to an NSC file to join the multicast. You can designate a unicast URL for players that cannot join the multicast.

**Note:** Unlike with RealMedia back-channel multicasts, Helix Server cannot make all Windows Media unicasts automatically available as multicasts. As well, archiving of Windows Media multicasts is not supported.

**Tip:** You need only one IP address and port for each Windows Media multicast.

Windows Media Multicast Configuration

To set up Windows Media multicasts, you first define one or more multicast channels by navigating to **Broadcast Distribution > Windows Media Multicasting** in Helix Administrator. In this page, you also set up the live source for a channel. Refer to the Helix Administrator online help for setup instructions. The online help also explains how to do the following:

- Run the Windows Media multicast.
- Link to the Windows Media multicast.
- Stop the Windows Media multicast.
Splitting enables one component to deliver a live or simulated live broadcast stream to another component. This allows Helix Producer to transmit a stream to one or more Helix Servers, for example, and a Helix Server to distribute a stream to additional Helix Servers. As a result, splitting fosters higher-quality broadcasts by distributing broadcast streams closer to viewers.

**Tip:** To distribute on-demand clips from one Helix Server to another, use the content caching feature described in the Helix Administrator online help topic *Helix Administrator > Content Management > Content Caching.*

**For More Information:** Before you configure a splitting arrangement, be sure that you understand the concepts and procedures of the basic broadcast methods described in Chapter 10.

### Understanding Splitting

A splitting arrangement involves encoders, transmitters, and receivers that can be set up for push splitting or pull splitting.

### Definitions

**Encoder**

An encoder is software that generates a live or simulated live stream. RealNetworks encoders include Helix Producer and SLTA. Additional, supported encoders include Windows Media Encoder, Sorenson Broadcaster, Flash encoders, and standards-compliant MPEG-4 or 3GPP encoders.
Encoder-to-Server Splitting
Splitting technology is used with SLTA or Helix Producer broadcasting in Helix Advanced Push, Helix Multicast Push, or Helix Pull mode. In these cases, the encoder is the transmitter and Helix Server is set up as a receiver.

Receiver
A receiver is a Helix Server that acquires a stream from a transmitter and broadcasts it to media players. The transmitter may be another Helix Server, Helix Producer, or SLTA.

Relay
A relay is a Helix Server that functions as both a receiver and a transmitter. It acquires a stream from a transmitter and retransmits that stream to another receiver. It may also broadcast the stream to media players.

Pull Splitting
In pull splitting, a receiver initiates the splitting session by requesting the stream from the transmitter. This typically occurs when the first media player requests the stream from the receiver.

Push Splitting
In push splitting, a transmitter initiates the splitting session by delivering the broadcast stream to one or more receivers. The session starts as soon as the encoder begins to generate stream packets.

Server-to-Server Splitting
In server-to-server splitting, a Helix Server transmitter unicasts or multicasts a live or simulated live stream to one or more Helix Server receivers. This allows you to distribute the same stream to multiple Helix Servers across an extended network.

Transmitter
A transmitter is a Helix Server that forwards a stream to a Helix Server receiver. Helix Producer and SLTA can also function as transmitters.

Push Splitting
The following illustration shows the push splitting. Here the encoder delivers the stream to the transmitter, which initiates a connection to the receiver. When a media player requests the broadcast, the receiver is ready to deliver the stream. In this form of splitting, Web page links typically point to the receiver. However, a transmitter can also replicate the broadcast stream to media players.
CHAPTER 12: Splitting

Push Splitting

Note: A transmitter can pull the stream from a pull-enabled encoder but push it to receivers. This lets you start the encoder without making the broadcast available to media players. Use a private URL on the transmitter to pull the stream from the encoder when the broadcast should start. Once this happens, the transmitter automatically pushes the stream to receivers.

Pull Splitting

The following illustration shows pull splitting. Here the transmitter does not deliver the stream to the receiver until the first media player makes a request (step 1). There’s a slight delay as the receiver requests (step 2), receives (step 3), and delivers (step 4) the stream. After that, the stream is live on the receiver, and subsequent player requests do not involve the session setup delay of step 2.

Pull Splitting Bandwidth Efficiency

Although pull splitting results in greater latency than push splitting on the first stream request by a media player, it can save on bandwidth because the stream is not transmitted to the receiver if no media player requests are active on the receiver. As well, if all media players disconnect from the receiver before the broadcast ends, the data stream between transmitter and receiver is dropped. Hence pull splitting does not consume bandwidth between transmitter and receiver if no one is viewing the broadcast on that receiver.
Tip: You can combine push and pull splitting for optimal results. Suppose that you are delivering a broadcast across many different time zones. You could push the stream to receivers that reside in daytime zones. Where it’s late at night and there are fewer potential viewers, you could have receivers pull the stream only on viewer request.

Push Encoding with Pull Splitting

If you set up pull splitting between transmitters and receivers, you can still use Helix Producer or SLTA to push the stream to the transmitter. This cues the broadcast so that the transmitter can respond faster to pull-splitting requests from receivers.

One-to-Many Splitting

A common splitting arrangement uses a single transmitter to broadcast to multiple receivers. If you do this through unicasting, each receiver gets a unique stream, so bandwidth consumption increases with each receiver. Multicasting uses less bandwidth, and is a better solution if all components are on a multicast-enabled network.

The following illustrations shows unicasting through push splitting, though you can also use pull splitting. Here, each receiver connects to the transmitter only when it needs the stream. Server-to-server multicasting is not available with pull splitting, however.

Tip: Helix Producer and SLTA can function as transmitters that deliver streams to multiple Helix Server receivers. In this case, a Helix Server is not needed as a transmitter. To reduce
outgoing bandwidth, use multicast delivery between the Helix Server receivers and the Helix Producer or SLTA transmitter.

**One-to-One Chaining**

Another option is to use one-to-one chaining, in which each receiver transmits to another receiver. Receivers in the middle of the chain thereby function as relays. This option is viable if a group of servers is spread across a wide area, and uses unicasting over the Internet to communicate. A transmitter in San Francisco might push a stream to a Tokyo receiver, which pushes it to a Sydney receiver, and so on.

Although you can use pull splitting with a relay chain, push splitting suits this setup better. With pull splitting, there may be a long latency period if the first broadcast request comes from far down the chain. In this case, the request has to make its way back the chain, causing each receiver in the chain to pull the stream from the preceding transmitter.

**Tip:** Pull-splitting links for a relay chain may become long and increase the likelihood of incorrect broadcast URLs. See the Helix Administrator online help topic Helix Administrator > Broadcast Distribution > Splitting Basics > Links to Split Content > URL Aliases for information about using URL aliases to shorten broadcast URLs.

**Setting Up Split Broadcasts**

You can split most types of broadcast streams originating from supported encoders. The broadcast mount points and set-up methods may vary with each type of broadcast method, however.

For More Information: The Helix Administrator online help section Helix Administrator > Broadcast Distribution > Splitting
Basics > Links to Split Content explains URLs for split broadcast streams.

**Helix Producer Broadcasts**

You can split any stream originating from Helix Producer, including RealMedia and H.264/AAC in an MPEG-4 or 3GPP output. The broadcast mount point for these streams is /broadcast/.

**For More Information:** For details about how to set up a specific type of broadcast on Helix Producer, refer to its online help.

**Helix Push Broadcasts**

The Helix Push broadcast method (see page 187) is the most basic method for delivering a stream from Helix Producer:


- In the Broadcast Distribution > Transmitter page on that Helix Server, you define the server as a transmitter.

- The other Helix Servers that receive the stream are defined as receivers in their Broadcast Distribution > Receiver pages.

**Helix Advanced Push and Helix Multicast Push**

The Helix Advanced Push and Multicast Push broadcast methods (see page 189) are virtually identical except that they use the transport udp/unicast or udp/multicast throughout the splitting chain:

- The Helix Server that receives the stream from Helix Producer is defined as a receiver in its Broadcast Distribution > Receiver page. Helix Producer acts as the transmitter in this setup.

- In the Broadcast Distribution > Transmitter page on the Helix Server that gets the stream from the encoder, you define the server as a transmitter. The server that receives the stream from Helix Producer therefore functions as a relay.

- The other Helix Servers that receive the split stream are defined as receivers in their Broadcast Distribution > Receiver pages.
Helix Pull

In the Helix Pull broadcast method (see page 191), Helix Producer is a pull-enabled transmitter that transmits packets only upon request from pull-enabled Helix Servers receivers:

- The Helix Server that pulls the stream from Helix Producer is defined as a pull-enabled receiver in its Broadcast Distribution > Receiver page.
- Additional Helix Servers can pull the stream from the initial Helix Server receiver by defining that receiver as a pull-source in their Broadcast Distribution > Receiver pages.

SLTA Broadcasts

The SLTA utility (see Chapter 13) allows you to stream on-demand clips from a Helix Server machine as if they were live broadcasts:

- The broadcast mount point for these streams is /broadcast/.
- In its basic mode (see page 232), SLTA functions like Helix Producer in a Helix Push broadcast.
- In its advanced mode (see page 233), SLTA can function like Helix Producer in a Helix Advanced Push, Helix Multicast Push, or Helix Pull broadcast.

RTP-based Broadcasts

You can split a stream from most third-party MPEG-4 or 3GPP encoders (see page 192):

- The mount point for these broadcasts is /rtpencoder/.
- You define the encoding method in the Helix Administrator Broadcasting > QT & RTP Encoding page for the Helix Server that receives the stream from the encoder.
- In the Broadcast Distribution > Transmitter page on that Helix Server, you define the server as a transmitter.
- The other Helix Servers that should receive the stream are defined as receivers in their Broadcast Distribution > Receiver pages.
Flash Broadcasts

Flash broadcasts (see page 199) that encode H.264/AAC streams can be split to different receivers:

- The mount point for these broadcasts is /rtsplive/rtmplive/.
  - The /rtmplive/ mount point signifies a broadcast from a Flash encoder.
  - The /rtsplive/ mount point precedes /rtmplive/ to allow Helix Server to repackage the stream as needed for splitting.
- You define the Flash broadcasting method in the Helix Administrator Broadcasting > Flash Media Encoding page for the Helix Server that receives the stream from the Flash encoder.

  **Note:** The encoder setup page includes an option to allow repackaging to an RTSP-based format. This option must be set to Yes to enable splitting.

- In the Broadcast Distribution > Transmitter page on the Helix Server where the Flash encoder connects, define the server as a transmitter.
- The other Helix Servers that should receive the stream are defined as receivers in their Broadcast Distribution > Receiver pages.

Windows Media Broadcasts

Windows Media push or pull broadcasts (see page 200) can be split to Helix Server receivers:

- The broadcast mount point is /wmtencoder/.
- You set up the Windows Media encoder connection (push or pull) in the Helix Administrator Broadcasting > Windows Media Encoding page for the Helix Server that receives the stream from the encoder.
- For a push broadcast:
  - Define the Helix Server that receives the stream from the encoder as a transmitter in its Broadcast Distribution > Transmitter page.
  - The other Helix Servers that receive the stream are defined as receivers in their Broadcast Distribution > Receiver pages.
- For a pull broadcast:
• The Helix Server that pulls the stream from the encoder is defined as a pull-enabled receiver in its Broadcast Distribution > Receiver page.

• Additional Helix Servers can pull the stream from the initial Helix Server receiver by defining that receiver as a pull-source in their Broadcast Distribution > Receiver pages.

Splitting Different Streams to Different Receivers

You may set up your transmitters and receivers to split every broadcast the same way. In this case, you always use a certain Helix Server as your primary transmitter. The other Helix Servers always function as receivers or relays. This is not necessary, however, and a single network of Helix Servers can support numerous splitting arrangements, enabling you to transmit from any Helix Server. As well, you can split broadcasts from a single transmitter in many different ways.

When you set up splitting, you can create multiple transmitter and receiver definitions on each Helix Server. When you set up a push transmitter, for example, you define how it connects to each receiver. For a single Helix Server receiver on one physical machine, you can create multiple receiver definitions. Flash broadcasts can use one definition, for example, while MPEG-4 broadcasts use another. This lets you multicast one format, for instance, while uncasting another.

Virtual Paths for Stream Direction

You can use multiple receiver definitions to split broadcasts in different ways according to the stream source names that appear in broadcast URLs. Each source name has three parts:

1. **mount point**—Every broadcast uses a mount point. Broadcasts pushed by Helix Producer use the /broadcast/ mount point, for example, while broadcasts from a standards-based MPEG-4 encoder use /rtpencoder/.

2. **virtual path**—An optional path inserted between the mount point and the stream name allows you to selectively split broadcast streams on the same mount point. How you define the path differs for each media format:

   • With RealNetworks encoders, you define the path name when setting up the broadcast method.
• For Windows Media pull broadcasts, you define the path when configuring the encoder pull session on Helix Server. If using push encoding, you define the path through Windows Media Encoder.

• For broadcasts from RTP-based encoders, the path name reflects the subdirectory where the encoder placed the SDP file. (See “RTP Push Setup” on page 194.)

• Flash encoders precede the stream name with an application name that functions as the virtual path.

3. **stream name**—In all broadcasts, the stream name appears last in the URL and looks like an on-demand clip name, often ending with the media format’s standard file extension.

**Stream Routing Example**

The following illustration shows how you can use mount points, paths, and stream names within receiver definitions to split different broadcasts in different ways. Three encoders connect to the same transmitter and deliver three separate streams. The first stream, `live.mp4`, uses no path. The second and third streams, `news/breaking.mp4` and `news/hourly.mp4`, use the same path name but different stream names.

**Stream Direction through Mount Points, Paths, and File Names**

![Stream Routing Diagram](image-url)
Each receiver in the preceding illustration uses a different broadcast source path. These path definitions are created on the transmitter, which directs each stream to its receiver:

- The first receiver accepts all broadcasts that use the broadcast mount point, /broadcast/.
- The second receiver gets all broadcasts that use the /broadcast/news/ mount point and path. It therefore receives streams 2 and 3 but not stream 1, which does not use the news/ path.
- The third receiver gets only the broadcast stream that uses the /broadcast/news/hourly.mp4 mount point, path, and stream name.

### Multicasting Split Streams

Helix Producer, SLTA, and Helix Server support multicast delivery if all hardware components are on a multicast-enabled network. The following sections provide examples of how to use multicasting to create efficient splitting arrangements.

### Unicast Delivery, Multicast Distribution

To deliver a split stream, you can combine unicasting and multicasting. For example, you can unicast streams to receivers across the Internet, an intranet, a wide area network, or a local area network. Then, within an intranet, you can multicast the stream from the receivers to media players. The following illustration shows this type of delivery.
Dual Unicast and Multicast Transport Methods

If your server network is multicast-enabled, you can simultaneously unicast and multicast a broadcast stream to receivers. Duplicate packets arriving by different transport methods increase the network overhead but do not cause problems for receivers. When a receiver reassembles the broadcast stream, it uses the packets that arrive first, regardless of their transport methods. If a unicast packet is late or missing, for example, the receiver may get the right packet through multicast. The following figure illustrates this dual delivery.

Broadcasting by Multiple Methods

**Tip:** Server-to-server multicasting requires a multicast-enabled network just like server-to-player multicasting. To configure server-to-server multicasts, you simply select multicast as your transport method when configuring transmitters and receivers.

Creating Redundant Streams

Helix Server supports encoder redundancy for all media formats. This redundancy carries over automatically with splitting, requiring no special configuration for transmitters and receivers. As shown in the following illustration, separate encoders deliver the same stream (delimited with an integer, as in live.mp4.1 and live.mp4.2) to the transmitter. If the primary stream fails, the transmitter uses the backup stream. It sends the receiver one stream, which may come from either the primary or the backup encoder.
Simple Redundant Source for Splitting

For More Information: Refer to the Helix Administrator online help topic Helix Administrator > Broadcasting > Broadcast Redundancy for details about broadcast redundancy.

Transmitter Redundancy

The setup in the preceding illustration provides a single level of encoder redundancy. You can increase redundancy within a splitting arrangement by adding transmitter redundancy, as shown in the following illustration. Here, both the primary and backup encoders send streams to two Helix Servers that each split the stream to three receivers. (Helix Producer and SLTA can push the same stream to multiple Helix Servers.) Under normal conditions, each receiver gets a version of live.mp4 from each transmitter.
Redundant Sources and Redundant Transmitters

If the primary encoder fails, both transmitters switch to the stream from the backup encoder. Again, the receivers get two streams, one from each transmitter. Note that in this configuration, each receiver still receives a stream even if one encoder and one transmitter fail. This provides both encoder and transmitter redundancy. Three out of the four encoder and transmitter components would have to fail for the entire broadcast to fail.

Transport Redundancy

The following illustration shows multiple encoders and delivery methods used to provide encoder and transport redundancy. The primary and backup encoders both unicast a separate stream to each receiver, as well as multicast the broadcast stream to all receivers. Each receiver gets four streams. It uses the primary live.mp4.1 stream as long as those packets arrive by either unicast or multicast. If the primary encoder fails, or its transport methods are blocked, each receiver switches to the live.mp4.2 backup sent over unicast or multicast.
Note that the receivers in the preceding illustration could also function as transmitters that split streams to other receivers, as described in “Transmitter Redundancy” on page 225. This would provide three layers of redundancy at the encoder, transmitter, and transport levels. Although such a complex arrangement and high degree of redundancy is generally not necessary, RealNetworks components provide support for all of these layers, which you can put together as needed.
Chapter 13: SIMULATED LIVE BROADCASTS

The Simulated Live Transfer Agent (SLTA) is a Helix Server utility that allows you to stream a prerecorded clip or a broadcast archive as if it were a live event. Using SLTA, you can deliver encore presentations, or simulate radio or TV programming using any number of clips. This chapter covers the basic and advanced modes of SLTA, explaining how to set up SLTA, create playlists, and run your simulated live broadcast.

Understanding Simulated Live Broadcasts

SLTA is a command line tool installed with Helix Server. Running on Windows or UNIX, it sends a media stream to Helix Server, which then broadcasts the stream to media players. SLTA supports all broadcasting features:

- **multicasting**—SLTA can deliver the same stream to multiple Helix Servers on a multicast-enabled network.

- **push delivery**—SLTA can initiate the broadcast by contacting one or more Helix Servers and delivering the stream. If you are unicasting rather than multicasting to each Helix Server, the number of connections is limited primarily by the outgoing bandwidth on the machine that hosts SLTA.

- **pull delivery**—Helix Server can initiate the broadcast by contacting SLTA and acquiring the stream when the first media player requests the broadcast. SLTA supports connections from multiple pull-enabled receivers. The number of connections is limited primarily by the outgoing bandwidth on the machine that hosts SLTA.

- **broadcast redundancy**—You can run separate SLTA applications on different machines to create primary and backup streams.

*For More Information:* For more on splitting, see Chapter 12.
Broadcast Formats

SLTA can simulate an audio or video broadcast in RealMedia, QuickTime, MP3, MPEG-4, 3GPP, F4V, AU, WAV, or Windows Media format. The Flash FLV format is not supported. SLTA does not read the RealNetworks-proprietary, multi-rate container file format (.mrc).

Note: Files are subject to the maximum size imposed by 32-bit operating systems. On 64-bit operating systems, files can be any size.

Basic and Advanced Modes

SLTA has two modes: basic and advanced. Basic mode simulates Helix Push broadcasting in a RealNetworks encoder. It allows you to push the broadcast stream to a single Helix Server. It automatically sets to predefined values certain stream delivery variables that are configurable in advanced mode. For example, basic mode always uses a 10% forward error correction rate and a 30-second data acquisition interval.

In advanced mode, SLTA functions like a transmitter in a splitting setup, sending a stream to one or more Helix Servers configured as receivers. The receivers then broadcast the stream to media players. The following table summarizes the features available in basic and advanced modes.

<table>
<thead>
<tr>
<th>Feature or Requirement</th>
<th>Basic</th>
<th>Advanced</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>All supported datatypes</td>
<td>yes</td>
<td>yes</td>
<td>page 230</td>
</tr>
<tr>
<td>Helix Server configured as a receiver</td>
<td>no</td>
<td>yes</td>
<td>page 231</td>
</tr>
<tr>
<td>SLTA configuration file necessary</td>
<td>no</td>
<td>yes</td>
<td>page 235</td>
</tr>
<tr>
<td>pull delivery</td>
<td>no</td>
<td>yes</td>
<td>page 241</td>
</tr>
<tr>
<td>push delivery</td>
<td>yes</td>
<td>yes</td>
<td>page 238</td>
</tr>
<tr>
<td>push delivery to multiple servers</td>
<td>no</td>
<td>yes</td>
<td>page 238</td>
</tr>
<tr>
<td>multicast delivery to multiple servers</td>
<td>no</td>
<td>yes</td>
<td>page 239</td>
</tr>
<tr>
<td>configurable forward error correction</td>
<td>no</td>
<td>yes</td>
<td>page 239</td>
</tr>
<tr>
<td>ignore server resend requests</td>
<td>no</td>
<td>yes</td>
<td>page 239</td>
</tr>
<tr>
<td>configurable metadata transmit rate</td>
<td>no</td>
<td>yes</td>
<td>page 239</td>
</tr>
<tr>
<td>path name preceding stream name</td>
<td>no</td>
<td>yes</td>
<td>page 240</td>
</tr>
</tbody>
</table>

(Table Page 1 of 2)
CHAPTER 13: Simulated Live Broadcasts

SLTA Basic and Advanced Mode Features and Requirements (continued)

<table>
<thead>
<tr>
<th>Feature or Requirement</th>
<th>Basic</th>
<th>Advanced</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLTA-buffered transport</td>
<td>no</td>
<td>yes</td>
<td>page 241</td>
</tr>
<tr>
<td>broadcast redundancy</td>
<td>yes</td>
<td>yes</td>
<td>page 248</td>
</tr>
<tr>
<td>UDP transport</td>
<td>yes</td>
<td>yes</td>
<td>page 251</td>
</tr>
<tr>
<td>TCP transport</td>
<td>yes</td>
<td>yes</td>
<td>page 251</td>
</tr>
<tr>
<td>clip playlists</td>
<td>yes</td>
<td>yes</td>
<td>page 242</td>
</tr>
<tr>
<td>shuffle play</td>
<td>yes</td>
<td>yes</td>
<td>page 251</td>
</tr>
<tr>
<td>clip title, author, and copyright overrides</td>
<td>yes</td>
<td>yes</td>
<td>page 251</td>
</tr>
<tr>
<td>wallclock synchronization</td>
<td>yes</td>
<td>yes</td>
<td>page 251</td>
</tr>
</tbody>
</table>

(Table Page 2 of 2)

Helix Server Setup

Using SLTA requires a small amount of setup on Helix Server.

Basic Mode Configuration

SLTA’s basic mode is authenticated, so you need to set up a user name and password on Helix Server:

- Although you can use the name and password that logs you into Helix Administrator, RealNetworks recommends that you define a new name and password.

- You select the encoder realm that contains the SLTA user name and password under **Broadcasting > Helix Push**.

  **Tip:** In basic mode, SLTA uses the port range defined on the Helix Push broadcasting page (**Broadcasting > Helix Push**). The default settings are generally sufficient.

Advanced Mode Configuration

When you run SLTA in advanced mode, you configure Helix Server as a receiver. RealNetworks recommends that you do this before you define your SLTA configuration file.

Command Line Operation

When you’re ready to broadcast, you run SLTA from the command line, specifying the configuration file, the clip or playlist, and any additional
options. For each broadcast, you define a single stream name, such as encore.rm, which is used in place of the actual clip or playlist name. When viewers click the link to this stream, they join the broadcast in progress.

Note: To run SLTA, you must be able to open a command prompt on your operating system, and navigate to a specific directory. This chapter does not explain how to perform these functions.

SLTA Quick Start Tutorials

The following tutorials are optional. They introduce you to the procedures for simulating a broadcast, using the smallest set of variables and commands required to configure and run SLTA. Once you understand the overall operation of SLTA, you’ll more easily pick up the many additional configuration features and command line options that you can use.

Note: In the following tutorials, you run SLTA on your Helix Server machine. In a production environment, however, RealNetworks recommends that you run SLTA and Helix Server on separate machines.

Quick Start for SLTA Basic Mode

In this tutorial you run SLTA in basic mode.

➤ To simulate a basic mode broadcast with a prerecorded clip:

1. Anywhere on your Helix Server computer, create a directory named Simulate, and copy the following files to it:
   a. The slta.exe (Windows) or slta (UNIX) file from the Helix Server Bin directory.
   b. The slta.bat (Windows) or slta.sh (UNIX) file from the Helix Server Bin directory.
   c. The realvideo10.rm clip from the Helix Server Content directory. You can use another RealMedia clip if you wish.

The location of Helix Server in a default installation on Windows is C:\Program Files\Real\Helix Server.
2. Open a command prompt and navigate to your `Simulate` directory. Enter one of the following commands to transmit the `realvideo10.rm` clip (or another clip if you wish) under the stream name `live.rm`. On Windows:

```bash
slta.bat 127.0.0.1 80 name password live.rm realvideo10.rm
```

On UNIX:

```bash
slta.sh 127.0.0.1 80 name password live.rm realvideo10.rm
```

- You need to substitute the actual Helix Server address for `127.0.0.1` if your files are not on the same machine as Helix Server, or you’re not using the local host address.

- The `80` entry refers to the standard HTTP port, which may be different on your Helix Server.

*For More Information:* See the Helix Administrator online help topic `Helix Administrator > Server Setup > Ports > Port Assignments` for port information.

- For this exercise, use the name and password you use to log into Helix Administrator.

3. Start RealPlayer, give the `File > Open` command, and enter the following URL:

```bash
rtsp://127.0.0.1/broadcast/live.rm
```

You need to substitute the actual Helix Server address if your RealPlayer is not on the same machine as Helix Server, or you’re not using the local host address. You’ll also need to include the RTSP port number if Helix Server does not use port 554.

4. When you finish testing, stop SLTA from the command line by pressing `Ctrl+c` on Windows, or giving the `kill` command with the process ID on UNIX. After a few seconds, the broadcast stops playing in RealPlayer.

**Quick Start for SLTA Advanced Mode**

Follow the steps in this tutorial to run SLTA in advanced mode, creating both a transmitter and a receiver on your Helix Server computer.

- To simulate an advanced mode broadcast with a prerecorded clip:

1. Anywhere on your Helix Server computer, create a directory named `Simulate`, and copy the following files to it:
a. The slta.exe (Windows) or slta (UNIX) file from the Helix Server Bin directory.

b. The slta.bat (Windows) or slta.sh (UNIX) file from the Helix Server Bin directory.

c. The realvideo10.rm clip from the Helix Server Content directory. You can use another RealMedia clip if you wish.

The location of Helix Server in a default installation on Windows is C:\Program Files\Real\Helix Server.

2. Through Helix Administrator, configure Helix Server as a receiver. Enter the following values, accepting the default values for any setting not listed.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitter Name</td>
<td>Simulation</td>
<td>A transmitter can have any name, but the name should not include spaces.</td>
</tr>
<tr>
<td>Transmitter Address</td>
<td>127.0.0.1</td>
<td>Using this value requires that your Helix Server binds to the local host address, which it does by default. If you’ve changed this, use the actual network address or host name.</td>
</tr>
<tr>
<td>Transmitter Netmask</td>
<td>32 bits</td>
<td>This is required when using local host. It’s not required if using a full address.</td>
</tr>
<tr>
<td>Security Type</td>
<td>None</td>
<td>This option requires no password. In an actual broadcast, RealNetworks recommends that you always use Basic security.</td>
</tr>
</tbody>
</table>

3. In a text editor, enter the following XML-based syntax, saving the file as transmit.cfg in your Simulate directory. You’ll need to change the Address variable if you’re not using the local host address. Also, make sure that the PortRange variable matches the setting on the receiver. Values must be quoted, and variable tags must end with a slash (/).

```
<List Name="BroadcastDistribution">
  <Var SourceName="Simulation"/>
  <List Name="Destinations">
    <List Name="TestReceiver">
      <Var PathPrefix="*"/>
      <Var PortRange="30001-30020"/>
      <Var Address="127.0.0.1"/>
      <Var Protocol="udp/unicast"/>
      <List Name="Security">
```
CHAPTER 13: Simulated Live Broadcasts

Note: This configuration sets up SLTA as a push transmitter. The variables for pull splitting, which are not shown in the preceding example, are different.

4. Open a command prompt and navigate to your Simulate directory. Enter one of the following commands to transmit the realvideo10.rm clip (or another clip if you wish) under the stream name live.rm. On Windows:

slta.bat -c transmit.cfg live.rm realvideo10.rm

On UNIX:

slta.sh -c transmit.cfg live.rm realvideo10.rm

5. Start RealPlayer, give the File > Open command, and enter the following URL:

rtsp://127.0.0.1/broadcast/Simulation/live.rm

You need to substitute the actual Helix Server address if your RealPlayer is not on the same machine as Helix Server, or you’re not using the local host address. You’ll also need to include the RTSP port number if Helix Server does not use port 554.

6. When you finish testing, stop SLTA from the command line by pressing Ctrl+c on Windows, or giving the kill command with the process ID on UNIX. After a few seconds, the broadcast stops playing in RealPlayer.

Configuring SLTA for Advanced Mode

You do not need to configure SLTA to run in basic mode. In advanced mode, SLTA relies on an XML-based configuration file for instructions on transmitting to each Helix Server receiver. When you run SLTA, you indicate which configuration file to use. This allows you to define multiple configuration files for different transmission scenarios. The following sections explain how to set up your configuration file for advanced mode broadcasting.

Tip: Set up Helix Server as a receiver before you write your configuration file.
Using the Configuration Template

The main Helix Server installation directory contains a template for the SLTA configuration file, slta.cfg. You can edit this template with any text editor, and save new configuration files as plain text under any name. The .cfg extension is recommended but not necessary. The following example shows the contents of the slta.cfg template:

```xml
<List Name="BroadcastDistribution">
  <Var SourceName="ExampleSourceName"/>
  <List Name="Destinations">
    <List Name="ExampleName">
      <Var PathPrefix="*"/>
      <Var PortRange="30001-30020"/>
      <Var AcquisitionDataInterval="30"/>
      <Var FECLevel="0"/>
      <Var SureStreamAware="0"/>
      <Var BufferlessTransport="1"/>
      <Var LocalAddress="0.0.0.0"/>
      <Var Address="127.0.0.1"/>
      <Var TTL="16"/>
      <Var ResendSupported="0"/>
      <Var Protocol="udp/unicast"/>
      <List Name="Security">
        <Var Type="Basic"/>
        <Var Password="ExamplePassword"/>
      </List>
    </List>
  </List>
</List>

<List Name="Pull Settings">
  <List Name="PullSource1">
    <List Name="Security">
      <Var Type="Basic"/>
      <Var Password="ExamplePassword"/>
    </List>
    <Var SureStreamAware="0"/>
    <Var ListenPort="2030"/>
    <Var PathPrefix="/"/>
    <Var LocalAddress="0.0.0.0"/>
  </List>
</List>
</List>
```
CHAPTER 13: Simulated Live Broadcasts

Setting Basic Transmitter Properties

The configuration template has an outer list named BroadcastDistribution, which you should not change. This main list contains a variable for the transmitter name, and separate sections for push splitting and pull splitting variables. Typically, you need to define only one of these sections depending on the type of splitting you use:

```
<List Name="BroadcastDistribution">
  <Var SourceName="ExampleSourceName"/>
  <List Name="Destinations">
    <List Name="ExampleName">
      ...all push splitting variables...
    </List>
  </List>
  <List Name="Pull Settings">
    ...all pull splitting variables...
  </List>
</List>
```

**Tip:** When using push splitting, you can delete the pull splitting variables, and vice versa. This is not necessary, though, and if you do so, be careful not to delete an element you need, such as a closing `</List>` tag. Missing elements will cause an error when you run SLTA.

Naming the Transmitter

Whether you use push splitting or pull splitting, you define a name for the SLTA transmitter in the SourceName variable at the top of the file:

```
<Var SourceName="ExampleSourceName"/>
```

If you plan to run multiple instances of SLTA on the same machine simultaneously, each configuration file should define a different source name. This name appears in Web page links for push transmissions, so it should not include spaces. Here is an example:

```
<Var SourceName="BroadcastEncore"/>
```

**Tip:** It helps you to keep track of transmitters and receivers if you use same name you defined for SLTA in the Broadcast Transmitters box on the Helix Server receiver.
Defining Push Splitting

In push splitting, SLTA initiates the connection, contacting one or more Helix Server receivers, and delivering the simulated live stream. The push splitting section starts with the Destinations list, which encompasses a single receiver definition predefined as ExampleName. Change the name to any value that describes the Helix Server receiver. Here’s an example:

```xml
<List Name="Destinations">
  <List Name="Sydney Receiver">
    ...all push splitting variables...
  </List>
</List>
```

Note: This name is for your convenience, and is not used in the broadcast. The name cannot contain a period character ("."), however.

Specifying Multiple Receivers

If you intend to deliver the same simulated live broadcast to more than one Helix Server, duplicate the receiver list within the Destinations list, and give the second receiver a new name, as shown in the following example. Each receiver list then defines the same set of variables but with different values as appropriate for each receiver. You can set up as many receivers as necessary:

```xml
<List Name="Destinations">
  <List Name="Sydney Receiver">
    ...all push splitting variables for the first receiver...
  </List>
  <List Name="Tokyo Receiver">
    ...all push splitting variables for the second receiver...
  </List>
</List>
```

SLTA can unicast a separate stream to each receiver, or multicast a single stream on a multicast-enabled network. As explained in Chapter 12, you can also use a Helix Server receiver to transmit (or “split”) the SLTA stream to any number of other receivers. This may be a better solution if your SLTA transmitter has limited outgoing bandwidth and you cannot multicast from SLTA to all of the receivers.

Tip: Using virtual paths, you can define a single configuration file for multiple receivers but deliver a stream only to certain
receivers when you run SLTA. For more information, see “Directing Streams through Paths” on page 240.

Setting Push Configuration Values

The following table describes the SLTA push splitting variables you define for each receiver you use. Several transmitter variables correspond to receiver variables that must have a similar value. Minimally, you need to ensure that the Address, PortRange, Protocol, and Password variables match the settings on the receiver.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathPrefix</td>
<td>* path</td>
<td>Streams all broadcasts with “*”, or specifies a virtual path name.</td>
</tr>
<tr>
<td>PortRange</td>
<td>port-port</td>
<td>Indicates the range of ports set up on the receiver for data transmission.</td>
</tr>
<tr>
<td>AcquisitionData Interval</td>
<td>seconds</td>
<td>Sets the frequency that the transmitter sends metadata information. Valid range is 0 to 60.</td>
</tr>
<tr>
<td>FECLevel</td>
<td>integer</td>
<td>Sets a percentage of corrective packets sent. The valid range is 0 to 100.</td>
</tr>
<tr>
<td>BufferlessTransport</td>
<td>0</td>
<td>Uses the clip’s timestamp information. The value 0 is “no,” 1 is “yes.” (Always set this value to 1.)</td>
</tr>
<tr>
<td>LocalAddress</td>
<td>address</td>
<td>Specifies the address or host name of the SLTA transmitter.</td>
</tr>
<tr>
<td>Address</td>
<td>address</td>
<td>Indicates the address or host name of the receiver.</td>
</tr>
<tr>
<td>TTL</td>
<td>integer</td>
<td>Sets the time to live for multicasting.</td>
</tr>
<tr>
<td>ResendSupported</td>
<td>0</td>
<td>Ignores (0) or honors (1) the receiver’s resend requests.</td>
</tr>
<tr>
<td>Protocol</td>
<td>protocol</td>
<td>Sets the protocol used in transmitting streams. Choices are udp/unicast, udp/multicast, and tcp.</td>
</tr>
<tr>
<td>Type</td>
<td>Basic</td>
<td>Sets the type of security used.</td>
</tr>
<tr>
<td>Password</td>
<td>password</td>
<td>Provides a password for Basic security.</td>
</tr>
</tbody>
</table>

**Tip:** It’s OK to ignore variables for features that you do not use. For example, TTL is used only with multicasts, so its setting does not affect unicasts.
Directing Streams through Paths

To broadcast all streams to all defined receivers, leave PathPrefix set to its default value of “*” for each receiver. As described in “Splitting Different Streams to Different Receivers” on page 221, however, you can use virtual path names and stream names to direct different broadcasts to different receivers. Suppose that you define three different receivers, and give each a different PathPrefix value as shown in the following example:

```xml
<List Name="Destinations">
  <List Name="Sydney Receiver">
    <Var PathPrefix="*"/>
  </List>
  <List Name="Tokyo Receiver">
    <Var PathPrefix="news/"/>
  </List>
  <List Name="Bombay Receiver">
    <Var PathPrefix="news/hourly.rm"/>
  </List>
</List>
```

The Sydney receiver, which uses the default path “*”, gets all streams broadcast by SLTA using this configuration file. The Tokyo receiver gets only the streams that specify the news/ virtual path along with the stream name, which can be anything, when you start the broadcast. The Bombay receiver gets only the broadcast streams named news/hourly.rm.

The advantage of using virtual paths is that you can use one configuration file to deliver different streams to different receivers at different times. Instead of creating a separate configuration file for each receiver, you can define all receivers in one configuration file, then use the virtual paths to determine which receivers get which streams when you run SLTA.

Alternatively, you can create multiple configuration files to direct broadcasts to different receivers. Then, instead of specifying a path when starting the broadcast, you select the appropriate configuration file. Either method can create the same results, and you should choose the method that you find more convenient.

Tip: As noted in the table “SLTA Pull Splitting Variables” on page 242, pull splitting definitions can include a virtual path prefix. There is generally no need to set this path, however. In
pull splitting, SLTA responds to requests. It does not actively
direct streams as it does in push splitting. You can therefore
leave the default path prefix of “/” set for pull splitting.

**For More Information:** The section “Starting SLTA” on page 246
explains how to indicate the path when you start the
broadcast.

**Using Bufferless Transport**
For most clip types, you should leave `BufferlessTransport` set to its default value
of “1”. In this mode, SLTA transmits the clip according to the clip’s internal
timestamp information, which is appropriate for most streaming audio and
video formats. If you set this variable to “0”, SLTA buffers clip data itself and
builds its own broadcast scheduler, which increases broadcast latency and
processor overhead. You can enable bufferless transport only for clips that
stream at a constant bit rate.

**Defining Pull Splitting**
In pull splitting, Helix Server initiates the connection, contacting SLTA to
acquire the stream when the first media player requests the simulated live
broadcast. The pull splitting section starts with the `Pull Settings` list, which
encompasses a single definition predefined as `PullSource1`. Change this
definition name, which is not used in broadcasts, to anything that describes
your SLTA transmitter (do not use a period character is the name):

```xml
<List Name="Pull Settings">
  <List Name="NewsEncoreTransmitter">
    <List Name="Security">
      <Var Type="Basic" />
      <Var Password="ExamplePassword" />
    </List>
    <Var SureStreamAware="0" />
    <Var ListenPort="2030" />
    <Var PathPrefix="/" />
    <Var LocalAddress="0.0.0.0" />
  </List>
</List>
```

Any number of receivers can pull the same stream from a single SLTA
transmitter. You therefore do not need to replicate the pull splitting variables
for each receiver in use, as you do with push splitting. You need to define
multiple configuration files only if you intend to deliver different streams to one or more receivers at the same time. You then run multiple instances of SLTA, using a different configuration file, stream name, and playlist for each instance.

Setting Pull Configuration Values

The following table describes the SLTA pull splitting configuration variables you define. Most configuration is done on the receiver, which must be enabled for pull splitting.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathPrefix</td>
<td>/</td>
<td>path</td>
</tr>
<tr>
<td>ListenPort</td>
<td>port</td>
<td>Set the port where SLTA listens for stream requests.</td>
</tr>
<tr>
<td>LocalAddress</td>
<td>address</td>
<td>Specifies the address or host name of the SLTA transmitter.</td>
</tr>
<tr>
<td>Type</td>
<td>Basic</td>
<td>None</td>
</tr>
<tr>
<td>Password</td>
<td>password</td>
<td>Provides a password for Basic security.</td>
</tr>
</tbody>
</table>

Creating a Playlist

You do not need to write a playlist to broadcast a single clip. If you have a series of clips to broadcast, though, you list them in sequence in the playlist. A playlist can contain any number of clips. When you run SLTA, you can play the clips in the order they’re listed, or in random order (shuffle play). Refer to “Using SLTA Options” on page 249 for information about command line options that affect playlists.

Writing a Basic Playlist

In a text file (file extension .txt), list on a separate line each file that you want SLTA to stream. If the files do not reside in the same directory as the playlist, include either their full paths, or paths relative to the location of the playlist. In the following example, clips reside in the same directory as the playlist:
Warning! All files in the playlist must be in the same media format and must be encoded at the same bit rate (or set of bit rates for multi-rate content).

Adding Title, Author, and Copyright Information

Prerecorded clips often have title, author, and copyright information encoded into them. Through the playlist, you can override this information on a clip-by-clip basis. Or, you can set the same title, author, and copyright values for all clips. RealPlayer users display this information through the clip info command (\texttt{Ctrl+i}). Information about a certain clip displays only as that clip is broadcast.

Tip: Although a playlist isn’t required to broadcast a single clip, you can write a playlist containing a single clip just to use the title, author, and copyright overrides.

Setting Information for All Clips

To include the same title, author, and copyright information for every clip, add title, author, and copyright tags to the beginning of the playlist. This information overrides any encoded information in the clip. End each tag with a colon, as shown in the following example:

\begin{verbatim}
title: Annual Report
author: Chris Lee, Executive Assistant
copyright: Copyright 2006
...clip1...
...clip2...
\end{verbatim}

Using Individual Clip Parameters

If you append individual title, author, and copyright parameters to a clip in the playlist, the specified information overrides the playlist values described above, as well as values encoded in the clip. You can use any combination of title, author, and copyright parameters for each clip. Use double quotation marks around values. Separate the first parameter from the clip file name
Mixing Information Styles

You can mix the different ways of delivering clip information. When you do this, keep in mind that clip information is used in the following order:

1. Information specified for each clip in the playlist.
2. Information specified for all clips at the start of the playlist.
3. Information encoded in each clip.

In the following playlist example, suppose that each clip has a title, author, and copyright value encoded directly in the clip:

copyright: (c) 2001-2006
CompanyLogo.rm?title=“Welcoming Remarks”
Welcome.rm?title=“Welcoming Remarks”
President.rm?title=“President’s Address”
Treasurer.rm?title=“Treasurer’s Summary”
Conclusions.rm?title=“The Future is Bright”

Within this playlist, the title parameter for each clip overrides the titles encoded in the clips. The playlist’s copyright parameter sets the same copyright value for all clips in the list, also overriding the clips’ encoded values. No author values are specified, though, so each clip uses its encoded author text.

Running SLTA

This section describes the SLTA command line syntax, environment variables, and options. To run SLTA in advanced mode, you must first configure your Helix Server receiver, set up your SLTA configuration file, and, optionally, write your playlist.

Running SLTA on a Different Machine

If you run SLTA infrequently, you can use your Helix Server machine to run the SLTA executable. In a production environment in which you run SLTA continually, however, RealNetworks recommends that you install SLTA on a machine other than the Helix Server machine. This distributes the processing load between SLTA and Helix Server.
**Note:** To install SLTA on a different machine, contact your RealNetworks representative to obtain Helix Live Transmitter. This product installs SLTA on a selected machine or multiple machines. Note, too, that you can run SLTA and Helix Server on different operating systems.

### Setting Environment Variables

Running SLTA on your Helix Server machine requires that you set two environment variables. If you plan to use SLTA on your Helix Server machine frequently, set the following environment variables permanently. This lets you run the SLTA executable file directly:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLTA_PLUGIN_PATH</td>
<td>Full path to the directory that holds Helix Server plug-ins. This is typically the Plugins subdirectory of the main Helix Server installation directory.</td>
</tr>
<tr>
<td>SLTA_SUPPORT_PATH</td>
<td>Full path to the directory that holds Helix Server libraries (DLLs). This is typically the Lib subdirectory of the main Helix Server installation directory.</td>
</tr>
</tbody>
</table>

**Tip:** If you plan to use SLTA only occasionally, you can run the provided batch (Windows) or shell (UNIX) file described in “Starting SLTA” on page 246 to set the variables for your current session.

### Setting Variables on Windows

To set SLTA environment variables permanently on a default installation of Helix Server on Windows, add the following values to your environment:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLTA_PLUGIN_PATH</td>
<td>C:\Program Files\Real\Helix Server\Plugins</td>
</tr>
<tr>
<td>SLTA_SUPPORT_PATH</td>
<td>C:\Program Files\Real\Helix Server\Lib</td>
</tr>
</tbody>
</table>

If you installed Helix Server in a location other than the default, modify the C:\Program Files\Real\Helix Server portion of the paths above to the actual base path where Helix Server resides.

### Setting Variables on UNIX

On UNIX platforms, add the environment variables to the .profile file or the shell personalization files (.bashrc or .cshrc) of the shell under which SLTA runs. For example:
SLTA_PLUGIN_PATH=/usr/local/Real/HelixServer/Plugins
export SLTA_PLUGIN_PATH

SLTA_SUPPORT_PATH=/usr/local/Real/HelixServer/Lib
export SLTA_SUPPORT_PATH

**Starting SLTA**

On Helix Server, SLTA files are located in the Bin subdirectory of the main Helix Server directory. The file you run depends on your operating system, and whether you’ve set environment variables permanently.

<table>
<thead>
<tr>
<th>SLTA Executable to Use</th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>With variables set permanently.</td>
<td>slta.exe</td>
<td>slta</td>
</tr>
<tr>
<td>Without having set variables permanently.</td>
<td>slta.bat</td>
<td>slta.sh</td>
</tr>
</tbody>
</table>

**Starting SLTA in Basic Mode**

From a command prompt, use the following syntax to run SLTA in basic mode.

slta[.ext] server_address HTTP_port name password stream_name.ext
clip.ext|playlist.txt [-options]

- Use the appropriate SLTA executable file as described in the preceding table.
- Next, specify the network address (DNS name, IPv4 address, or IPv6 address) and HTTP port of your Helix Server.

**For More Information:** See the Helix Administrator online help topic Helix Administrator > Server Setup > Ports > Port Assignments for port information.

- The name and password entries give the encoder access to Helix Server. For more information, see “Helix Server Setup” on page 231.
- For stream_name.ext, specify the name of the simulated live stream. This name appears in links to the broadcast, and uses the appropriate media extension, such as .rm for a RealMedia broadcast. When broadcasting QuickTime or MPEG, use .mov or an appropriate MPEG extension (such as .mpeg, .3gp, or .mp4) for the stream name. You do not need to use an SDP file.
• Next, you specify either the single clip you want to broadcast, or the playlist you wrote according to the instructions in “Creating a Playlist” on page 242. If the clip or playlist is not in the SLTA directory, specify the full path, or a path relative to the location of the SLTA executable file.

• Optionally, you can use command line options as described in “Using SLTA Options” on page 249. For example, SLTA loops a clip or playlist by default. With command line options, you can specify how many repetitions to play.

Basic Mode Example
If you set the SLTA environment variables permanently on Helix Server, you would use a command such as the following to broadcast a single RealMedia archive (awards.rm) as a simulated live event (encore.rm), using no additional SLTA options:

```
slta.exe helixserver.example.com 80 simulator k56weiq9 encore.rm awards.rm
```

Notes on Running SLTA in Basic Mode
• As SLTA begins the broadcast, it displays the progress indicator described in “Monitoring and Stopping SLTA” on page 252.

• SLTA contacts the receiver immediately to deliver the stream.

• You can verify connections on Helix Server using the Server Monitor described in the Helix Administrator online help topic Helix Administrator > Logging and Monitoring > Server Monitor.

Starting SLTA in Advanced Mode
From a command prompt, use the following syntax to run SLTA in advanced mode.

```
slta[.ext] -c config_file.cfg stream_name.ext clip.ext|playlist.txt [-options]
```

• Use the appropriate SLTA executable file as described in “Starting SLTA” on page 246.

• For `config_file.cfg`, give the full or relative path to a configuration file you created according to instructions in “Configuring SLTA for Advanced Mode” on page 235.

• For `stream_name.ext`, specify the name of the simulated live stream. This name appears in links to the broadcast, and uses the appropriate media extension, such as .rm for a RealMedia broadcast. When broadcasting QuickTime or MPEG, use .mov or an appropriate MPEG extension (such
as .mpeg, .3gp, or .mp4) for the stream name. You do not need to use an SDP file.

If you defined virtual paths for receivers, as described in “Directing Streams through Paths” on page 240, you can include a virtual path in front of the stream name, as in news/live.rm. This path, which does not correspond to any directory on the machine, uses a Web-style forward slash even on Windows.

• Next, you specify either the single clip you want to broadcast, or the playlist you wrote according to the instructions in “Creating a Playlist” on page 242. If the clip or playlist is not in the SLTA directory, specify the full path, or a path relative to the location of the SLTA executable file.

• Optionally, you can use command line options as described in “Using SLTA Options” on page 249. For example, SLTA loops a clip or playlist by default. With command line options, you can specify how many repetitions to play.

**Advanced Mode Example**

If you set the SLTA environment variables permanently on Helix Server, you would use a command such as the following to broadcast a single RealMedia archive (awards.rm) as a simulated live event (encore.rm), using no additional SLTA options:

```bash
slta.exe -c transmit.cfg encore.rm awards.rm
```

**Notes on Running SLTA in Advanced Mode**

• As SLTA begins the broadcast, it displays the progress indicator described in “Monitoring and Stopping SLTA” on page 252.

• If you set up push splitting, SLTA contacts the receiver immediately to deliver the stream.

• If you’re using pull splitting, SLTA does not deliver the stream until the receiver requests it.

• You can verify connections on the Helix Server receiver using the Server Monitor described in the Helix Administrator online help topic *Helix Administrator > Logging and Monitoring > Server Monitor*.

**Running Redundant SLTA Encoders**

In both basic and advanced modes, SLTA supports redundant encoders. This allows the Helix Server receiver to switch to a backup stream if the primary
CHAPTER 13: Simulated Live Broadcasts

encoder stream fails. Although you can run two instances of SLTA on the same machine, this provides process redundancy only. For true redundancy, run an instance of SLTA on two separate machines that use different power supplies and network connections.

To set up SLTA redundancy, install SLTA from the Helix Live Transmitter package on your backup machine. Copy over the configuration file, the playlist (if used), and the broadcast clip or clips. Because the content is prerecorded, you want to start SLTA on both machines at the same time, or as close as possible. You may want to set up a script appropriate for your operating system that launches both SLTA processes on the two machines simultaneously.

The only difference between the two SLTA instances is the stream name used in the command that starts SLTA. Each stream name needs a delimiter that identifies it as the primary or the backup. The following example builds on an example in a preceding section by appending .1 to the stream name to identify it as the primary stream:

```
slta.exe -c transmit.cfg encore.rm.1 awards.rm
```

With SLTA on your backup machine, you add .2 to the stream name:

```
slta.exe -c transmit.cfg encore.rm.2 awards.rm
```

Using SLTA Options

SLTA provides optional command-line arguments that affect the simulated live broadcast. The options appear last on the command line in any order, each option preceded by a hyphen. Here is an example:

```
slta.exe -c transmit.cfg encore.rm playlist.txt -e -n10 -r
```

The following table summarizes the options. Some options are useful only with playlists. Unless noted otherwise, all options are available in basic or advanced mode.

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>Use the specified configuration file (advanced mode only)</td>
<td>page 247</td>
</tr>
<tr>
<td>e</td>
<td>Disable playlist title, author, and copyright information.</td>
<td>page 251</td>
</tr>
<tr>
<td>f</td>
<td>Force playlist title, author and copyright updates when broadcasting RealMedia clips.</td>
<td>page 251</td>
</tr>
</tbody>
</table>

(Table Page 1 of 2)
Modifying the Playback Order

By default, SLTA transmits a clip or playlist in a continuous loop. If you specify just one file, for example, SLTA continuously transmits that file until you stop the broadcast. You can use command line `-n N` and `-r` options to modify the default playback.

**Specifying the Number of Clips to Play**

The `-n N` switch specifies the total number of files to play. To play a clip just once, for instance, you specify `-n 1`. To play it twice, use `-n 2`. For a single playback of a playlist that contains 3 clips, for example, you specify `-n 3`. A value such as `-n 5` means to play the single clip five times, or to play the first five clips on the playlist (not to play each clip in the playlist five times). If you use `-n 5` for a playlist that has just three clips, SLTA plays the three clips in order, then returns to the start of the list to play the first two clips again, for a total of five clips played.

**Changing a Playlist During a Broadcast**

When SLTA loops a playlist, either continuously (the default behavior) or a predetermined number of times because of the `-n N` option, it reads the playlist each time it starts a loop. This allows you to change the playlist during the broadcast. Simply edit the existing playlist, or replace the playlist with another of the same name. SLTA uses the modified playlist after it has played all of the clips in the current playlist.

**Tip:** This feature lets you stream long broadcasts without writing a single, long playlist. Your first playlist might last an hour. During that hour of the broadcast, you can change the playlist to specify the clips that play during the second hour, and so on.
Using Shuffle Play

When you use a playlist, you can create shuffle play by including the \(-r\) option. If you use just this option without \(-nN\), the playlist cycles continuously, playing clips in a random order during each cycle. You can use both the \(-r\) and \(-nN\) switches to cycle randomly through the playlist a certain number of times. For example, \(-r\) and \(-n10\) cause SLTA to transmit a playlist of five clips twice, with clips playing in a random order each time.

Forcing RTP Packetization for AAC Audio

Including the \(-fp\) option causes SLTA to ignore the hint tracks for MPEG-4 clips and packetize them using RTP packetizers. Do this when streaming clips that use \texttt{mpeg4-generic} (ISMA) packetization for AAC audio that will be streamed to Flash clients.

\textbf{Note:} Including this option is generally safe for all other types of media clients receiving the broadcast.

Disabling Title, Author, and Copyright Overrides

If you use the \(-e\) option when broadcasting any media format, SLTA does not use the title, author, and copyright information included in the playlist. The media player still displays any title, author, and copyright information encoded in the clips, however.

The \(-f\) option preserves playlist overrides, which are described in “Adding Title, Author, and Copyright Information” on page 243. You generally do not need to include this option because it is the default for RealMedia broadcasts. Additionally, using this option when broadcasting content in a format other than RealMedia may cause the broadcast to fail.

Using TCP Transport

In basic mode, SLTA opens a UDP connection to Helix Server unless you use the \(-t\) option to force a TCP connection. Although TCP provides a more reliable connection in a lossy environment, it increases the network overhead. In advanced mode, the \(-t\) option is not available because you specify TCP or UDP transport in the SLTA configuration file.

Synchronizing to a Wallclock

When you include the \(-w\) option, SLTA sets the first timestamp of the packet stream to the time defined by the clock on the SLTA computer. This allows Helix Server to synchronize multiple streams.
Monitoring and Stopping SLTA

As SLTA runs, it displays the name of the file it is currently transmitting. A line of asterisks indicates the percentage of the file that has been sent. For example, a row of asterisks that lines up below the number 5 indicates that the file is approximately 50 percent complete. Here is an example:

Transmitting Welcome.rm...
0----1----2----3----4----5----6----7----8----9----10
***********************

SLTA stops automatically and displays the text "Done" when it has finished transmitting all of the files according to the playlist (if used) and the specified options. If it cannot transmit a file in a playlist because it cannot find the file, or the file is encoded in a format different from the preceding files, it skips that file, prints an error message, and transmits the next file.

Ordinarily, you will not need to stop SLTA manually except to halt the broadcast before it completes, or to terminate a continuously looping broadcast. To stop SLTA, press Ctrl+c at the command line from which you started SLTA on Windows. On UNIX, use the kill command with the process ID of the SLTA process.

Linking to the Simulated Live Broadcast

The following sections explain simulated live broadcast link formats in general. They provide example links for MPEG-4 broadcasts. For broadcasts in other formats, you need to use transmitter mount points, client launch utilities, and stream names as appropriate for that format.

For More Information: The Helix Administrator online help explains the basic broadcast URL formats used with different types of media clients.

Writing Basic Mode Links

Links to SLTA broadcasts in basic mode use standard unicast URL formats. A Web page link to an MPEG-4 broadcast in which Helix Server uses the default port 80 for HTTP might look like this:

http://helixserver.example.com/sdpgen/broadcast/archive.mp4

• The /sdpgen/ mount point launches media players.
The standard broadcast mount point is /broadcast/ (or /rtpencoder/ for an RTP Push broadcast). If you are using redundant SLTA transmitters, use the /redundant/ mount point instead.

The requested file, shown above as archive.mp4, is the stream name you specify when starting SLTA. You do not use the actual name of the broadcast clip or playlist.

Creating Push Splitting Links

Links for advanced mode broadcasts use the same format as split broadcasts, which are described in the Helix Administrator online help topic [Helix Administrator > Broadcast Distribution > Splitting Basics > Links to Split Content > Push Splitting Links]. Push splitting links point to the Helix Server receiver but include the name of the SLTA transmitter to identify the broadcast.

A Web page link to an MPEG-4 broadcast in which Helix Server uses the default port 80 for HTTP might look like this:

http://helixserver.example.com/sdpgen/broadcast/Simulated/news/archive.mp4

The link is similar to the basic mode link show above, but includes additional components:

- Following the broadcast mount point, you specify the transmitter name, such as /Simulated/. The name is set in the SourceName variable of the configuration file.


- A virtual path name such as news/ is optional. You include it only if you’ve specified a value other than “*” for the PathPrefix variable in the configuration file.


Writing Pull Splitting Links

As with push splitting, a pull splitting link requests the stream from the receiver and indicates the transmitter where the broadcast originates. The link does not use the transmitter name given in the SLTA configuration file,
however. Instead, it provides the address and listen port of the transmitter so that the receiver can locate the transmitter and pull the broadcast stream.

A Web page link to an MPEG-4 broadcast in which the Helix Server receiver uses the default port 80 for HTTP might look like this:

http://helixserver.example.com/sdpgen/broadcast/pull/
simulator.example.com:2030/news/archive.mp4

- The /sdpgen/ mount point launches the media players.
- The standard broadcast mount point is /broadcast/ (or /rtpencoder/ for an RTP Push broadcast).
- Following the broadcast mount point, you specify the pull splitting mount point defined on the Helix Server receiver, such as /pull/. The name is set in the receiver definition.
- After the pull splitting mount point, you give the transmitter’s address and listen port. The example above is /simulator.example.com:2030/. The SLTA configuration file defines the listen port, as described in “Setting Pull Configuration Values” on page 242.

Note: When broadcasting Windows Media, you need to mask the address and port through an alias. For details, see the Helix Administrator online help topic Helix Administrator > Broadcast Distribution > Splitting Basics > Links to Split Content > URL Aliases.

- A path name such as news/ is optional. You include it only if you’ve specified a value other than “/” for the PathPrefix variable in the configuration file.


- The requested file, shown above as archive.mp4, is the stream name you specify when starting SLTA. You do not use the actual name of the broadcast clip or playlist.
Chapter 14: CHANNEL SWITCHING

Channel switching allows Helix Server to switch the media input for a continuous stream, eliminating the need to set up a separate RTSP session for each stream. Channel switching works with on-demand clips, live streams, and simulated-live streams broadcast by SLTA.

Understanding Channel Switching

A channel is a stream that can be switched to a different media input (a new channel). This switching capability eliminates the need to set up and tear down a user’s RTSP session when the user selects a new stream. As a result, a user can “change channels” without experiencing disruption in playback when a new channel begins. To the user, the succession of streams appears to be a broadcast of user-selected content that ends when a channel finishes streaming and no subsequent channel has been queued.

The following figure illustrates a simple channel switching scenario. Here, the user selects the first channel, “Pop Hits.” The user does not make this choice directly through Helix Server in an RTSP session. Rather, the user interacts with a third-party Web portal that supplies the user’s browser with links to the available channels.

Once the user selects the channel, the Web portal sends the channel choice information to Helix Server as a specially-formatted HTTP or HTTPS URL directed toward a secure port. Helix Server then delivers the media player instructions in an SDP file that enable it to connect to Helix Server and receive the channel. The channel content may be an on-demand clip, a live broadcast, or a simulated-live broadcast.
First Channel Playback

The next figure shows a user changing the streaming content to a second channel, “Jazz.” Again, the user interacts with the Web portal to do this. In this scenario, the change request results from user feedback, such as clicking a link in an on-line guide. However, change requests can occur without user interaction. For example, change requests may result from a predefined playlist that the Web portal manages.

Once the Web portal sends Helix Server an HTTP or HTTPS command to change the channel, Helix Server queues the new source. It can switch to the new source immediately, or after the current source has finished playing. When the new channel starts, it replaces the existing stream within the media player’s RTSP session, providing a seamless switchover between the channel streams.

Second Channel Playback
Third-Party Components

The following sections describe the components that are necessary for channel switching but that are not part of Helix Server. Each section provides details about the requirements that the component must meet to enable channel switching.

Web Portal

A Web portal is a third-party application that provides the user with information about available channels. The portal carries out any billing duties, and submits channel requests to Helix Server according to the workflow described in the section “Channel Switching Workflows” on page 265.

User Browser

Each user must have available a browser that allows the user to select channels by interacting with the Web portal. There are no specific requirements for the browser required by the channel switching system.

Note: A rich media client may fulfill browser functions depending on the player’s capabilities and the means used to deliver channel information.

User Media Players

The user must have an RTSP-based media player that supports the codec used to encode the stream. Because non-standard RTSP commands are not used, most media players compliant with the RTSP standard can participate in channel switching.

Codecs and File Formats

Because a media player cannot switch its decoding method within a media stream, all channel streams must be encoded using the same streaming rates and the same codec (same profile and level).

Tip: Because stream quality can differ across encoders, creating content using the same hardware or software encoder helps to prevent streaming errors.
Note: Clips used for channel switching are subject to the maximum size imposed by 32-bit operating systems. On 64-bit operating systems, files can be any size.

Supported Codecs

Channel switching works with the following codecs:

Video Codecs
- H.264 (recommended)
- H.263
- MPEG-4
- RealVideo

Audio Codecs
- AAC or AAC+ (recommended)
- Enhanced AAC+
- AMR-NB or AMR-WB
- RealAudio

Supported File Formats

Supported file formats are the following:
- MPEG-4 (.mp4 and variants)
- 3GPP (.3gp)
- RealMedia (.ra, .rv, .rm)
- QuickTime (.mov)
- F4V (.f4v)

Note: Channel switching does not function with Flash clients, which use the RTMP protocol. However, you can use the F4V format with RTSP clients.

Unsupported Formats

The following formats are not supported:
- Windows Media
- FLV
• any format that use digital rights management (DRM) protection

Protocols and Transports

Channel switching works only with RTSP-based streams. This includes RTSP streams cloaked as HTTP. MMS streaming, RTMP streaming, and HTTP download are not supported. The network transport can be either UDP or TCP. You can use the RealNetworks RDT data packet format or the standards-based RTP format.

User Authentication

Channel switching is compatible with security features such as username and password authentication. When Helix Server authenticates a user, it validates only the initial RTSP request made by the media player. HTTP or HTTPS commands requesting a switch to a different channel are not authenticated by Helix Server. It is the responsibility of the Web portal to verify that channel switch requests made through the portal have come from the actual users.

Note: User name and password authentication is not viable in all situations, especially for mobile players that do not pass usernames and passwords. Contact your RealNetworks representative for information about additional ways to generate and authenticate secure URLs.

Error Conditions

Attempting any of the following changes when switching from one channel to another causes an error condition that closes the channel:

• Switching from one file format to another, such as from RealMedia (.rm) to MPEG-4 (.mp4).

• Change in the codec used to encode the channel, such as from H.263 to H.264.

• Switching the streaming bit rate. For multi-rate clips, all bit rate encodings must be identical.

• Switching from the RDT packet format to RTP, or vice versa.

• Streaming content located on different servers.

• Switching to a non-existing source.
Tip: If an error occurs, Helix Server can still stream the new channel (see the section “SDP File on Error” on page 264). However, this requires tearing down the user’s existing RTSP session and setting up a new one, thereby eliminating the seamless nature of channel switching.

For More Information: For more about error messages, refer to “HTTP Status Codes” on page 264.

Compatibility with Other Features

The following table summarizes the compatibility of channel switching with other Helix Server features.

<table>
<thead>
<tr>
<th>Other Feature</th>
<th>Compatibility Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduced start-up delay</td>
<td>Reduced start-up delay is compatible with channel switching.</td>
</tr>
<tr>
<td>splitting</td>
<td>Channel switching is compatible with splitting live streams, which Chapter 12 describes. However, there will be a delay if a channel switch request is for a pull-split stream that has not yet been pulled on the receiver that manages the user’s RTSP session.</td>
</tr>
<tr>
<td>live rate adaptation</td>
<td>Live rate adaptation is compatible with channel switching, as long as all channels are encoded at the same streaming rates. See “Live Broadcasts and Rate Control” on page 113 for more information.</td>
</tr>
<tr>
<td>basic logging</td>
<td>To record channel switching events in the access log file, use basic logging style 7.</td>
</tr>
<tr>
<td>advanced logging</td>
<td>You can capture channel switching events using the customized logging templates. Refer to the client properties section of the Helix Administrator online help for information about the FinalSwitch, SessionControlID, and SwitchCount properties.</td>
</tr>
<tr>
<td>Helix Proxy</td>
<td>Helix Proxy delivers switchable streams only in pass-through mode. Because the proxy does not accept HTTP requests, a Web portal must send all channel switch commands to the origin server.</td>
</tr>
<tr>
<td>HTTP proxies</td>
<td>Third-party HTTP proxies can be used to proxy the HTTP commands for the Web portal.</td>
</tr>
</tbody>
</table>
Channel Controller

A Web portal issues channel switch commands to the channel controller on Helix Server. Commands consist of HTTP or HTTPS requests directed to the channel controller port. They include parameter and value pairs separated from the main URL by standard query string syntax (“?” and “&”). The parameter values instruct Helix Server on the channel switching actions to take. The following example illustrates the command syntax:

http://helixserver:controller_port/stream.ext?param1=value1&param2=value2...

Channel Switching Control Port

The channel controller listens for HTTP or HTTPS requests on a user-configured port. The control port is initially set to 8008, but you can change this to any free port using Helix Administrator (Server Setup > Ports). RealNetworks does not recommend using the main HTTP port (typically port 80) as the control port, however.

**Tip:** The setting for the Enable Control Port Security field on the Helix Administrator ports page determines if only HTTP (security disabled) or HTTPS (security enabled) requests are accepted on the control port.

For More Information: Refer to the Helix Administrator online help topic Helix Administrator > Server Setup > Ports > Port Assignments for instructions about setting port values.

Control Port Security

To prevent unauthorized channel switching, ensure that only authorized Web portals and HTTP proxies can issue HTTP commands to Helix Server on its control port. Media users should not have direct access to the control port.

**Tip:** You can restrict port access by configuring your firewall appropriately or by defining an access rule. The Helix Server online help section Helix Administrator > Security > Access Control describes access rules.

HTTP Request Parameters

The following table summarizes the parameters included in an HTTP or HTTPS GET request directed toward the Helix Server controller port. The last
two columns indicate if the attribute is required in the initial request that sets up the channel session, as well as a channel switch request.

### HTTP Attributes for Channel Controller Requests

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Function</th>
<th>Example</th>
<th>Initial Request</th>
<th>Channel Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switching Allowed Marker</td>
<td>Indicates that the requested stream is a channel that can be switched.</td>
<td>hxs=1</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>Session ID</td>
<td>Identifies the user’s RTSP session.</td>
<td>guid=ID</td>
<td>required</td>
<td>required</td>
</tr>
<tr>
<td>Rate Control Flag</td>
<td>Enables rate control: 0 – rate control disabled (default) 1 – rate control enabled</td>
<td>mdp=1</td>
<td>optional</td>
<td>not used</td>
</tr>
<tr>
<td>Queuing Indicator</td>
<td>Specifies when the next channel starts to play: 0 – switch immediately (default) 1 – switch at end of playback</td>
<td>queue=1</td>
<td>not used</td>
<td>optional</td>
</tr>
<tr>
<td>SDP File on Error</td>
<td>Generates an SDP file on an error: 0 – no new SDP on error 1 – new SDP on error (default)</td>
<td>sdp=0</td>
<td>optional</td>
<td>optional</td>
</tr>
</tbody>
</table>

**Note:** The order of query string parameters in a controller URL does not matter.

**Tip:** You can change the query string parameter names, such as hxs, through the Helix Server configuration file. See the Helix Administrator online help topic [Configuration File > Content Management Configuration > Channel Switching Configuration](#).

### Switching Allowed Marker

The switching marker is the query string parameter hxs. The value for this parameter is 0 (stream is not switchable) or 1 (stream is switchable). The parameter and value hxs=1 must be included in each request URL to instruct Helix Server to allow the stream in a channel switching session.

**Note:** The presence of this query string parameter in the initial channel request causes Helix Server to generate an SDP file automatically. You therefore do not need to include the SDPgen mount point in the request URL.
Session ID

The session ID name is the value of the guid query string parameter. Helix Server identifies the stream-switching session using this parameter’s value. The Web portal is responsible for generating a unique value for each user, and including the parameter and value pair in each request to switch a user’s channel. The value should be 8 to 32 characters in length, and may contain any of the following characters:

- a-z
- A-Z
- 0-9
- - (hypen)

Rate Control Flag

The rate control flag is the query string parameter mdp. The default value of mdp=0 prevents server-side rate shifting. This means that, unless the media player uses a client-side rate control method, Helix Server delivers the channel streams at a single streaming rate. Including the parameter mdp=1 in the initial channel request enables Helix Server to use its server-side rate control feature.

**Note:** To enable rate control, add the mdp parameter to the request for the first channel. Once rate control is enabled or disabled, it stays in effect for the entire RTSP session. Adding this parameter to a request to switch to a subsequent channel has no effect.

**For More Information:** For server-side rate control to function, the feature must be enabled and configured for the client. For details about rate control, refer to Chapter 8.

Queuing Indicator

Adding the parameter queue=1 to the query string informs Helix Server to queue the new channel but not switch to the new channel until the current channel has ended. The default value of queue=0 stops the current channel and switches to the new channel immediately.

**Note:** Use the queue=1 parameter and value only when the current and requested channels have defined endpoints. This includes on-demand clips and finite SLTA playlists. For
broadcasts and looping SLTA playlists, the channel switch must occur immediately. Otherwise, it may not occur at all.

**Tip:** If a channel change is pending, and the user issues another channel change request, the new request overrides the queued request. This means that a user can have only one pending request at a time.

**SDP File on Error**

The default value of \texttt{sdp=1} causes Helix Server to return an SDP file for the requested channel if the channel switch request fails. If you add \texttt{sdp=0} to the query string, Helix Server returns only an HTTP code if the channel switch fails.

As an example of channel switch failure, suppose that the requested channel is encoded in a media format that differs from the current channel. The request causes an error that tears down the user’s RTSP session. However, if Helix Server returns a new SDP file to the Web portal, the portal can forward the SDP file to the user’s media player. This sets up a new channel session.

**HTTP Status Codes**

The following are the HTTP status codes that Helix Server may return for an HTTP request sent to its session controller port.

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>204 No Content</td>
<td>Initial SDP request or a channel switch request succeeded.</td>
</tr>
<tr>
<td>404 Not Found</td>
<td>Switch request failed because the hxs or guid parameter was not included.</td>
</tr>
<tr>
<td>273 No Session</td>
<td>Switch request failed because the RTSP channel session has already been closed, or the requested content does not exist.</td>
</tr>
<tr>
<td>274 Content Not Compatible</td>
<td>Switch request failed because the current channel stream and requested channel stream are incompatible. (See “Error Conditions” on page 259.)</td>
</tr>
</tbody>
</table>
Channel Switching Workflows

The following sections explain how the channel switching system carries out its essential tasks.

Initial Channel Playback

This section explains the sequence of events that occurs when a user initiates a new channel session. The following figure illustrates the process described in the subsequent steps.

Request First Channel

To start a new session by requesting a channel:

1. The user selects a channel using information supplied by the Web portal. The channel information may be delivered to the Web browser or to a rich media client.

2. The Web portal requests a channel SDP file from Helix Server using an HTTP or HTTPS GET request directed toward the server’s channel controller port. To the base URL, the Web portal adds the switching allowed and session ID parameters. For example, the URL to an on-demand channel that the Web portal requests from Helix Server might look like the following:

   http://helixserver.example.com:8008/station5.mp4?hxs=1&mdp=1&guid=a5230cgj321

3. Helix Server checks for an active channel that uses the submitted session ID value. If no such channel exists, it returns an SDP file for the requested channel to the Web portal. That SDP file encapsulates an RTSP channel...
URL that includes the same query string parameters and values that the Web portal submitted in the request URL.

4. The Web portal delivers the SDP file to the user, identifying the stream with the SDP MIME type, application/sdp.

   **Note:** The Web portal may pre-generate SDP files and link them to a Web page dynamically generated for the user. This means that the actions described in Step 2 occur before the user selects a channel. Once the user selects a channel as described in Step 1, the media player contacts Helix Server as described in Step 5.

5. The user’s media player initiates the RTSP session by contacting Helix Server using the channel URL.

6. As Helix Server begins to stream the channel, the channel controller registers the channel under the session ID supplied in the URL.

### Switch to a New Channel

As the initial channel plays, the stream can be changed to a different channel. The following figure illustrates the process described in the subsequent steps.

**Switch to New Channel**

To switch to a new channel:

1. The user selects a new channel, typically by clicking on an HTTP link supplied by the Web portal. Alternatively, the channel change request may
result from a playlist that the Web portal manages. In this case, this first step of user interaction is not required.

2. The Web portal directs an HTTP or HTTPS GET request to the Helix Server channel controller port. The URL does not request an SDP file. Rather, it requests a channel switch, and includes the hxs and guid query string parameters that were included in the initial channel request. To these parameters, the Web portal may add the queue or sdp parameter. For example, the URL request that the Web portal sends to Helix Server might look like the following:

   http://helixserver.example.com:8008/station9.mp4?hxs=1&queue=1&sdp=0&guid=a5230cgj321

3. When Helix Server receives the new channel URL, it does the following:
   - Checks that the session identified by the session ID value (guid value) is still active.
   - Verifies that the newly requested channel, whether an on-demand clip or a broadcast, exists.
   - Determines if the current channel and the new channel are compatible.

   If any of these verifications fail, Helix Server returns an HTTP error code and, optionally, an SDP file to the new channel.

4. Optionally, the Web portal delivers a response to the user. This might be a message indicating that the channel switch succeeded or failed. On a failure, the Web portal may simply deliver the SDP file for the new channel, if one was generated.

5. At the point indicated by the queue value (immediately, or after the current channel finishes), Helix Server switches the stream within the RTSP session to the new channel. The user’s media player begins to play the new channel stream as soon as all buffered data for the previous channel is consumed. The channel switch then occurs without any pause or rebuffering.

   **Tip:** The rate control feature allows you to specify the maximum amount of playback time (such as three seconds) the media player buffers for each stream. For details, refer to the section “Limiting Stream Buffering with Max Advance” on page 139.
End Session

If the current stream for a channel ends playback and Helix Server has not received a channel switch URL, Helix Server closes the channel.

➤ To end a channel session:

1. Helix Server issues the media player an RTSP BYE packet.

2. The media player may then issue an RTSP TEARDOWN command and close its TCP connection.

3. Helix Server records the final stream in its access log entry with a final_switch value of 1.

For More Information: For more about how streams within a channel session are recorded in the access log, refer to the Helix Server online help topic Helix Administrator > Logging and Monitoring > Basic Logging > Basic Access Log Fields.

Configuring Channel Switching

Do the following in Helix Administrator to enable channel switching.

• Click Server Setup > Ports and verify that the channel control port is set to the port value you wish to use.

  Note: If Enable Control Port Security is set to Yes, the controller port accepts only HTTPS connections.

• Click Content Management > Channel Switching, to enable channel switching and select the SDPgen mount point.

For More Information: For configuration details, refer to the Helix Administrator online help topic Helix Administrator > Content Management > Channel Switching.
APPENDIXES

The following appendixes contain useful reference information.
Chapter 3 explains the standard procedures for installing Helix Server or Helix Proxy on a dedicated machine. This appendix explains issues in the specific cases of installing Helix Server and Helix Proxy on the same machine. It also covers installing Helix Server on a machine that runs a Web server.

Installing the Server and Proxy on the Same Machine

RealNetworks recommends that each Helix Server and each Helix Proxy runs on a dedicated machine. However, it is possible to install both applications on the same machine as long streaming loads are light.

Machine Requirements for Dual Use

To host both applications, a computer must meet the following criteria:

• The machine cannot host a Web server in addition to both Helix Server and Helix Proxy.

• The machine needs a minimum of 512 Megabytes of available RAM. At least 1 Gigabyte of RAM is recommended.

• The machine must have adequate disk space for caching media clips. Helix Server also needs adequate disk space for the clips that it streams. In addition, you need approximately 50 Megabytes of free space for the software and log files.

For More Information: The Helix Proxy online help topic Helix Administrator > Proxy Setup > Cache explains how to change the cache size after installation.

• The machine needs at least two IP addresses, one for Helix Server and one for Helix Proxy. You can accomplish this by using a multi-homed machine that has two or more network interfaces. Most operating systems also
support virtual addressing in which you assign multiple IP addresses to one physical network card. Be sure to have your IP addressing configured on the Helix Gateway machine before you install the applications.

**For More Information:** Consult your hardware or operating system documentation for information about installing a network interface card or assigning virtual IP addresses, respectively.

### Installing the Server for Dual Use

The following procedure outlines the steps for installing Helix Server. After you do this, you can install Helix Proxy on the same machine.

➤ **To install Helix Server on the Gateway Machine:**

1. Run the installation program according to the instructions in the section “Installing Helix Server” on page 27. This installs the necessary software files.

2. After the installation, do **not** start Helix Server. If Helix Server is running because you installed it as a Windows Service, for example, shut it down according to the instructions in the section “Shutting Down on Windows” on page 44.

3. Using any text editor, open the Helix Server configuration file, which resides in the directory in which you installed Helix Server. The default configuration file name is *rmserver.cfg*.

4. In the configuration file, find the following lines, or add them to the bottom of the configuration file if they are not already present:

   ```xml
   <List Name="IPBindings">
     <Var Address_1="any"/>
   </List>
   ```

   This list defines the IP addresses that Helix Server uses. The default value of *any* for *Address_1* binds Helix Server to all IPv4 or IPv6 addresses used by the machine. This will cause conflicts with Helix Proxy. You therefore need to change this value to the IPv4 or IPv6 address that you want Helix Server to use. The following is an IPv4 example. Substitute one of your machine’s IP addresses for the *Address_1* value:

   ```xml
   <List Name="IPBindings">
     <Var Address_1="197.168.0.100"/>
   </List>
   ```
**Tip:** Be careful to change only the numeric values. If you accidentally delete one of the quotation marks that surrounds the IP value, for instance, Helix Server will not correctly bind to that IP address.

**Note:** Do not bind Helix Server to the localhost address (127.0.0.1 for IPv4 or ::1 for IPv6). Helix Proxy will use this address instead.

5. Save and close the configuration file. Be sure to save the file as plain text.

6. Do not restart Helix Server until after you have installed Helix Proxy. The section “Starting the Server with a Heartbeat Option” on page 274 provides start-up instructions.

### Installing the Proxy for Dual Use

After you install Helix Server on the Helix Gateway machine, install Helix Proxy as described in the following procedure.

➤ **To install Helix Proxy on the Gateway Machine:**

1. Run the binary installer according to the instructions in the section “Installing Helix Proxy” on page 29. This installs the necessary software files.

2. After the installation completes, do not start Helix Proxy. If Helix Proxy is running because you installed it as a Windows Service, for example, shut it down according to the instructions in the section “Shutting Down on Windows” on page 44.

3. Using any text editor, open the Helix Proxy configuration file, which resides in the directory in which you installed Helix Proxy. The default configuration file name is rmproxy.cfg.

4. In the configuration file, find the following lines, or add them to the bottom of the configuration file if they are not already present:

```xml
<List Name="IPBindings">
  <Var Address_1="any"/>
</List>
```

This list defines the IP addresses that Helix Proxy uses. The default value of any for Address_1 binds Helix Proxy to all IP addresses used by the machine. You therefore need to change this value to the IPv4 or IPv6
address that you want Helix Proxy to use, which must be different from the IP address that you assigned to Helix Server.

Additionally, create a new line using the variable Address_2 to assign the localhost address (127.0.0.1 for IPv4 or ::1 for IPv6) to Helix Proxy. The following is an IPv4 example. Substitute your machine’s IP address for the Address_1 value:

```
<List Name="IPBindings">
  <Var Address_1="197.168.0.101"/>
  <Var Address_2="127.0.0.1"/>
</List>
```

5. Save and close the configuration file. Be sure to save the file as plain text.

6. You can start Helix Proxy now or after you have started Helix Server. For start-up instructions, refer to Chapter 4.

**Starting the Server with a Heartbeat Option**

After you have installed and configured the IP addresses for both applications, you can start Helix Server with a heartbeat start-up parameter, --hbi, that prevents server operation from conflicting with the proyx. Add the parameter along with the IP address to which Helix Server is bound.

*For More Information:* See “Heartbeat Check” on page 49 for more about the heartbeat option.

**Startup Example**

The following example illustrates the heartbeat check and memory allocation options used when starting Helix Server from the command line:

```
Bin\rmserver rmserver.cfg --hbi 197.168.0.100 -m 512
```

**Startup Procedures**

Refer to the following sections for information about including startup options in the various procedures for starting the server or proxy:

- “Running a Windows Service” on page 39
- “Using the Start Menu or a Desktop Icon” on page 41
- “Starting Up from the Windows Command Line” on page 42
- “Starting the Server or Proxy on UNIX or Linux” on page 42
Installing Helix Server with a Web Server

RealNetworks recommends that you do not install Helix Server on the same physical machine that runs a Web server. This helps to balance loads so that streaming is not affected by heavy Web server use. If you decide to install Helix Server on a Web server machine, both products may try to claim ports 80 and 443 for HTTP and HTTPS requests. There are two ways to avoid this conflict.

Use Nonstandard Ports for Helix Server

During Helix Server installation, you can set ports other than 80 and 443 for HTTP requests. In this case, all HTTP URLs directed toward Helix Server must specify the alternate port number. This creates potential for errors in writing URLs, however, and may limit client access if firewalls restrict HTTP requests to ports 80 and 443.


Bind Helix Server to Specific IP Addresses

A better approach for avoiding port conflicts is to use two IP addresses for the same computer, one for Helix Server and the other for the Web server. This requires a multihomed machine that has two or more network interfaces.

In a multihomed configuration, you assign an IP address to each network interface and bind Helix Server to one of the IP addresses. In this way, Helix Server and your Web server both use ports 80 and 443 on a different network addresses.

Note, though, that Helix Server attempts to bind to all available IP addresses on startup. The following procedure explains how to work around this problem.

➤ To prevent an HTTP port conflicts on a multihomed machine:

1. When installing Helix Server, choose any unused ports other than 80 and 443 for HTTP and HTTPS, respectively. This prevents Helix Server startup from failing due to a port conflict with the Web server.

2. After installation, start Helix Server and bind it to one or more IP addresses not used by the Web server on the multihomed machine.

Tip: In Helix Administrator, choose Server Setup > IP Binding. Refer to the online help for details about IP binding options.
3. Change Helix Server’s HTTP port to 80 and its HTTPS port to 443.

   **Tip:** In Helix Administrator, choose **Server Setup > Ports**. Refer to the online help for detailed information.

4. Restart Helix Server. It now claims ports 80 and 443 only on the IP addresses on which it is bound.
INDEX

A  Administration system
   see Helix Administrator
archiving
   broadcasts, 186
ASXgen
   NAT firewall workaround, 14
   virtual IP addresses, 14
authentication databases, 37

B  back-channel multicasting
   overview, 207
   session announcement, 203
   setting up the broadcast, 209
   starting the broadcast, 209
   time to live, 207
   unicast failover, 208
   URL formats, 209
background process on UNIX, 43
bandwidth
   server-side rate control, 111
   unicasting limitations, 187
BCC, 141
   see also rate control
bookmarking, 61
broadcasting
   rate control, 113
   redundant encoders, 187
   standby message, 187
buffer size on Windows, 26

C  capabilities exchange, see rate control
channel negotiation, 15
channel switching
   browser requirements, 257
   channel controller
      HTTP requests, 261
   listen port, 261
   queue parameter, 263
   security, 261
   session ID, 263
   switching allowed marker, 262
codec support, 257
configuration, 268
error conditions
   causes of, 259
   HTTP status codes, 264
   SDP generation, 264
   file format support, 257
   media player requirements, 257
   overview, 255
   protocol support, 259
   Web portal requirements, 257
   workflows
      initial channel playback, 265
      select new channel, 266
      session end, 268
Chrome browser support, 60
command options
   debugging, 46
   heartbeat check, 49
   help, 50
   overview, 45
   redirecting output, 49
   server statistics, 49
   version number, 50
configuration file, 37
ccontrol channel for multicasts, 207
conventions in this manual, 3

crash avoidance
   disabling, 46
   maximum number, 46
   printing details, 46
DES encryption for SNMP, 71

Encoders
- firewall issues, 18
  see also redundant encoders
- end-user license, 28
  see also license file

File descriptors, 23
Firewalls
- media player issues, 15
- NAT address specification, 14
- proxies, 19
- receivers, 19
- server placement, 13
- Windows Media Encoder, 19

Flash broadcasting
- overview, 199
- server setup, 200
- splitting, 220
- URL formats, 200

Gateway installation
- proxy
  installation, 273
  start-up, 274
- server
  installation, 272
  start-up, 274
- system requirements, 271

Google Chrome support, 60

Heartbeat check, 49
Helix Administrator
- Admin port, 32
- applying changes, 58
- deleting configuration values, 58
- duplicating configuration values, 58
- password, 28, 30
- restarting the server, 59
- SecureAdmin port, 32
- starting, 57
- user name, 28, 30
- Helix Advanced Push broadcast, 189

Helix Live Transmitter, 244
Helix Mobile Producer
- Helix Advanced Push broadcasting, 189
- Helix Pull broadcasting, 191
- Helix Push broadcasting, 187

Helix Proxy
- multiple instances, 52
- service configuration, 52

Helix Pull broadcast
- broadcast steps, 191
- splitting, 219
- stream acquisition latency, 192
- URL formats, 192

Helix Push broadcast
- broadcast steps, 188
- splitting, 218
- URL formats, 189

Helix Rate Adaptation, see rate control
Helix Server
- placement in a network, 13
- service configuration, 52
- help system, 59
- .hpl file extension, 163

HTTP
- cloaking, 17
- port
  - conflict resolution, 275
  - nonstandard port use, 275
  - port 80 use, 18
- hyperthreading, 23

Installation, 27, 28
- machine ID, 31
- multi-homed machine, 275
- reinstalling the server, 36
  testing, 60
  Windows service, 29

iOS support, see media segmentation

Ir query string parameter, 123

License file
- delivery, 27
invalid files, 60
machine ID, 31
multiple files, 60
working with, 60
Link-Char header, 134
LogPath variable, 44

M
m3ugen
  mount point in URLs, 91
  overview, 91
machine ID, 31
master agent for SNMP, 72
master program on UNIX, 79
master.cfg file, 72
master.exe program on Windows, 78
MD5 for SNMP, 70
media formats
  multicasting, 203
  simulated live broadcasts, 230
  unicasting, 185
media players
  channel negotiation, 15
  firewall issues, 15
media segmentation
  audience groups, 87
  audio-only stream, 86
  bandwidth order in playlists, 92
  basic steps
    broadcast, 90
    on-demand, 88
  bookmarks, 93
  client redirection, 100
  codec requirements, 86
  content caching, 88
  directory names, 102
  encryption, 106
  algorithm, 107
  key file, 107
  name, 107
  updating, 107
  write location, 108
error message on missing content, 100
file names, 102
  multi-rate playlists, 105
  playlist file, 103
segments, 106
special characters, 106
timestamp, 104
gzip compression, 87
initial rate selection, 92
ir parameter, 93
mount points
  base directory, 96
  broadcast archives, 97
  m3ugen, 91
  overview, 94
  predefined, 94
  purging content, 98
  set-up, 101
  source paths, 96
  user-defined, 95
overview, 85
progressive download alternative, 109
purging content
  by size limit, 99
  on restart, 99
query parameters, 94
rate shifting, 86
Solaris 10 patch, 26
source paths, 96
memory
  -m option, 47
  maximum
    on UNIX, 43
    on Windows, 39
  memory-mapped I/O, 47
  server maximum, 48
  usage characteristics, 48
message URL, 27
MIB file for SNMP, 71
mount points
  /admin/, 57
  /m3ugen/, 91
MPEG
  MP3 broadcast archives, 186
MPEG-4
  unicasting, 192
multicasting
  control channel, 207
formats supported, 203
interactions with other features, 204
IP addresses
  IPv4, 206
  IPv6, 206
multi-homed machines, 207
network configuration, 206
on the Internet, 206
overview, 203
packet time to live (TTL), 207
see also back-channel multicasting
see also scalable multicasting
see also Windows Media multicasting
multi-homed machines
  installation issues, 275
  multicasting, 207

N
  NADU feedback, 126
  network interface card (NIC), see multi-homed machines

O
  online help, 59

P
  packet time to live, 207
password for Helix Administrator, 58
PID file, 44
PidPath variable, 44
playlist format
  chapter skipping
    chapter-skip attribute, 165
    example, 165
    overview, 157
  chapters, 170
clipBegin attribute, 167
comments, 163
dur attribute, 168
examples, 172
expiration time, 164
file extension
  .hpl, 163
  .hpl.3gp, 163
  .hpl.mp4, 163
  .hpl.rm, 163
playback error, 163
id attributes
  case-sensitivity, 171
  first characters, 171
  length, 171
  spaces in, 171
  uniqueness, 171
media sources, 167
metadata
  presentation, 164
secondary playlists, 171
seq attribute, 170
skippable attribute, 170
skip-point attribute, 169
SMIL background, 163
src-dur attribute, 168
timing values, 163
video source tags, 166
playlist management, 149
authentication
  playlist upload, 180
  user access, 180
browser requirements, 146
codec support
  iOS devices, 149
  RTSP players, 146
deleting a playlist, 175
enabling, 181
error conditions
  causes of, 176
  effect on timeline, 177
  HTTP status codes, 177
externally controlled session
  chapter skipping, 157
overview, 144
parameters
  guid, 161
  hpl, 160
  mdp, 162
  plref, 162
  seek, 161
time, 162
playlist selection, 154
rate control, 162
seeking
  implementing, 159
seeking to previous clip, 161
session ID, 161
skipping, 157
workflows, 154
feature compatibility, 150
file format support
  RTSP players, 146
internally controlled session
  overview, 144
  seeking, 154
workflows, 151
iOS device requirements, 148
iOS devices
  live broadcasts, 149
playlist naming, 150
segment creation, 149
segmentation configuration, 181
logging
  style, 176
variables, 176
mount point requirements, 179
noncontrolled session
  overview, 145
playlist selection, 151
workflows, 151
overview, 143
ports
  control port, 33
  file system control port, 33
protocol support, 148
RTSP media player requirements, 146
RTSP seeking, 154
security requirements, 179
updating a playlist, 175
uploading a playlist, 174
Web portal
  requirements, 146
  SDP modifications, 153, 156
ports
  control channel, 261
hinting, 18
HTTP, 275
port 80 conflict, 275
selecting, 28
tips for selecting values, 31
Process ID, 44
proxy firewall issues, 19

Q  QuickTime broadcasting, 192
R  Ramgen
    NAT firewall workaround, 14
    virtual IP addresses, 14
rate control
  3GPP adaptation, 112
  basic requirements, 111
  BCC
    overview, 141
    specification, 141
  broadcasting, 113
    encoder support, 114
    reduced start-up delay, 113
    splitting, 113
  buffer model, 127
    buffer size, 128
    minimum preroll setting, 129
    preroll, 128
  capabilities exchange, 117
  compatibility
    media formats, 114
    media players, 112
  configuring, 142
  congestion control, 140
  downshifting, 131
    client-initiated, 134
    downshift depth, 132
    downshift target-time depth, 133
    stepwise downshifting, 133
  encoding guidelines, 115
  Helix adaptation, 112
  initial oversend rate, 124
  initial rate (ir) query string parameter, 123
  initial rate selection, 123
  Link-Char header, 134
  max advance, 138
  maximum sending rate, 122
  NADU feedback, 126
  overview, 111
profiles
  default profile, 120
  inheritance, 117
  overview, 116
pre-defined profiles, 120
user-agent matching
  exact, 119
  partial, 119
RDF files, 117
receiver reports, 124
  receiver report frequency, 125
  report timeout, 126
  sender report frequency, 126
slowdown mode, 137
  exiting, 139
  max advance, 138
SLTA, 114
  stream.id evaluation, 123
SureStream comparison, 115
target time, 130
TCP, 141
TFRC
  overview, 141
  specification, 141
types of, 112
UAS files, 116
upshifting, 135
  client-initiated, 137
  stepwise upshifting, 136
  upshift depth, 136
  upshift target-time depth, 136
user-agent strings, 118
RDF files, 117
RealMedia broadcast archives, 186
RealPlayer rate control, 122
RealProducer
  Helix Advanced Push broadcasting, 189
  Helix Pull broadcasting, 191
  Helix Push broadcasting, 187
  sending streams to multiple receivers, 216
  splitting technology, 214
receiver firewall issues, 19
redundant encoders
  overview, 187
  SLTA, 248
  with splitting, 224
reinstallation, 36
  backing up configuration file, 37
  restarting the server, 59
RHEL file descriptors, 24
RSS command line report, 49
RTCP, 124
S
  64-bit support, 10
  SAP, 206
  scalable multicasting
    authentication, 210
    live channels, 210
    overview, 209
  SDP files
    overview, 209
    session announcement, 203
    setting up the broadcast, 210
    statistics, 210
    time to live, 207
    unicast failover, 210
  SDP files, 194
    additional reading, 206
    multicasting overview, 209
    QuickTime and RTP-based media
      directory scan, 195
      syntax requirements, 195
  SDPgen
    NAT firewall workaround, 14
    virtual IP addresses, 14
  server-side playlists, see playlist management
  Session Announcement Protocol, 203
  Session Description Protocol, see SDP files
  SHA for SNMP, 70
  simulated live broadcasts, see SLTA
  SLTA
    advanced mode
      bufferless transport, 241
      command line syntax, 247
      compared to basic mode, 230
      configuration file, 236
      configuration requirements, 231, 235
      example of running, 248
      pull splitting
        configuration, 241
        multiple configuration files, 241
        URL format, 253
        variables, 242
      push splitting
Index

defining, 238
multiple receivers, 238
stream direction, 240
URL format, 253
variables, 239
transmitter name, 237
tutorial, 233
XML file, 236

basic mode
command line syntax, 246
compared to advanced mode, 230
configuration requirements, 231
deamon example of running, 247
TCP transport, 251
tutorial, 232
URL format, 252
command line options, 249
environment variables, 245
setting on UNIX, 245
setting on Windows, 245
executable file, 246
features, 229
Flash and AAC, 251
media formats available, 230
monitoring, 252
multiple installations, 244
number of clips to play, 250
playlists
changing during a broadcast, 250
creating, 242
media formats and bit rates, 243
title, author, and copyright
overring, 251
setting, 243
quick start, 232
rate control, 114
redundant instances, 248
RTP packetization, 251
screen output while running, 252
shuffle play, 251
stopping, 252
wallclock synchronization, 251

SNMP
AgentX protocol and port, 74
authentication, 70
DES encryption, 71
management system, 79
master agent
address and port, 73
configuration, 72
security model, 74
UNIX startup, 79
VACM setup, 75
Windows Service installation, 29, 31
Windows startup, 78
MD5, 70
MIB file, 71
overview, 69
plug-in configuration, 71
privacy, 71
security, 74
SHA, 70
Solaris 10 update, 26
traps, 71
trees
configuration, 81
control, 82
monitor, 80
VACM setup, 75
version support, 70
Solaris
SLTA soft file descriptors, 25
system file descriptors, 25
update for SNMP, 26

splitting
encoder-to-server, 214
multicast distribution to players, 223
multiple splitting arrangements, 221
one-to-many, 216
one-to-one, 217
overview, 213
pull splitting
bandwidth efficiency, 215
configuring, 219
overview, 215
with push encoding, 216
push splitting
configuring, 217
overview, 214
transmitter setup, 214
with pull encoding, 215
rate control, 113
simultaneous unicast and multicast, 224, 226
split stream direction, 221
terminology definitions, 213
transmitter redundancy, 225
starting up
command options, 45
UNIX, 42
background process, 43
memory maximum, 43
Windows, 39
memory maximum, 39
stopping the server
UNIX, 44
Windows, 44
SureStream
rate control comparison, 115

TCP congestion control, 141
terminology, 3
test clips, 60
TFRC, 141
see also rate control
3GPP rate adaptation, see rate control
time to live (TTL), 207
transmitter
pull splitting, 219
push splitting, 214
redundancy, 225
typographical conventions, 3

UDP resend requests, 17
unicasting
bandwidth constraints, 187
MPEG-4
overview, 192
port selection, 194
SDP files, 194
setting up the broadcast, 194
splitting, 219
stopping the broadcast, 194
multicast failover
back-channel multicast, 208
scalable multicast, 210
overview, 185
QuickTime
overview, 192
SDP files, 194
setting up the broadcast, 194
splitting, 219
stopping the broadcast, 194
RealMedia
Helix Advanced Push mode, 189
Helix Pull mode, 191
Helix Push mode, 187
RTP-based media
overview, 192
SDP files, 194
setting up the broadcast, 194
stopping the broadcast, 194
URL formats, 194
standby message, 187
Windows Media
firewall issues, 19
pull broadcasts, 201
push broadcasts, 201
UNIX
PID, 44
starting the server, 42
stopping the server, 44
URLs
back-channel multicasts, 209

version number, 50
VIPs, 14
virtual IP addressing, 14

Web server
installing with Helix Server, 275
multi-homed machine installation, 275
Windows
account privileges, 26, 40
multiple proxy instances, 52
proxy service
automatic configuration, 31
manual configuration, 52
send buffer size, 26
server service
automatic configuration, 29
manual configuration, 52
starting the server, 39
stopping the server, 44

Windows Media
broadcasts, 200
multicasts
  overview, 211
  setting up, 211
see also uncasting

Windows Media multicasting
cannel definition, 211
overview, 211
time to live, 207