



DELIVERING MEDIA TO THE APPLE IPHONE

Helix™ Mobile Server Version 13.0

Revision Date: 4 September 2009

RealNetworks, Inc.
PO Box 91123
Seattle, WA 98111-9223
U.S.A.

<http://www.real.com>
<http://www.realn networks.com>

©2009 RealNetworks, Inc. All rights reserved.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of RealNetworks, Inc.

Printed in the United States of America.

Helix, Helix DNA, the Helix logo, the Real "bubble" (logo), RBN, RealArcade, RealAudio, Real Broadcast Network, Real.com, RealJukebox, RealMedia, RealNetworks, RealOne, RealPix, RealPlayer, RealPresenter, RealProducer, RealProxy, RealSystem, RealText, RealVideo, SureStream, and TurboPlay are trademarks or registered trademarks of RealNetworks, Inc.

Other product and corporate names may be trademarks or registered trademarks of their respective companies.

CONTENTS

CONTENT DELIVERY FOR THE APPLE IPHONE.....	1
IPHONE MEDIA DELIVERY METHODS.....	1
Progressive Download	1
Segmented Download	1
Rate Shifting with Segmented Downloads.....	2
HTTP-Based Limitations of Segmented Downloads.....	2
USING PROGRESSIVE DOWNLOAD.....	2
Video Codecs	3
Audio Codecs	3
SETTING UP SEGMENTED DOWNLOAD.....	3
Encoding and Segmenting in a Single Step	4
Available Encoding Products	5
Encoding and Segmenting in Multiple Steps.....	5
Broadcasting.....	6
CONFIGURING HELIX MOBILE SERVER.....	6
Creating a Mount Point.....	6
Enabling HTTP.....	7
Configuring MIME Types	8

CONTENT DELIVERY FOR THE APPLE IPHONE

This document describes how to use Helix Mobile Server Version 13.0 to deliver media to Apple iPhone clients running OS 3.0 or later. It supplements the information in the document *HTTP Live Streaming Overview*, which you can find on the Apple Web site by clicking the following link:

<http://developer.apple.com/iphone/library/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/iPhoneStreamingMediaOverview/iPhoneStreamingMediaOverview.html>

Note: Delivering media to the iPhone requires Helix Mobile Server Version 13.0 or later. Earlier versions of Helix Mobile Server do not implement the features required by the iPhone client.

iPhone Media Delivery Methods

The Apple iPhone provides a client-driven system for downloading audio and video files. There are two methods for delivering content: *progressive download* and *segmented download*. The following sections summarize these download methods. Later sections explain how to implement each method.

Progressive Download

The iPhone client can use HTTP to download and play an entire QuickTime or MPEG file formatted for progressive download. Each file begins to play shortly after the download starts. Viewers can seek back and forth through the entire presentation. If the viewer seeks forward to a point in the timeline that has not yet downloaded, the media player pauses playback until the data arrives.

Segmented Download

A segmented media presentation approximates on-demand streaming and live broadcasting by encoding each presentation as a series of MPEG-2 transport

stream (.ts) files. An index file lists the order of, and the URLs to, each segment file. After receiving the index file, the iPhone media client downloads segment files using HTTP, buffering and playing each segment of the media presentation in order. The viewer can seek backward and forward through media segments that have arrived.

Rate Shifting with Segmented Downloads

Using the segmented download option, you can implement a rate-shifting feature by creating and segmenting separate media streams that target different bandwidths. In this case, the index file provides different sets of URLs for transport files encoded at different bandwidths. The iPhone client chooses which files to request based on its ongoing assessment of the available bandwidth.

For More Information: Refer to the Apple documentation for more information about this feature, which is called *stream alternates*.

HTTP-Based Limitations of Segmented Downloads

Content delivery from Helix Mobile Server to the Apple iPhone is entirely client-driven using the HTTP protocol. Because of this, the following features, which require the use of RTSP, are not available:

- server-side rate control
- RTSP-based, live broadcasts
- simulated-live broadcasts using SLTA
- server-side playlist management
- channel switching
- proxied content using Helix Mobile Proxy

Note: Helix Mobile Server does **not** provide support for secure HTTP downloads (https://).

Using Progressive Download

The progressive download option is the easier method for delivering content. Using a product such as Helix Mobile Producer, encode content in one of the following formats:

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

Content **must** be encoded using the progressive download option. This allows the iPhone client to begin playback shortly after the download begins. Host the content on an HTTP-enabled mount point.

Note: The iPhone client does **not** support RTSP-based streaming of these media formats.

For More Information: See “Configuring Helix Mobile Server” on page 6.

Video Codecs

Supported video codecs for progressive download are the following:

- MPEG-4
- H.263
- H.264

Audio Codecs

Supported audio codecs for progressive download are the following:

- AAC
- AAC+
- AMR-NB
- AMR-WB
- MP3

Setting Up Segmented Download

The following are the general steps required to deliver segmented content to the iPhone from Helix Mobile Server Version 13.0.

► To deliver segmented content to the iPhone client:

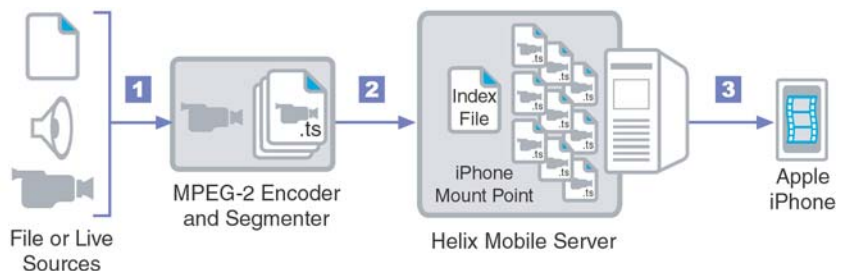
1. Encode media using the following settings:
 - Video – H.264 baseline profile, level 3 or below.
 - Audio – AAC or AAC+ at 64 Kbps. The sampling rate should be 44.1 kHz or 48 kHz.
2. Package the encoded media in the MPEG-2 transport stream (.ts) format.
3. Segment the transport stream into a series of files, each file providing 10 to 30 seconds of media playback.
4. Create a plain-text index file (.m3u8) that lists segment files in order and provides the URLs used to request the files.
5. Place the segmented media files and the index file in a content directory accessible under an HTTP-enabled mount point on Helix Mobile Server.

Once the content is ready, the iPhone client requests the index file using HTTP. Using the URLs listed in the index file, the iPhone client requests the segmented files using HTTP.

Encoding and Segmenting in a Single Step

Certain encoding products can perform several of the steps required to make segmented media available to iPhone clients.

Encoding and Segmenting Using a Single Product



1. In this scenario, the encoding product does the following:
 - encodes the input media using the required codecs
 - creates segmented transport stream files (.ts)
 - writes an index file (.m3u8)

2. All files are transferred to the appropriate content directory on Helix Mobile Server.
3. Content is made available to iPhone clients using HTTP links in Web pages.

Available Encoding Products

RealNetworks encoding products do **not** provide all of the required encoding features. Apple, Inc. has tested the following commercial encoders, which provide the requisite functionality described in Step 1:

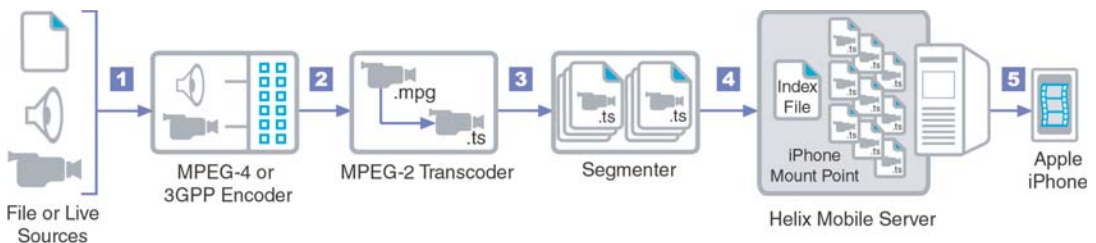
- **Inlet Technologies Spinnaker 7000**
- **Envivio 4Caster C4**

For More Information: Check the Apple documentation for the latest information about supported media encoding and segmenting products.

Encoding and Segmenting in Multiple Steps

You can also use multiple products to perform the required steps for producing segmented media.

Encoding and Segmenting Using Multiple Products



1. An MPEG or 3GPP encoder such as Helix Mobile Producer encodes the input media using the required codecs.
2. A third-party product transcodes the MPEG or 3GPP output into the MPEG-2 transport stream format (.ts).
3. A product such as the Apple segmenter receives the transport stream and segments it into a series of .ts files. It also generates the required index file (.m3u8).

For More Information: The Apple segmenter is available to members of Apple Developer Connection at the following Web address: <https://connect.apple.com/cgi-bin/WebObjects/MemberSite.woa/wa/getSoftware?bundleID=2033>

4. All files are transferred to the appropriate content directory on Helix Mobile Server.
5. Content is made available to Apple iPhone clients using HTTP links in Web pages.

Broadcasting

Broadcasting a live event to Apple iPhone clients requires that your encoding process deliver segmented files to Helix Mobile Server in near-real-time. The iPhone client does not receive a continuous broadcast stream. Rather, it requests the broadcast media as a series of discrete .ts files, just as with an on-demand presentation. Each client therefore starts at the first file listed in the current index file regardless of when the broadcast URL was clicked.

Note: The segmenter can update the index file at regular intervals. This ensures that clients joining the broadcast start at a segment that was recently encoded. Refer to the segmenter documentation for details.

Configuring Helix Mobile Server

To deliver media to the Apple iPhone client, you set up one or more HTTP-enabled mount points to host the content. Then, you configure Helix Mobile Server to support the required media MIME types.

Creating a Mount Point

The following procedure explains how to create a mount point for Apple iPhone content. Do this if you are using either the progressive download method or the segmented download method.

Note: Do **not** use an existing mount point that holds content streamed using RTSP or MMS. Enabling HTTP for a streaming mount point allows Web browsers to download and cache the content.

► To create a mount point for iPhone content:

1. In Helix Administrator, click **Server Setup>Mount Points**.
2. Click the “+” icon to add a new mount point.
3. Enter a name, such as iPhone, to replace the generic mount point name that appears in the **Edit Description** box.
4. In the **Mount Point** box, specify the mount point as it will appear in the request URL. Do not use spaces in the name, and enclose the name with forward slashes, as in /iPhone/.
5. In the **Base Path** box, enter the full path to the directory that stores the content. Each mount point typically has a unique base path that points to a directory on the local Helix Mobile Server or a network file system.

Note: The process that delivers the transport stream files and index files to Helix Mobile Server will need write permission for this directory.

6. If content resides in an external data store such as a Network File System (NFS) or a storage area network (SAN), you might see a performance increase by selecting **Network** from the **Base Path Location** box.
7. Leave **Allow File Creation** and **Allow File Deletion** set to No.
8. Click **Apply**.

For More Information: Refer to *Helix Mobile Server Administration Guide* for additional information about managing mount points.

Enabling HTTP

After you create the new mount point, enable it for HTTP delivery.

► To allow HTTP delivery for content under an iPhone mount point:

1. Click **Server Setup> HTTP Delivery**.
2. Click the “+” icon to add a new path.
3. Change the name in the **Edit Path** box to that of the mount point you created. Include only the initial forward slash, as in /iPhone.
4. Click **Apply**.

Configuring MIME Types

If you are using the segmented download method, configure Helix Mobile Server to recognize the MIME types for the transport stream and index files. You do **not** need to add MIME types if you are using only the progressive download method.

► To add the MIME types for the transport stream and index file:

1. Click **Server Setup>MIME Types** in Helix Administrator.
2. Click the “+” icon to add a new MIME type.
3. Enter the following in the **Edit MIME Type** box:
application/x-mpegURL
4. Enter the following in the **Extensions** box (do not precede the file extension with a period):
m3u8
5. Click the “+” icon again to add a second MIME type.
6. Enter the following in the **Edit MIME Type** box:
video/MP2T
7. Enter the following in the **Extensions** box:
ts
8. Click **Apply**.