

HTTP Live Streamingの 概要

目次

はじめに 5

概要 6

特別なサーバソフトを使用しないオーディオとビデオの送信 7

オプションの暗号化を使用したオンデマンドのライブストリームまたはビデオの送信 7

お読みになる前に 7

関連項目 8

HTTP Streamingのアーキテクチャ 9

概要 9

サーバコンポーネント 11

メディアエンコーダ 11

ストリームセグメンタ 11

ファイルセグメンタ 12

メディアセグメントファイル 12

インデックスファイル（プレイリスト） 12

ディストリビューションコンポーネント 13

クライアントコンポーネント 14

HTTP Live Streamingの使用 15

ツールのダウンロード 15

メディアストリームセグメンタ 15

メディアファイルセグメンタ 15

メディアストリームバリデータ 16

バリエーションプレイリストクリエータ 16

メタデータタグジェネレータ 16

セッションタイプ 16

コンテンツ保護 17

プロトコルのキャッシュと送信 18

代替ストリーム 18

携帯電話ネットワークのビデオ送信 21

アプリケーションの要件 21

フェイルオーバー保護 22

タイムドメタデータの追加 23

クローズドキャプションの追加 24

メディアのiOSベースのデバイスへの配信準備	24
サンプルストリーム	27

HTTP Live Streamingの配備 28

HTMLページの作成	28
ウェブサーバの設定	29
ストリームの検証	30
HTTPSを使ったキーファイルの安全な送信	31

よく訊ねられる質問 33

書類の改訂履歴 40

図、表、リスト

HTTP Streamingのアーキテクチャ 9

図 1-1 基本的な設定 10

HTTP Live Streamingの使用 15

図 2-1 代替ストリーム 19

表 2-1 iPhone、iPod touch、iPad、Apple TVのエンコーダ設定、アスペクト比16:9 25

表 2-2 iPhone、iPod touch、iPad、Apple TVのエンコーダ設定、アスペクト比4:3 26

表 2-3 エンコーダ追加設定（iPadとApple TVのみ）アスペクト比16:9 26

表 2-4 エンコーダ追加設定（iPadとApple TVのみ）アスペクト比4:3 26

HTTP Live Streamingの配備 28

リスト 3-1 ウェブページを使ったHTTP Live Streamingの配布 28

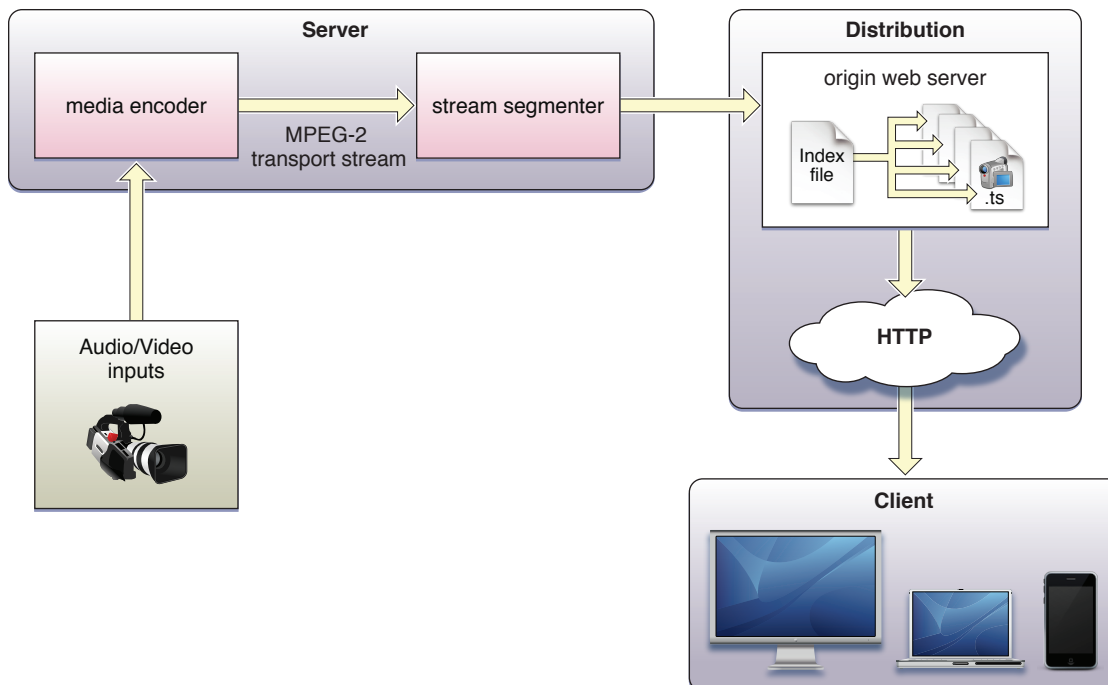
はじめに

本書の対象者は、以下の内容に関心がある方々です。

- iPhone、iPod touch、iPad、Apple TVへのオーディオまたはビデオのストリーミング
- 特別なサーバソフトウェアを使用しないストリーミングライブイベント
- 暗号化と認証を使用したオンデマンドのビデオ送信

対象者はHTTP Live Streamingについて学習する必要があります。

HTTP Live Streamingは、通常のウェブサーバからHTTPを使ってオーディオとビデオを送信し、iPhone、iPad、iPod touch、Apple TVなどのiOSベースのデバイス、およびデスクトップコンピュータ（Mac OS X）で再生するための機能です。HTTP Live Streamingは、ライブ放送と既録のコンテンツ（ビデオオンデマンド）のいずれにも対応しています。HTTP Live Streamingは、ビットレートの異なる複数の代替ストリームをサポートし、クライアントソフトウェアはネットワーク帯域幅の変更に応じて、インテリジェントにストリームを切り替えられます。HTTP Live Streamingはまた、HTTPSを使用したメディアの暗号化とユーザ認証にも対応しているため、発行者は作品を保護することができます。



iOS 3.0以降を実行するすべてのデバイスは、HTTP Live Streaming用のクライアントソフトウェアをインストールしています。Safariブラウザは、iPadとデスクトップコンピュータのウェブページ内でHTTPストリームを再生します。iPhoneやiPod touchなどの小型画面のiOSデバイスの場合、SafariではHTTPストリーム用のフルスクリーンメディアプレイヤーが起動します。Apple TV 2以降には、HTTP Live Streamingクライアントがインストールされています。

Important 携帯電話ネットワークで大量のオーディオまたはビデオデータを送信する、iPhoneとiPadのアプリケーションは、HTTP Live Streamingの使用が必須です。[「アプリケーションの要件」](#)（21 ページ）を参照してください。

Safariは、<VIDEO>タグのソースとしてHTTP Liveストリームをネイティブに再生します。Mac OS Xデベロッパは、UIKitとAVFoundationフレームワークを使用して、HTTP Live Streamを再生するデスクトップアプリケーションを作成し、iOSデベロッパは、MediaPlayerとAVFoundationフレームワークを使用してiOSアプリケーションを作成します。

Important HTTP Live Streamingを組み込む場合は、必ず<video>タグを使用してください。<object>タグや<embed>タグを使用しないでください（フォールバックコンテンツを指定する場合を除く）。

この種類のストリーミングはHTTPを使用するため、自動的にほぼすべてのエッジサーバ、メディアディストリビュータ、キャッシュシステム、ルータ、ファイアウォールでサポートされています。

注意 既存の多くのストリーミングサービスは、コンテンツのエンドユーザへの配信に特別なサーバを必要とします。これらのサーバをセットアップおよび維持するためには特殊な技能が必要とされ、大規模な配置の場合、コストを要します。HTTP Live Streamingは、メディアの配信に標準のHTTPを使用するため、このような問題が発生しません。さらに、HTTP Live Streamingは、大規模な操作の場合でも、メディアディストリビューションネットワークとシームレスに連動するように設計されています。

HTTP Live Streamingの仕様は、IETFインターネットドラフトです。

概要

HTTP Live Streamingは、デスクトップのクライアントソフトウェアまたはiOSベースのデバイスに、ウェブサーバからHTTPを使用してオーディオとビデオを送信する方法です。

特別なサーバソフトを使用しないオーディオとビデオの送信

HTTP Live Streamingのオーディオとビデオは、通常のウェブサーバから配信できます。クライアントソフトウェアは、SafariブラウザまたはiOSまたはMac OS X用に記述したアプリケーションになります。

HTTP Live Streamingは、メディアセグメントファイルと呼ばれる一連の低容量ファイルとして、通常は約10秒間でオーディオとビデオを送信します。インデックスファイル、またはプレイリストで、メディアセグメントファイルのURLがクライアントに伝えられます。プレイリストは、メディアセグメントファイルが継続的に生成されるライブ放送に対応して、定期的に更新することができます。プレイリストは、リンクをウェブページに埋め込むことも、デベロッパが記述したアプリケーションに送信することもできます。

関連する章 「[HTTP Streamingのアーキテクチャ](#)」 (9 ページ)

オプションの暗号化を使用したオンデマンドのライブストリームまたはビデオの送信

既録のメディアを配信するビデオオンデマンドについては、H.264ビデオ圧縮方式のMPEG-4ビデオまたはQuickTimeムービー、あるいはAACまたはMP3圧縮方式のオーディオファイルからメディアセグメントファイルとプレイリストを作成する無料ツールが、Appleから配布されています。プレイリストとメディアセグメントファイルは、ビデオオンデマンドやストリーミングラジオなどに使用できます。

ライブストリームについては、H.264ビデオ、AACオーディオ、MP3オーディオを搬送するライブMPEG-2トランスポートストリームからメディアセグメントファイルとプレイリストを作成する、無償ツールがAppleから配布されています。MPEG-4ビデオとAACオーディオを搬送するMPEG-2トランスポートを実時間で作成する、ハードウェアおよびソフトウェアエンコーダもいくつかあります。

上記のツールのいずれかに、メディアの暗号化と復号キーの生成を指示します。すべてのストリームに共通のキーを使用するか、ストリームごとに異なるキーを使用するか、ストリームの間に一定間隔で変更される、ランダムに生成された一連のキーを使用するか、いずれかの方法を選べます。キーはさらに、初期化ベクトルの要件により保護され、定期的な変更を初期化ベクトルに設定できます。

関連する章 「[HTTP Live Streamingの使用](#)」 (15 ページ)

お読みになる前に

読者が共通のオーディオおよびビデオのファイル形式の一般的な知識を有していること、またウェブサーバとブラウザの動作に慣れていることを前提としています。

関連項目

- *iOS Human Interface Guidelines* —iOSベースのデバイス用のウェブコンテンツのデザイン方法。
- [HTTP Live Streaming プロトコル](#) —IETFインターネットドラフトのHTTP Live Streamingの仕様。

HTTP Streamingのアーキテクチャ

HTTP Live Streamingを使用すると、暗号化と認証がサポートされたライブまたは既録のオーディオとビデオを、通常のウェブサーバからiOS 3.0以降が稼働するデバイス（iPadとApple TVを含む）、またはSafari 4.0以降がインストールされたコンピュータに送信することができます。

概要

概念上は、HTTP Live Streamingは、3つのパート、すなわちサーバコンポーネント、ディストリビューションコンポーネント、クライアントソフトウェアから構成されています。

サーバコンポーネントは、入力されたメディアストリームの取り込みとデジタルエンコーディング、同メディアの配信に適した形式でのカプセル化、およびカプセル化したメディアのディストリビューションの準備を担当します。

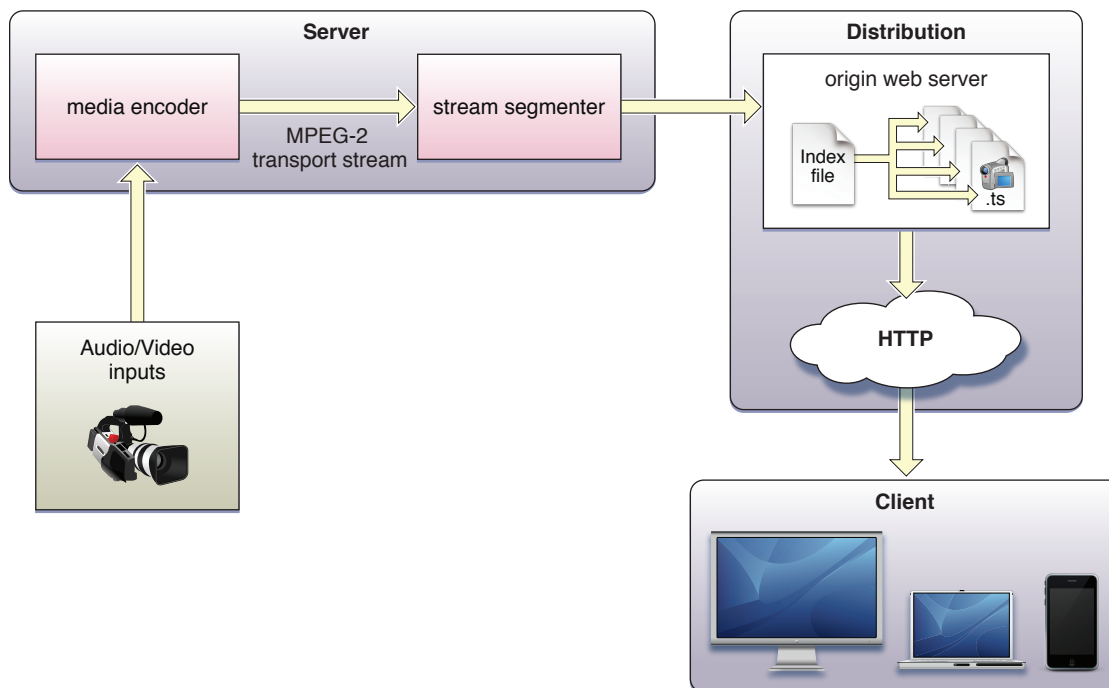
ディストリビューションコンポーネントは、標準のウェブサーバです。ウェブサーバは、クライアントの要求の受付と、準備されたメディアと関連リソースのクライアントへの配信を担当します。大規模なディストリビューションの場合、エッジネットワークやその他のコンテンツ配信ネットワークも使用できます。

クライアントソフトウェアは、要求に適したメディアの決定、それらのメディアのためのリソースのダウンロード、メディアを連続的なストリームでユーザに提供するための再アセンブルを担当します。クライアントソフトウェアは、iOS 3.0以降、およびSafari 4.0以降がインストールされたコンピュータに付属しています。

通常の設定では、ハードウェアエンコーダはオーディオビデオ入力を取り込み、H.264ビデオとAACオーディオでエンコードし、MPEG-2トランスポートストリームで出力します。これはソフトウェアストリームセグメントにより、一連の短いメディアファイルに分割されます。これらのファイルは、ウェブブラウザに配置されます。またセグメントは、メディアファイルのリストを含むインデックスファイルを作成し、維持します。インデックスファイルのURLは、ウェブブラウザで発行されます。クライアントソフトウェアはインデックスを読み取り、リスト内のメディアファイルを順番にリクエストし、セグメント間にポーズやギャップを挟まずにメディアファイルを表示します。

簡単なHTTPストリーミングの設定の例を、「基本的な設定」に示します。

図 1-1 基本的な設定



入力はいずれもライブで、または既録のソースを使用できます。通常はMPEG-4（H.264ビデオとAACオーディオ）でエンコードされ、既製のハードウェアによりMPEG-2トランスポートストリームにパッケージングされます。MPEG-2トランスポートストリームは、セグメントに分割され、1つまたは複数の.tsメディアファイルとして保存されます。このタスクは通常は、Appleストリームセグメンタなどのソフトウェアツールを使用して実行されます。

オーディオ専用ストリームは、ADTSヘッダを使用するAACまたはMP3でフォーマットされた、一連のMPEGエレメンタリオーディオファイルです。

セグメンタはインデックスファイルも作成します。インデックスファイルは、メディアファイルのリストを含みます。またインデックスファイルはメタデータを含みます。インデックスファイルは、.M3U8プレイリストです。インデックスファイルのURLにクライアントがアクセスし、クライアントはインデックス内のファイルを順番にリクエストします。

サーバコンポーネント

サーバはメディアエンコーダ（既製のハードウェアの使用可）と、エンコードされたメディアをセグメントに分割し、ファイルとして保存する方法（Appleから提供されるメディアストリームセグメンタなどのソフトウェアの使用可）を必要とします。

メディアエンコーダ

メディアエンコーダは、オーディオビデオデバイスからリアルタイム信号を取り込み、メディアをエンコードし、トランスポート用にカプセル化します。エンコーディングは、H.264ビデオやHE-AACオーディオなど、クライアントデバイスでサポートされる形式に設定する必要があります。現在サポートされている配信形式は、オーディオビデオのMPEG-2トランスポートストリームと、オーディオ専用のMPEGエレメンタリストリームです。

エンコーダは、ローカルネットワークを介して、エンコードされたメディアをMPEG-2トランスポートストリームでストリームセグメンタに配信します。

注意 MPEG-2トランスポートストリームを、MPEG-2ビデオ圧縮方式と混乱しないでください。トランスポートストリームは、複数の異なる圧縮形式で使用するパッケージング形式です。オーディオビデオコンテンツについては、現時点で、H.264ビデオとAACオーディオのMPEG-2トランスポートストリームのみサポートされています。オーディオ専用コンテンツは、ADTSヘッダを使用するAAC形式またはMP3形式の、MPEG-2トランスポートまたはMPEGエレメンタリオディオストリームがサポートされています。

Important ビデオエンコーダは、ストリームのエンコーディングの間に、ビデオ寸法やコーデックタイプなどのストリーム設定を変更できません。

ストリームセグメンタ

ストリームセグメンタは、ローカルネットワークからトランスポートストリームを読み取り、等時間の一連の低容量メディアファイルに分割するプロセス、通常はソフトウェアです。各セグメントが別のファイル内にある場合でも、ビデオファイルはシームレスに再構築できる連続的なストリームから作成されます。

セグメンタは、個々のメディアファイルの参照を含むインデックスファイルも作成します。セグメンタが新しいメディアファイルを終了するたびに、インデックスファイルは更新されます。インデックスは、メディアファイルの可用性と位置の追跡に使用されます。セグメンタは各メディアセグメントを暗号化し、プロセスの一環としてキーファイルを作成する場合があります。

メディアセグメントは、.tsファイル（MPEG-2トランスポートストリームファイル）として保存されます。インデックスファイルは、.M3U8プレイリストとして保存されます。

ファイルセグメンタ

メディアファイルをサポートされるコーデックを使ってエンコードしている場合は、ファイルセグメンタを使用してMPEGストリームにカプセル化し、同じ長さのセグメントに分割することができます。ファイルセグメンタでは、HTTP Live Streamingを使ったオンデマンドのビデオ送信時に、既存のオーディオとビデオファイルのライブラリを使用できます。ファイルセグメンタは、ストリームセグメンタと同じタスクを実行しますが、ストリームではなく入力としてファイルを取り込みます。

メディアセグメントファイル

メディアセグメントファイルは、通常はエンコーダからの入力に基づき、ストリームセグメンタにより生成されます。H.264ビデオとAACオーディオを搬送するMPEG-2トランスポートストリームのセグメントを含む、一連の.tsファイルから構成されます。オーディオ専用の放送の場合、セグメンタはADTSヘッダを使用するAACオーディオまたはMP3オーディオを含む、MPEGエレメンタリオーディオストリームを生成します。

インデックスファイル（プレイリスト）

インデックスファイルは通常は、ストリームセグメンタまたはファイルセグメンタにより生成され、MP3プレイリストに使用される.m3u形式の拡張形式である、.M3U8プレイリストとして保存されます。

注意 インデックスファイルの形式は、.m3uプレイリスト形式の拡張形式であるため、またシステムは.mp3オーディオメディアファイルもサポートするため、クライアントソフトウェアはインターネットラジオのストリーミングに使用される通常のMP3プレイリストと互換性を持つ場合があります。

.M3U8プレイリストの形式の、インデックスファイルの非常に簡単な例を以下に示します。これはストリーム全体が、暗号化されていない10秒の3つのメディアファイルに分割されている場合に、セグメンタにより生成されるインデックスファイルです。

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:1
```

```
#EXTINF:10,  
http://media.example.com/segment0.ts  
#EXTINF:10,  
http://media.example.com/segment1.ts  
#EXTINF:10,  
http://media.example.com/segment2.ts  
#EXT-X-ENDLIST
```

注意 Appleから提供されるファイルセグメントを使用すると、MPEG-4ビデオまたはAACやMP3オーディオファイルをソースに使用した、さまざまなサンプルプレイリストを生成できます。詳細については、「[メディアファイルセグメント](#)」（15 ページ）を参照してください。

インデックスファイルは、暗号化キーファイル用のURLと、各帯域幅の代替インデックスファイルを含む場合もあります。インデックスファイルの形式の詳細については、[HTTP Live Streaming仕様](#)のIETFインターネットドラフトを参照してください。

インデックスファイルは、通常は、メディアセグメントファイルを作成する同じセグメントにより作成されます。または、いずれも発行済みの仕様に適合している場合は、.M3U8ファイルとメディアセグメントファイルを個別に作成することも可能です。音声専用の放送の場合など、テキストエディタを使用して、既存の一連の.MP3ファイルをリストアップした.M3U8ファイルを作成することができます。

ディストリビューションコンポーネント

ディストリビューションシステムは、メディアファイルとインデックスファイルをHTTPでクライアントに配信する、ウェブサーバかウェブキャッシュシステムになります。コンテンツの配信にカスタムサーバモジュールは不要です。通常はウェブサーバでの非常にわずかな設定で十分です。

推奨される設定は、通常は.M3U8ファイルと.tsファイルのMIMEタイプの関連付けの指定に限定されます。

ファイル拡張子	MIMEタイプ
.M3U8	application/x-mpegURLまたはvnd.apple.mpegURL
.ts	video/MP2T

要求されるキャッシュ動作をダウンストリームのウェブキャッシュに適用する場合、.M3U8ファイルのtime-to-live (TTL) 値のチューニングが必要になる場合があります。これらのファイルはライブ放送の間にオーバーライトされることが多く、それぞれのリクエストに対して最新のバージョンをダウンロードする必要があるためです。

クライアントコンポーネント

クライアントソフトウェアは起動後、ストリームを識別するURLに基づき、インデックスファイルを取得します。インデックスファイルは、この後、使用可能なメディアファイル、復号キー、利用可能なすべての代替ストリームの場所を指定します。選択されたストリームに対して、クライアントは利用可能なメディアファイルを順番にダウンロードします。各ファイルには、ストリームの連続したセグメントを含みます。十分な量のデータがダウンロードされると、クライアントは再アセンブルされたストリームのユーザへの表示を開始します。

クライアントは、復号キーの取得、認証または認証を許可するユーザインターフェイスの表示、必要に応じてメディアファイルの復号を行います。

このプロセスは、クライアントがインデックスファイルで#EXT-X-ENDLISTタグを検出するまで続きます。#EXT-X-ENDLISTタグが存在しない場合、インデックスファイルは配信中の放送の一部です。配信中の放送の間に、クライアントは新しいバージョンのインデックスファイルを定期的にロードします。クライアントは更新されたインデックスで新しいメディアファイルと暗号化キーを探し、これらのURLをキューに追加します。

HTTP Live Streamingの使用

ツールのダウンロード

HTTP Live Streamingサービスの設定に役立つ、複数のツールが利用できます。ツールには、メディアストリームセグメンタ、メディアファイルセグメンタ、ストリームバリデータ、id3タグジェネレータ、バリエーションプレイリストジェネレータなどがあります。

ツールは頻繁に更新されるため、最新版のHTTP Live StreamingツールをApple Developerウェブサイトからダウンロードする必要があります。iPhone Developerプログラムの会員であれば、ツールにアクセスできます。ツールにナビゲートする場合、connect.apple.comにログインし、「Downloads」の見出しで「iPhone」をクリックする方法があります。

メディアストリームセグメンタ

mediastreamsegmenter コマンドラインツールは、MPEG-2 トランスポートストリームを入力として取り込み、この入力から、HTTP Live Streaming の用途に適した同じ長さの一連のファイルを作成します。このツールはまた、（プレイリストと呼ばれる）インデックスファイルの生成、メディアの暗号化、暗号化キーの生成、オーバーヘッドの低減によるファイルの最適化、複数の代替ストリームの自動生成に必要なファイルの作成も行います。詳細については、ターミナルウィンドウから `man mediastreamsegmenter` と入力してください。

使用例：`mediastreamsegmenter -s 3 -D -f /Library/WebServer/Documents/stream 239.4.1.5:20103`

この使用例では、ネットワークのアドレス 239.4.1.5:20103 からライブストリームをキャプチャし、ここからメディアセグメントファイルとインデックスファイルを作成しています。インデックスファイルは、最新の3つのメディアセグメントファイルのリストを含みます（`-s 3`）。メディアセグメントファイルは、使用後に削除されます（`-D`）。インデックスファイルとメディアセグメントファイルは、ディレクトリ `/Library/WebServer/Documents/stream` に保存されます。

メディアファイルセグメンタ

mediafilesegmenter コマンドラインツールは、エンコードされたメディアファイルを入力として取り込み、MPEG-2 トランスポートストリームでラップし、ここから HTTP Live Streaming の用途に適した同じ長さの一連のファイルを作成します。またメディアファイルセグメンタは、インデックスファイ

ル（プレイリスト）と復号キーも生成します。ファイルセグメンタは、ストリームセグメンタとほぼ同じく振る舞いますが、エンコーダから到着するストリームではなく既存のファイル进行操作します。詳細については、ターミナルウィンドウから`man mediafilesegmenter`と入力してください。

メディアストリームバリデータ

`mediastreamvalidator` コマンドラインツールは、サーバ上でインデックスファイル、代替ストリーム、メディアセグメントファイルを検証し、HTTP Live Streamingクライアントと連動するかどうかを判断するテストを行います。詳細については、ターミナルウィンドウから`man mediastreamvalidator`と入力してください。

バリエーションプレイリストクリエータ

`variantplaylistcreator` コマンドラインツールは、ストリームまたはファイルセグメンタの出力を使用して、各種のビットレートの代替ストリームのインデックスファイルをリストアップした、マスタインデックスファイルを作成します。バリエーションプレイリストクリエータに必要な出力を生成する場合、セグメンタを`-generate-variant-playlist` 引数で呼び出す必要があります。詳細については、ターミナルウィンドウから`man variantplaylistcreator`と入力してください。

メタデータタグジェネレータ

`id3taggenerator` コマンドラインツールは、ID3メタデータタグを生成します。これらのタグは、ファイルに書き込まれるか、出力されるストリームセグメントに挿入されます。詳細については、「[メタデータの追加](#)」（23 ページ）を参照してください。

セッションタイプ

HTTP Live Streaming プロトコルは、ライブ放送セッションとビデオオンデマンド（VOD）セッションをサポートしています。ライブセッションは、イベントの完全なレコードとして、またはユーザが要求する時間内に制限されたスライドウィンドウとして提示されます。

ライブセッションの場合、新しいメディアファイルが作成され、利用可能になると、インデックスファイルが更新されます。新しいインデックスファイルに、新しいメディアファイルがリストアップされます。次々に生成されるメディアファイルをインデックスの対象にしながら、古いメディアファイルをインデックスから削除して破棄します。このタイプのセッションは、連続的なブロードキャストに適しています。また新しいメディアファイルを既存のリストに追加するだけのインデックスもあります。このタイプのセッションは、イベントの終了後に簡単にVODに変換できます。

VODセッションの場合、表示の全体の時間に相当するメディアファイルが利用できます。インデックスファイルは静的ファイルであり、表示の開始以降に作成されたすべてのファイルを示した完全なリストを含みます。この種のセッションでは、クライアントはプログラム全体にフルにアクセスできます。

ビデオオンデマンドに即座に利用できる、イベントのライブ放送を作成することができます。ライブ放送をVODに変換する場合、サーバから古いメディアファイルを削除したり、インデックスファイルから古いメディアファイルのURLを削除したりしないでください。イベントの終了時に、`#EXT-X-ENDLIST`タグをインデックスに追加してください。この操作により、クライアントは後で放送を再開し、イベント全体を視聴することができます。また時間や手間をかけずに、放送用にイベントをアーカイブすることもできます。

VODは「canned（用意された）」データの配信にも使用できます。HTTP Live Streamingは、VODのプログレッシブなダウンロードよりも有利です。メディアの暗号化がサポートされ、また接続速度の変化に応じた、異なるデータレートのストリームの動的な切り替えにも対応しているためです。

（QuickTimeはプログレッシブなダウンロードによるmultiple-data-rateムービーにも対応していますが、QuickTimeムービーは、中間のデータレートの動的な切り替えをサポートしていません。）

コンテンツ保護

ストリームセグメントを含むメディアファイルは、個別に暗号化できます。暗号化が採用される場合、クライアントが復号キーを取得できるように、対応するキーファイルの参照がインデックスファイルに表示されます。

キーファイルがインデックスファイルに表示される場合、インデックスファイルのリスト内の後続メディアファイルの復号に必要な、暗号キーがキーファイルに含まれます。現在のHTTP Live Streamingは、16オクテットキーを使用するAES-128暗号化をサポートしています。キーファイルの形式は、これらの16オクテットのバイナリ形式のパッキング配列です。

Appleから提供されるメディアストリームセグメンタは、暗号化を実行し、3モードの暗号化設定をサポートしています。

1番目のモードは、ディスク上の既存のキーファイルのパスを指定する場合に使用します。このモードでは、セグメンタは既存のキーファイルのURLをインデックスファイルに挿入します。このキーを使用して、セグメンタはすべてのメディアファイルを暗号化します。

2番目のモードは、ランダムキーファイルの生成、ファイルの指定された場所への保存、およびインデックスファイル内でのファイルの参照をセグメンタに指示します。すべてのメディアファイルは、このランダムに生成されたキーを使って暗号化されます。

3番目のモードは、 n 個のメディアセグメントごとのランダムキーファイルの生成、ファイルの指定された場所への保存、およびインデックスファイル内でのファイルの参照をセグメントに指示します。このモードは、キーローテーションと呼ばれます。 n 個のファイルグループごとに、異なるキーを使って暗号化されます。

注意 すべてのメディアファイルは、同じキーを使って暗号化されるか、一定の間隔で新しいキーが必要になる場合があります。メディアファイルごとに1つのキーを使用するという理論的な制限がありますが、メディアの各キーにより、後続のメディアセグメントを表示するためのオーバーヘッドにファイル要求と送信が追加されるため、新しいキーに定期的に変換する方が、セグメントごとにキーを変更するよりもシステムのパフォーマンスへの影響を少なく抑えられます。

キーファイルは、HTTPまたはHTTPSを使って供給します。また独自のセッションベースの認証方式を使って、キーファイルの配信を保護する選択も可能です。詳細については、「[HTTPSによるキーファイルの安全な供給](#)」（31 ページ）を参照してください。

キーファイルは、暗号化されたメディアを復号するために初期化ベクトル（IV）を必要とします。IV はキーと同様に、定期的に変更できます。

プロトコルのキャッシュと送信

HTTPSはキーファイルの送信に共通に使用されます。HTTPSはまた、メディアセグメントファイルとインデックスファイルの送信にも使用できますが、スケーラビリティが重視される場合には推奨されません。HTTPSはウェブサーバキャッシュのバイパスを要求することが多く、これによりすべてのコンテンツ要求がサーバ経由でルート設定され、エッジネットワークディストリビューションシステムの目的にそぐわないためです。

しかし、まさにこの理由により、インデックスファイルが動的に変更されるライブ放送の場合、1つのメディアセグメント時間よりも長く、M3U8インデックスファイルがキャッシュされないことを、使用しているコンテンツ配信ネットワークに確実に認識させることが重要になります。

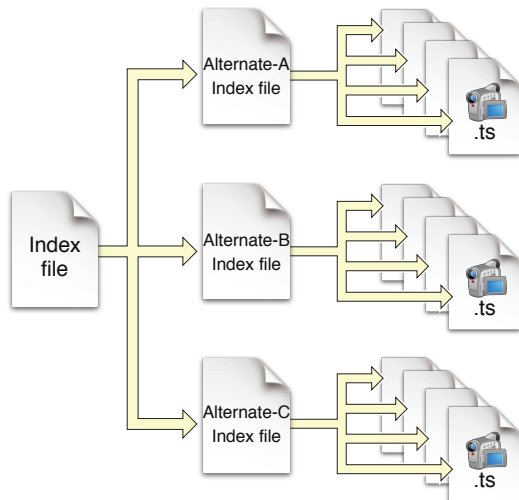
代替ストリーム

マスタインデックスファイルは、コンテンツの代替ストリームを参照する場合があります。参照を使用して、各帯域幅またはデバイスごとに品質レベルを変えて、同じコンテンツの複数のストリームの配信をサポートできます。HTTP Live Streamingは、利用可能な帯域幅が変更される場合の、ストリー

ムの動的な切り替えをサポートしています。クライアントソフトウェアは、ヒューリスティクスを使用して代替ストリームを切り替える適切なタイミングを判断します。現在、これらのヒューリスティクスは、測定対象のネットワークスループットの最新の傾向に基づきます。

マスタインデックスファイルは、図 2-1 に示すように、ほかのインデックスファイルを示す特別にタグの付いたリストを含め、メディアの代替ストリームを示します。

図 2-1 代替ストリーム



マスタインデックスファイルと代替インデックスファイルは、いずれも .M3U8 プレイリスト形式です。マスタインデックスファイルは、1度だけダウンロードされますが、ライブ放送の場合、代替インデックスファイルが定期的に再ロードされます。マスタインデックスファイルに最初に表示される代替ストリームは、最初に使用されるストリームです。その後はクライアントが、利用可能な帯域幅により代替ストリームを選択します。

クライアントは、モバイルデバイスがWiFiホットスポットに接続または切断する場合のように、任意に代替ストリームの変更する方法を選択できます。すべての代替ストリームは、ストリーム間をスムーズにトランジションさせるために、同一のオーディオを使用する必要があります。

variantplaylistcreator ツールを使用し、mediafilesegmenter ツールまたはmediastreamsegmenter ツールに `-generate-variant-playlist` オプションを指定すると、一連の代替ストリームを作成できます（詳細については、「[ツールのダウンロード](#)」（15 ページ）を参照）。

代替ストリームを使用する場合、以下の事項に留意する必要があります。

- バリエーションプレイリストへの最初のエントリは、ユーザがストリームに加入したときに再生され、最も適切なストリームを決定するテストの一環として使用されます。その他のエントリの順序は関連性がありません。

- 可能であれば、広範囲な接続速度で最高品質のストリームが供給されるように、十分なバリエーションをストリームをエンコードします。たとえば、バリエーションを150 kbps、350 kbps、550 kbps、900 kbps、1500 kbpsでエンコードします。
- 可能であれば、バリエーションプレイリストと個別の.M3U8プレイリストファイルで、相対パス名を使用します。
- すべてのストリームのオーディオトラックが正確に同一でなければ、ストリームの変更時のオーディオのアーティファクトを防ぐことができません。

注意 異なるVODストリームを同期させる場合、圧縮後、セグメント化の前に、メディアファイルのオーディオトラックをコピーし、これをほかの各メディアファイルに貼り付ける方法があります。

- 代替ストリームのビデオのビデオアスペクト比は、正確に同一でなければなりません。ただし同じアスペクト比を持つ場合、代替ストリームのピクセル寸法が異なっても構いません。たとえば、同じ4:3のアスペクト比を持つ2つの代替ストリームは、400 x 300と800 x 600など寸法が異なっても構いません。
- iOSアプリケーションデベロッパーであれば、最初の接続が携帯電話かWiFiのいずれであるかの判断をユーザのデバイスに照会し、適切なマスタインデックスファイルを選択することができます。初回のネットワーク接続の種類に関わりなく、ストリームが最初に再生されときのユーザ体験を高めるために、同じ代替インデックスファイルから構成され、先頭のストリームの異なる複数のマスタインデックスファイルを使用する必要があります。

携帯電話のバリエーションプレイリストには、150kのストリームが推奨されます。

Wi-Fiバリエーションプレイリストには、240kまたは440kのストリームが推奨されます。

注意 iOSベースのデバイスへのネットワーク接続タイプの照会方法の詳細については、以下のサンプルコードを参照してください。 *Reachability*。

- ストリームバリエーションにビットレートを指定する場合、所定のストリームで要求される実際の帯域幅にBANDWIDTH属性をほぼ一致させることが重要です。実際の帯域幅要件がBANDWIDTH属性と大きく異なる場合、ストリームの自動切り替えがスムーズに、あるいは正しく動作しない場合があります。

携帯電話ネットワークのビデオ送信

iPhoneやiPadなどのモバイル機器にビデオを送信する場合、クライアントのインターネット接続をいつでも携帯電話ネットワークに移動でき、いつでも切断できます。

HTTP Live Streamingでは、ネットワーク帯域幅の変更に従って、クライアントが動的に代替ストリームを選択できるため、デバイスが携帯電話とWiFi接続、または3GとEDGE接続の間を移動する場合など、最適なストリームを供給できます。これはプログレッシブなダウンロードに比べて、相当に有利な条件です。

ビデオオンデマンドの場合でも、すべての携帯電話対応のデバイスへのビデオ配信に、HTTP Live Streamingを使用することを強くお勧めします。変化する条件の下で最高の体験を、視聴者に提供できるためです。

また、低速のデータ接続に対しては、携帯電話対応のクライアントに64 Kbps以下の代替ストリームを提供します。64 Kbps以下で許容される品質のビデオを配信できない場合、音声専用のストリームか静止画像のオーディオを供給します。

携帯電話ネットワーク接続をターゲットとする場合、16:9コンテンツには400 x 224、4:3コンテンツには400 x 300のピクセル寸法を選択するのが適切です（「[iOSベースのデバイスへの配信用メディアの準備](#)」（24 ページ）を参照）。

アプリケーションの要件



Warning App Storeでディストリビューション用に提供されるiOSアプリケーションは、以下の要件に適合する必要があります。

アプリケーションが携帯電話ネットワークにビデオを配信する場合、ビデオの長さが10分以上であるか、5分間のデータ量が5MB以上であれば、HTTP Live Streamingの使用が要求されます。（サイズの小さいクリップには、プログレッシブなダウンロードを使用できます。）

アプリケーションが携帯電話ネットワーク向けのHTTP Live Streamingを使用する場合、帯域幅が64 Kbps以下の1つ以上のストリームの送信が要求されます（低帯域幅のストリームはオーディオ専用または静止画のオーディオです）。

上記の要件は、App Storeでディストリビューション用に提供される、Apple製品用のiOSアプリケーションに適用されます。互換性のないアプリケーションは、Appleの意向で拒絶または削除される場合があります。

フェイルオーバー保護

プレイリストに代替ストリームが含まれる場合、代替帯域幅または代替デバイスとして動作する以外に、失敗のフォールバックとしても動作します。iOS 3.1以降、クライアントがストリームのインデックスファイルを再ロードできなくなると（404エラーなどを原因として）、クライアントは代替ストリームへの切り替えを試みます。

特定のストリームでインデックスのロードが失敗した場合、クライアントはネットワーク接続がサポートする帯域幅の最も高い代替ストリームを選択します。同じ帯域幅の代替ストリームが複数存在する場合、クライアントはプレイリストの表示順に代替ストリームを選択します。

この機能を使用すると、サーバのクラッシュやコンテンツディストリビュータノードの停止などの、ローカルで深刻な障害が発生した場合でもメディアのクライアントへの到達を可能にする、冗長ストリームを提供することができます。

フェイルオーバー保護を実装する場合、ストリームまたは複数の代替帯域幅ストリームを作成し、通常と同様にプレイリストファイルを生成します。次に、独立したサーバまたはコンテンツディストリビューションサービスで、平行ストリームまたはストリームセットを作成します。プレイリストファイルにバックアップストリームのリストを追加し、各帯域幅のバックアップストリームがプライマリストリームの後にリストされるようにします。たとえば、プライマリストリームがサーバALPHAから到着し、バックアップストリームがサーバBETA上に存在する場合、プレイリストファイルには以下のように表示されます。

```
#EXTM3U

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=200000
http://ALPHA.mycompany.com/lo/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=200000
http://BETA.mycompany.com/lo/prog_index.m3u8

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=500000
http://ALPHA.mycompany.com/md/prog_index.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=500000
http://BETA.mycompany.com/md/prog_index.m3u8
```

バックアップストリームは、プレイリスト内のプライマリストリームと混合され、各帯域幅のプライマリストリームの後にリストされるバックアップストリームと混合されることに注意してください。

バックアップストリームは、1組に制限されません。上記の例では、ALPHAとBETAの後に、GAMMAなどが続く可能性があります。同様に、完全な平行ストリームセットを提供する必要はありません。バックアップサーバで、低帯域幅のストリームを1つ提供する、などの処理を行います。

タイムドメタデータの追加

メディアストリームセグメントに、さまざまな種類のメタデータを追加できます。たとえば、オーディオストリームにアルバムのジャケット画像、アーティスト名、曲名を追加できます。また、野球の試合のビデオに画像内のバッターの名前と統計情報を追加できます。

音声専用ストリームに画像がメタデータとして含まれている場合、Appleクライアントソフトウェアは画像を自動的に再生します。現在、Appleから提供されるクライアントソフトウェアで自動的に再生される唯一のメタデータは、音声専用ストリームに付随する静止画像です。

ただし、MPMoviePlayerControllerまたはAVPlayerItemを使用して、独自のクライアントソフトウェアを作成する場合、timedMetaDataプロパティを使用してストリーミングメタデータにアクセスできます。

ストリームセグメントまたはファイルセグメントの-fコマンドラインオプションで、メタデータファイルを追加する方法で、タイムドメタデータを追加することができます。指定されたメタデータソースは、ID3形式のファイルか画像ファイル（JPEGまたはPNG）です。この方法で指定されたメタデータは、自動的に各メディアセグメントに挿入されます。

これがタイムドメタデータと呼ばれるのは、一定の時間オフセットでメディアストリームにメタデータが挿入されるためです。オプションで、一定の時間の後に、すべてのセグメントにメタデータを挿入することもできます。

ライブストリームにタイムドメタデータを追加する場合、id3taggeneratorツールを使用して、出力をストリームセグメントに設定します。このツールはID3メタデータを生成し、送信ストリームにインクルードされるように、このメタデータをストリームセグメントに渡します。

たとえば、タグジェネレータをシェルスクリプトから実行して、所定の時間または所定の間隔でメタデータを挿入することができます。新しいタイムドメタデータは、自動的に既存のメタデータと置換されます。

メタデータがメディアセグメントに挿入されると、永続的に存続します。ライブ放送がビデオオンデマンドとして再利用される場合など、本放送の間に挿入されたすべてのメタデータが保持されています。

ファイルセグメントを使って作成されたストリームにタイムドメタデータを追加する方法は、多少複雑です。

1. まず、メタデータサンプルを生成します。ID3メタデータは、id3taggeneratorコマンドラインツールを使用し、出力をファイルに設定して生成します。
2. 次に、**メタデータマクロファイル**を作成します。これは、各行にメタデータの挿入時間、メタデータのタイプ、メタデータファイルのパスとファイル名を示したテキストファイルです。

たとえば、以下のメタデータマクロファイルの場合、1.2秒後に写真がストリームに挿入され、10秒後にIDタグが挿入されます。

```
1.2 picture /meta/images/picture.jpg
```

```
10 id3 /meta/id3/title.id3
```

- 最後に、メディアファイルセグメントを呼び出すときに、`-M`コマンドラインオプションを使用して、メタデータマクロファイルを名前で指定します。

詳細については、`mediastreamsegmenter`、`mediafilesegmenter`、`id3taggenerator`の、各manページを参照してください。

クローズドキャプションの追加

HTTP Live Streamingは、ストリームへのクローズドキャプションの追加をサポートしています。

注意 クローズドキャプションをサブタイトルと混乱しないでください。

ストリームセグメントを使用する場合、ATSC A/72の指定に従って、MPEG-2トランスポートストリーム（メインのビデオエレメンタリストリーム内）にCEA-608クローズドキャプションを追加する必要があります。

ファイルセグメントを使用する場合、メディアをQuickTimeムービーファイルにカプセル化し、クローズドキャプショントラック（'clcp'）を追加します。

ムービーファイルまたはMPEG-2トランスポートストリームにクローズドキャプションを追加するのに使用するツールの一つに、Final Cut Studioに付属するCompressorがあります。Compressorは、Scenaristソフトウェアと連動してクローズドキャプションを生成します。詳細については、Compressorのマニュアルを参照してください。

アプリケーションを作成する場合、AVFoundationフレームワークがクローズドキャプションの再生をサポートします。

メディアのiOSベースのデバイスへの配信準備

iOSベースのデバイスで使用される、ストリームの推奨エンコード設定を、以下の4つの表に示します。ライブストリームの場合、これらの設定はハードウェアまたはソフトウェアエンコーダから利用できます。ビデオオンデマンドをマスタファイルから再エンコードする場合、Compressorなどのビデオ編集ツールを使用できます。

ファイルセグメンタのファイル形式は、指定されたエンコーディングを使用した、QuickTimeムービー、MPEG-4ビデオ、MP3ビデオのいずれかになります。

ストリームセグメンタのストリーム形式は、MPEG-2トランスポートストリームでラップし、以下のエンコーディングを使用したMPEGエレメンタリオディオ・ビデオストリームとします。

- H.264圧縮形式を使用したビデオのエンコーディング
 - iPhone 3G以降と第2世代以降のiPod touchは、H.264 Baseline 3.1をサポートします。ただし、旧バージョンのiPhoneまたはiPod touchでアプリケーションが実行される場合、互換性の理由によりH.264 Baseline 3.0を使用する必要があります。
 - iPad、Apple TV 2以降、およびiPhone 4以降については、Baselineプロファイル3.0、Baselineプロファイル3.1、Mainプロファイル3.1のいずれかを使用できます。

注意 Baselineプロファイルは、Mainプロファイルと同じではありません。iPhoneとiPod touchは、Baselineプロファイルを使用します。iPadとApple TVは、BaselineまたはMainプロファイルを使用します。Mainプロファイルは、コンテンツを大画面のデバイス専用で作成している場合に使用します。Baselineプロファイルは、コンテンツが小型画面専用、または大画面と小型画面の両方での閲覧が予定されている場合に使用します。

- 200 kbps未満のビデオストリームについては、フレームレート10 fpsを推奨します。300 kbps未満のビデオストリームの場合、12~15 fpsのフレームレートが推奨されます。その他のストリームについては、29.97のフレームレートが推奨されます。
- 以下のいずれかでオーディオをエンコードします。
 - HE-AACまたはAAC-LC、ステレオ
 - MP3（MPEG-1 Audio Layer 3）、ステレオ
- いずれの場合も、オーディオサンプルレート22.05 kHz、オーディオビットレート40 kbpsが推奨されます。

表 2-1 iPhone、iPod touch、iPad、Apple TVのエンコーダ設定、アスペクト比16:9

接続	寸法	合計 ビットレート	ビデオ ビットレート	キーフレーム
携帯電話	480 x 320	64 kbps	オーディオ専用	なし
携帯電話	480 x 224	150 kbps	110 kbps	30
携帯電話	480 x 224	240 kbps	200 kbps	45
携帯電話	480 x 224	440 kbps	400 kbps	90

接続	寸法	合計 ビットレート	ビデオ ビットレート	キーフレーム
WiFi	640 x 360	640 kpbs	600 kbps	90

表 2-2 iPhone、iPod touch、iPad、Apple TVのエンコーダ設定、アスペクト比4:3

接続	寸法	合計 ビットレート	ビデオ ビットレート	キーフレーム
携帯電話	480 x 320	64 kpbs	オーディオ専用	なし
携帯電話	480 x 300	150 kpbs	110 kbps	30
携帯電話	480 x 300	240 kpbs	200 kbps	45
携帯電話	480 x 300	440 kpbs	400 kbps	90
WiFi	640 x 480	640 kpbs	600 kbps	90

表 2-3 エンコーダ追加設定（iPadとApple TVのみ）アスペクト比16:9

接続	寸法	合計 ビットレート	ビデオ ビットレート	キーフレーム
WiFi	640 x 360	1240 kpbs	1200 kbps	90
WiFi	960 x 540	1840 kpbs	1800 kbps	90
WiFi	1280 x 720	2540 kpbs	1500 kbps	90
WiFi	1280 x 720	4540 kpbs	4500 kbps	90

表 2-4 エンコーダ追加設定（iPadとApple TVのみ）アスペクト比4:3

接続	寸法	合計 ビットレート	ビデオ ビットレート	キーフレーム
WiFi	640 x 480	1240 kpbs	1200 kbps	90
WiFi	960 x 720	1840 kpbs	1800 kbps	90
WiFi	960 x 720	2540 kpbs	2500 kbps	90
WiFi	1280 x 960	4540 kpbs	4500 kbps	90

サンプルストリーム

Appleのデベロッパサイトでは、一連のテスト用のHTTPストリームが利用できます。これらのサンプルには、ストリームを埋め込むためのHTMLの正しいフォーマット化、ストリームをインデックス化する.M3U8ファイル、.tsメディアセグメントファイルが示されます。ストリームには、以下のURLでアクセスできます。

- <http://devimages.apple.com/iphone/samples/bipbopgear1.html>
- <http://devimages.apple.com/iphone/samples/bipbopgear2.html>
- <http://devimages.apple.com/iphone/samples/bipbopgear3.html>
- <http://devimages.apple.com/iphone/samples/bipbopgear4.html>
- <http://devimages.apple.com/iphone/samples/bipbopall.html>

サンプルには、4種類の解像度とデータレートに共通のNTSCテストパターンが示されます。最後のサンプルは、複数のデータレートでストリーミングします。ストリームはサンプル1から開始し、接続でサポートされる最速のサンプルに切り替わります。

iPhoneまたはiPod touchで上記のサンプルを再生する場合、iOSバージョン3.0以降をインストールしている必要があります。デスクトップでの再生には、Safari 4.0以降が必要です。

HTTP Live Streamingの配備

実際にHTTP Live Streamingを配備する場合、ブラウザ用のHTMLページか、レシーバとして動作するクライアントアプリケーションを作成する必要があります。またウェブサーバを使用し、ライブストリームをMPEG-2トランスポートストリームとしてエンコードする方法か、ソースマテリアルからH.264およびAACエンコーディングでMP3またはMPEG-4メディアファイルを作成する方法を使用する必要があります。

Appleから提供されるツールを使用すると、ストリームまたはメディアファイルをセグメント化し、またインデックスファイルとバリエーションプレイリストを生成することができます（「[ツールのダウンロード](#)」（15 ページ）を参照）。

ストリームを配信する前に、Appleが配布するメディアストリームバリデータを使用して、HTTP Live Streamingと完全に互換性があるかどうかを確認する必要があります。

ストリームの暗号化も可能です。この場合、配信対象のクライアントのみが復号できるように、HTTPSで暗号化キーファイルを安全に送信する必要があります。

HTMLページの作成

HTTP Live Streamingメディアを配布する場合、.M3U8プレイリストファイルをビデオソースに使用して、HTML5 <video>タグを含むウェブページを作成するのが最も簡単な方法です。リスト 3-1に例を示します。

リスト 3-1 ウェブページを使ったHTTP Live Streamingの配布

```
<html>
<head>
  <title>HTTP Live Streaming Example</title>
</head>
<body>
  <video
    src="http://devimages.apple.com/iphone/samples/bipbop/bipbopall.m3u8"
    height="300" width="400"
  >
```

```
</video>  
</body>  
</html>
```

HTML5 videoエレメントをサポートしないブラウザの場合、またはHTTP Live Streamingをサポートしないブラウザの場合、<video>タグと</video>タグの間にフォールバックコードをインクルードできます。たとえば、QuickTimeプラグインを使用して、プログレッシブダウンロードまたはRTSPストリームにフォールバックできます。例については、『*Safari HTML5 Audio and Video Guide*』を参照してください。

ウェブサーバの設定

HTTP Live Streamingは、通常のウェブサーバから配布できます。配布されるファイルのMIMEタイプをファイル拡張子に関連付ける以外、特別な設定は不要です。

HTTP Live Streamingの以下のMIMEタイプを設定します。

ファイル拡張子	MIMEタイプ
.M3U8	application/x-mpegURLまたは vnd.apple.mpegURL
.ts	video/MP2T

ウェブサーバのMIMEタイプが制約される場合、.m3uで終わる、MIMEタイプaudio/mpegURLのファイルを配布して互換性に対応します。

インデックスファイルは長く、頻繁に再ダウンロードされますが、テキストファイルであるため非常に効率的に圧縮できます。.M3U8インデックスファイルのオンザフライgzip圧縮を有効にすると、サーバのオーバーヘッドを減らすことができます。HTTP Live Streamingクライアントは、圧縮されたインデックスファイルを自動的に解凍します。

適切なキャッシュ動作をダウンストリームのウェブキャッシュに適用する場合、.M3U8ファイルのtime-to-live (TTL) 値を小さくする必要が生じる場合もあります。これらのファイルはライブ放送の間にオーバーライトされることが多く、それぞれのリクエストに対して最新のバージョンをダウンロードしなければならないためです。具体的な推奨値については、コンテンツの配信サービスプロバイダに確認してください。VODの場合、インデックスファイルは静的ファイルであり、ダウンロードは1度だけ行われるため、キャッシュは問題になりません。

ストリームの検証

mediastreamvalidatorツールは、HTTP Live Streamingストリームとサーバを検証するためのコマンドラインユーティリティです（ツールの入手方法については、「[ツールのダウンロード](#)」（15 ページ）を参照）。

メディアストリームバリデータは、HTTP Live Streamingセッションをシミュレートし、インデックスファイルとメディアセグメントがHTTP Live Streamingの仕様に適合しているかどうかを検証します。ストリーミングの信頼性を確認するために、複数の検査を実行します。エラーまたは問題が検出されると、詳細な診断レポートが表示されます。

新しいストリームまたは代替ストリームセットを配信する前に、必ずバリデータを実行してください。

メディアストリームバリデータには、配信するストリームに続いて、各ストリームのタイミング結果がリストで表示されます。（実際のタイミングの計算には数分を要します。）バリデータの出力例を以下に示します。

```
Validating http://devimages.apple.com/iphone/samples/bipbop/gear3/prog_index.m3u8
against iPhone OS 3.1.0
```

```
Average segment duration: 8.77 seconds
```

```
Average segment bitrate: 510.05 kbit/s
```

```
Average segment structural overhead: 96.37 kbit/s (18.89 %)
```

```
Video codec: avc1
```

```
Video resolution: 480x360 pixels
```

```
Video frame rate: 29.97 fps
```

```
Average video bitrate: 407.76 kbit/s
```

```
H.264 profile: Baseline
```

```
H.264 level: 2.1
```

```
Audio codec: aac
```

```
Audio sample rate: 22050 Hz
```

```
Average audio bitrate: 5.93 kbit/s
```

HTTPSを使ったキーファイルの安全な送信

暗号化によりメディアを保護できます。ファイルセグメントとストリームセグメントはいずれも、暗号化オプションが利用でき、両セグメントに暗号化キーの定期的な変更を指示することもできます。キーを共有する相手はデベロッパが決定します。

キーファイルは、暗号化されたメディアを復号するために初期化ベクトル (IV) を必要とします。IV はキーと同様に、定期的に変更できます。オーバーヘッドを最小限に抑えながらメディアを暗号化する場合、3~4時間ごとにキーを変更し、50 MbのデータごとにIVを変更するというのが最新の推奨内容です。

ただし、キーへのアクセスが制限されている場合でも、キーファイルがHTTPで送信された場合に、そのコピーを入手して傍受することが可能です。この問題は、HTTPSを使ってキーを安全に送信する方法で解決できる場合があります。

HTTPSを使ったキーファイルの送信を試みる前に、内部ウェブサーバからHTTPにキーを送信するテストを実施することを推奨します。このテストにより、HTTPSを追加する前に設定をデバッグできます。HTTPでの動作を確認できたら、HTTPSへの切り替えをいつでも実行できます。

HTTPSを使用してHTTP Live Streamingのキーを送信するためには、3つの条件を満たさなければなりません。

- 信頼された認証機関により署名されたSSL証明書を、HTTPSサーバにインストールする必要があります。
- キーファイルの認証ドメインは、最初のプレイリストファイルの認証ドメインと同じでなければなりません。これには、HTTPSサーバからバリエーションプレイリストファイルを送信するのが最も簡単な方法です。バリエーションプレイリストファイルのダウンロードは1度だけであるため、余分な負荷は生じません。ほかのプレイリストファイルは、HTTPを使用して送信できます。
- ユーザの認証のために独自のダイアログを起動するか、クライアントデバイスに資格情報を保存する必要があります。認証用のユーザダイアログは、HTTP Live Streamingから提供されません。独自のクライアントアプリケーションを作成する場合、CookieベースまたはHTTPダイジェストベースの資格情報を保存し、この資格情報を`didReceiveAuthenticationChallenge`コールバックに指定します（詳細については、「[NSURLConnectionの使用](#)」と「[認証チャレンジ](#)」を参照）。指定した資格情報はキャッシュされ、メディアプレイヤーにより再利用されます。

Important HTTP Live StreamingでHTTPSサーバを使用するためには、信頼された認証機関が署名したSSL証明書を取得する必要があります。

信頼された認証機関により署名されたSSL証明書をHTTPSサーバが取得していない場合、自己署名SSL認証局と、サーバのリーフ証明書を作成して、設定をテストすることができます。認証局の証明書を電子メールに添付し、Live Streamingクライアントとして使用するデバイスに送信し、メールの添付ファイルをタップして、デバイスのサーバに対する信頼性を確立します。

よく訊ねられる質問

1. どのような種類のエンコーダがサポートされていますか？

プロトコル仕様では、エンコーダの選択を制限していません。ただし、Appleの現在の実装では、H.264ビデオとAACオーディオ（HE-AACまたはAAC-LC）を含む、MPEG-2トランスポートストリームを生成するエンコーダとの相互運用が確認されています。UDPによる出力ストリームのブロードキャストに対応したエンコーダは、Appleから供給され、現在実装されているセグメントソフトウェアとの互換性も必要です。

2. サポートされるビデオおよびオーディオ形式の詳細について教えてください。

プロトコル仕様ではビデオとオーディオの形式を制限していませんが、現在のAppleの実装では以下の形式がサポートされています。

- ビデオ：H.264 Baseline Level 3.0、Baseline Level 3.1、Main Level 3.1
- オーディオ：
 - HE-AACまたはAAC-LC、最大48 kHz、ステレオ
 - MP3（MPEG-1 Audio Layer 3）8 kHz～48 kHz、ステレオ・オーディオ

注意 iPad、iPhone 3G、iPod touch（第2世代以降）は、H.264 Baseline 3.1をサポートします。ただし、旧バージョンのiPhoneまたはiPod touchでアプリケーションが実行される場合、互換性の理由によりH.264 Baseline 3.0を使用する必要があります。コンテンツがiPad、Apple TV、iPhone 4以降、およびMac OS Xコンピュータ専用であれば、Main Level 3.1を使用する必要があります。

3. メディアファイルの長さ（時間）はどれぐらいですか？

主に考慮すべき点は、セグメントが短くなるとインデックスファイルの更新回数が増え、結果的にクライアント側で不要なネットワークのオーバーヘッドが生じるということです。セグメントが長くなると、ブロードキャスト固有の遅延が増長し、初回の起動時間が長くなります。10秒間のメディアファイルが、ほとんどのブロードキャストコンテンツに妥当な時間とされています。

4. 連続的に継続されるセッションでは、インデックスファイルに何ファイルまで入りますか？

通常の推奨ファイル数は3ですが、最適な数が多くなる場合もあります。

最適な数を選択する場合の重要な考慮事項は、ライブセッション中に利用できるファイルの数により、クライアントの再生/一時停止および検索操作のときの動作が制約されるということです。リストのファイルが増えると、クライアントがブロードキャストの再生場所を記憶した状態で一

時停止できる時間が長くなり、新しいクライアントがストリームに参加するときのブロードキャスト内の位置が一層後退し、クライアントが検索する期間が拡大します。この場合、インデックスファイルが長くなるとネットワークのオーバーヘッドが増えるというトレードオフがあります。ライブ放送の間、クライアントは定期的に一斉にインデックスファイルを更新するため、このような現象になります。インデックスファイルが小さい場合でも同様です。

5. サポートされるデータレートを教えてください。

コンテンツプロバイダがストリームに選択するデータレートは、ターゲットクライアントのパフォーマンスと予測されるネットワークポロジの影響を最も多く受けます。ストリーミングのプロトコルそのものにより、使用可能なデータレートに制限は課せられません。現在の実装は、64 Kbpsを下限、3 Mbpsを上限としたデータレートの、iPhone向けのオーディオビデオストリームを使ってテストされています。低速の携帯電話接続の配信には、代替ストリームとして64 Kbpsのオーディオ専用ストリームが推奨されます。

推奨されるデータレートについては、「[メディアのiOSベースのデバイスへの配信準備](#)」（24ページ）を参照してください。

注意 データレートが利用可能な帯域幅を超えた場合、起動前の遅延が延長され、クライアントは定期的な一時停止して、増大したデータをバッファリングする必要が生じる場合があります。ブロードキャストが、コンテンツ内の移動ウィンドウを提供するインデックスファイルを使用する場合、クライアントは最終的に脱落し、1つまたは複数のセグメントがドロップします。VODの場合、セグメントは損失しませんが、帯域幅が不十分であれば起動が遅くなり、データのバッファリング中に断続的に停止します。

6. .tsファイルとは何ですか？

.tsファイルには、MPEG-2トランスポートストリームが含まれます。これはエンコードされた一連のメディアサンプル、通常はオーディオとビデオをカプセル化するファイル形式です。このファイル形式は、MP3オーディオ、AACオーディオ、H.264ビデオなどを含むさまざまな圧縮形式をサポートします。Apple HTTP Live Streamingの実装では、現在、サポートされていない圧縮形式もあります。（サポートされている形式のリストについては、「[メディアエンコード](#)」（11ページ）を参照してください。

MPEG-2トランスポートストリームはコンテナです。MPEG-2圧縮形式と混乱しないでください。

7. .M3U8ファイルとは何ですか？

.M3U8ファイルは、拡張可能なプレイリストファイルの形式です。UTF-8エンコーディングテキストを含むm3uプレイリストです。m3uファイル形式は、メディアファイルのURLのリストの搬送に適した、デファクトスタンダードのプレイリスト形式です。これはHTTP Live Streamingのインデックスファイルとして使用される形式です。詳細については、[HTTP Live Streamingの仕様のIETFインターネットドラフト](#)を参照してください。

8. クライアントソフトウェアは、ストリームを切り替えるタイミングをどのように判断しますか？

現在実装されているクライアントは、ストリームの再生中に有効な帯域幅を観察します。高品質のストリームが利用可能であり、帯域幅がそのストリームに対応できると見なされる場合、クライアントは現在よりも品質の高いストリームに切り替えます。低品質のストリームが利用可能であり、現在の帯域幅が現在のストリームに対応できないと見なされる場合、クライアントは現在よりも品質の低いストリームに切り替えます。

注意 代替ストリーム間をシームレスに移行するために、ストリームのオーディオ部分は全バージョンを通じて同一でなければなりません。

9. Appleから配布されるメディアストリームセグメンタのコピーは、どこで入手できますか？

メディアストリームセグメンタ、ファイルストリームセグメント、その他のツールは頻繁に更新されるため、最新版のHTTP Live StreamingツールをAppleデベロッパウェブサイトからダウンロードする必要があります。詳細については、「[ツールのダウンロード](#)」（15 ページ）を参照してください。

10. Appleのメディアセグメンタで使用する、通常のHTTPストリームと代替ストリームに推奨される設定の内容を教えてください。

「[メディアのiOSベースのデバイスへの配信準備](#)」（24 ページ）を参照してください。

上記の設定が現在の推奨設定です。特定の要件も設けられています。現在のmediastreamsegmenterツールは、ISO/IEC 13818で定義されるMPEG-2トランスポートストリームでの操作に限定されます。このトランスポートストリームには、必ずH.264（MPEG-4、パート10）ビデオと、AACまたはMPEGオーディオが含まれます。AACオーディオが使用される場合、ADTSヘッダが含まれなければなりません。H.264ビデオアクセス装置は、アクセスユニットデリミタNALを使用すること、また固有のPES/パケット内に存在することが要求されます。

セグメンタの一部の設定はユーザ設定可能です。コマンドラインの引数とそれぞれの意味のリストは、ターミナルアプリケーションからman mediastreamsegmenterを入力すると取得できます。ターゲット期間（メディアセグメントの長さ）には10秒が推奨され、ターゲット期間が指定されていない場合は、10秒がデフォルトになります。

11. ストリームの再生に必要なコーデックとH.264プロファイルを、どのように指定しますか？

EXT-X-STREAM-INFタグのCODECS属性を使用します。この属性が指定される場合、ストリームの再生に必要なすべてのコーデックとプロファイルを含める必要があります。現在、以下の値が承認されています。

AAC-LC	"mp4a.40.2"
HE-AAC	"mp4a.40.5"
MP3	"mp4a.40.34"

H.264 Baselineプロファイルレベル3.0	"avc1.42001e"または"avc1.66.30" 注：iOSバージョン3.0～3.12との互換性の理由により、「avc1.66.30」を使用してください。
H.264 Baselineプロファイルレベル3.1	"avc1.42001f"
H.264 Mainプロファイルレベル3.0	"avc1.4d001e"または"avc1.77.30" 注：iOSバージョン3.0～3.12との互換性の理由により、「avc1.77.30」を使用してください。
H.264 Mainプロファイルレベル3.1	"avc1.4d001f"

属性値は引用符で囲みます。複数の値を指定する場合、1組の引用符ですべての値を囲み、各値はコンマで区切ります。以下に例を示します。

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=500000
mid_video_index.M3U8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=800000
wifi_video_index.M3U8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=3000000,
CODECS="avc1.4d001e,mp4a.40.5"
h264main_heaac_index.M3U8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=64000, CODECS="mp4a.40.5"
aacaudio_index.M3U8
```

12. オーディオ/ビデオ入力から、どのような方法でオーディオ専用ストリームを作成しますか？

ストリームまたはファイルセグメントを呼び出すときに、`-audio-only`引数を追加します。

13. オーディオ専用ストリームに、どのような方法で静止画像を追加しますか？

ストリームまたはファイルセグメントを呼び出すときに`-meta-file`引数を使用し、画像をすべてのセグメントに追加する`-meta-type=picture`を指定します。この指定により、たとえば、画像`poster.jpg`が、ファイル`track01.mp3`から作成されたオーディオストリームのすべてのセグメントに追加されます。

```
mediafilesegmenter -f /Dir/outputFile -a --meta-file=poster.jpg --meta-type=picture
track01.mp3
```

留意事項として、通常、画像は10秒ごとにリセットされるため、ファイルサイズを小さく抑えるのが適切です。

14. オーディオビデオストリームの代替としての、オーディオ専用ストリームをどのように指定しますか?

EXT-X-STREAM-INFタグのCODECS属性とBANDWIDTH属性を同時に使用します。

BANDWIDTH属性は、それぞれの代替ストリームに必要な帯域幅を指定します。利用可能な帯域幅がオーディオ代替ストリームに十分であり、最下位のビデオ代替ストリームに十分ではない場合、クライアントはオーディオストリームに切り替えます。

CODECS属性がインクルードされる場合、この属性はストリームの再生に必要なすべてのコーデックをリスト化します。オーディオコーデックのみが指定される場合、ストリームはオーディオ専用と特定されます。現在、ストリームのオーディオ専用を指定する必要があるため、CODECS属性の使用はオプションです。

高速接続用に500 Kbpsのビデオストリームを、低速接続用に150 Kbpsのビデオストリームを、超低速接続用に64 Kbpsのオーディオ専用ストリームを指定する例を以下に示します。可聴的な混乱を起こさず、ストリーム間の移行を可能にするために、すべてのストリームが同じ64 Kbpsオーディオを使用する必要があります。

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=500000
mid_video_index.M3U8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=150000
3g_video_index.M3U8
#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=64000, CODECS="mp4a.40.5"
aacaudio_index.M3U8
```

15. サーバのハードウェア要件または推奨事項について教えてください。

エンコーダのハードウェア推奨事項は、質問No.1を参照してください。

Appleストリームセグメンタは、Intel搭載Macでの稼働に対応しています。Mac ProやXserveなど、イーサネットネットワークインターフェイスを2基内蔵したMacの使用をお勧めしています。一方のネットワークインターフェイスは、エンコードされたストリームをローカルネットワークから取得するために使用し、もう一方のネットワークインターフェイスは広域ネットワークへのアクセスに使用します。

16. Appleで実装されるHTTP Live StreamingはDRMをサポートしていますか?

サポートしていません。ただし、メディアの暗号化と、HTTPS認証によるキーアクセスの制限が可能です。

17. どのクライアントプラットフォームがサポートされていますか?

iPhone、iPad、iPod touch (iOSバージョン3.0以降を要求)、Apple TV (バージョン2以降)、Mac OS Xコンピュータ。

18. プロトコル仕様は入手できますか?

はい。プロトコル仕様は、IETFインターネットドラフト (<http://tools.ietf.org/html/draft-pantos-http-live-streaming>) から入手できます。

19. クライアントはコンテンツをキャッシュしますか?

コンテンツのキャッシュを禁止するクライアントへの命令を、インデックスファイルに含めることができます。命令を含めなければ、クライアントはメディア内の検索の間に、データをキャッシュしてパフォーマンスを最適化する場合があります。

20. 実時間配信システムですか?

違います。ストリームセグメントを含むメディアファイルのサイズと時間に応じた、固有の遅延が生じます。1つ以上のセグメントが完全にダウンロードされなければ、クライアントからの閲覧は不可能です。セグメント間のシームレスな移行を確保するためには、2つのセグメントが必要です。また、エンコーダとセグメンタは入力からファイルを作成する必要があります。このファイルの時間の長さが、メディアがダウンロード可能になるまでの最低の遅延になります。推奨される設定の通常の遅延は約30秒です。

21. 遅延の長さはどのくらいですか?

推奨される設定で、約30秒です。質問No.15を参照してください。

22. ハードウェアエンコーダを使用する必要がありますか?

ありません。プロトコル仕様を使用した場合、ソフトウェアエンコーダの実装が可能です。

23. このアプローチはRTP/RTSPに対して、どのような利点がありますか?

HTTPはルータ、NAT、ファイアウォール設定で禁止されることはほとんどありません。デフォルトで通常は閉鎖されているポートを、開く必要がありません。したがって、特別な設定を必要とせずに、より多くの場所のクライアントにコンテンツを配信することができます。また、HTTPは多くのコンテンツディストリビューションネットワークでサポートされているため、大規模な分散モデルにはコストに影響する場合があります。一般に、変更を加えずに動作する、またRTP/RTSPではなくHTTPの使用を予定したハードウェアとソフトウェアの数は増えています。PHPなどのツールを使用した、HTTPコンテンツ配信のカスタマイズに関する専門知識も、広く普及しています。

また、HTTP Live Streamingは、Safariでサポートされ、iOSのメディアプレイヤーフレームワークでもサポートされています。RTSPストリーミングはサポートされていません。

24. ストリームの全体のビットレートが、オーディオとビデオの合計ビットレートよりも高くなるのはなぜですか?

MPEG-2トランスポートストリームは、相当のオーバーヘッドを含んでいる可能性があります。このストリームは固定パケットサイズを活用し、パケットのコンテンツがデフォルトのパケットサイズを下回るとパディングされます。エンコーダとマルチプレクサの実装は、このような固定パケットサイズにメディアデータをパッキングするときの効率性が異なります。パディングの量は、フレームレート、サンプルレート、解像度により異なります。

25. どのような方法で、オーバーヘッドを減らし、ビットレートを下げられるでしょうか?

より効率的なエンコーダを使用すると、エンコーダの設定をチューニングする場合と同様に、オーバーヘッドの量を減らせる場合があります。

注意 ストリームセグメンタの最適化は、デフォルトで有効に設定されています。

26. すべてのメディアファイルを、同じMPEG-2トランスポートストリームに組み込むのでしょうか?

ちがいます。異なるトランスポートストリームのメディアファイルを混合できます。その場合、これらのファイルはEXT-X-DISCONTINUITYタグで区切られている必要があります。詳細については、プロトコルの仕様を参照してください。ただし、最適な結果を得るためには、すべてのビデオメディアファイルの高さと幅の寸法（ピクセル数）を同一にする必要があります。

27. HTTPオーディオ/ビデオサーバの設定に関する、ヘルプあるいはアドバイスはどこで閲覧できますか?

Appleデベロッパフォーラム (<http://devforums.apple.com/>) を訪問してください。

また『*Best Practices for Creating and Deploying HTTP Live Streaming Media for the iPhone and iPad*』も確認してください。

書類の改訂履歴

この表は「*HTTP Live Streamingの概要*」の改訂履歴です。

日付	メモ
2011-04-01	クローズドキャプションのサポートの記述、iOSベースのデバイス向けのメディア準備のベストプラクティス、HTTPSによるキーファイルの送信方法の記述を追加しました。
2010-11-15	タイムドメタデータの記述を追加しました。
2010-03-25	ストリームセグメンタのiPadおよび最適化オプションを追加して更新しました。
2010-02-05	携帯電話ネットワークを使用した、ビデオ用のHTTP Live Streamingを使用するためのアプリケーションの要件と、64 Kbpsストリームの要件を定義しました。
2010-01-20	ツールとサンプルストリームのリンクを更新しました。CODECS属性のドキュメントと、ビデオストリームおよびオーディオ専用代替ストリームを追加しました。
2009-11-17	誤植を修正し、サンプルのURLを訂正しました。
2009-09-09	フェイルオーバーサポートを追加して更新しました。
2009-06-04	サンプルコード、QuickTime Xのサポートを追加しました。ドキュメントを再編して、読みやすくしました。

日付	メモ
2009-05-22	『iPhone Streaming Media Guide for Web Developers（ウェブデベロッパのためのiPhoneストリーミングメディアガイド）』のタイトルを変更しました。メディアセグメンタのダウンロード用のURLを追加しました。
2009-03-15	iPhone向けの、HTTPによるオーディオとビデオのライブストリーミングについて解説した新規ドキュメント。



Apple Inc.
© 2011 Apple Inc.
All rights reserved.

本書の一部あるいは全部を Apple Inc. から書面による事前の許諾を得ることなく複写複製（コピー）することを禁じます。また、製品に付属のソフトウェアは同梱のソフトウェア使用許諾契約書に記載の条件のもとでお使いください。書類を個人で使用する場合に限り1台のコンピュータに保管すること、またその書類にアップルの著作権表示が含まれる限り、個人的な利用を目的に書類を複製することを認めます。

Apple ロゴは、米国その他の国で登録された Apple Inc. の商標です。

キーボードから入力可能な Apple ロゴについても、これを Apple Inc. からの書面による事前の許諾なしに商業的な目的で使用すると、連邦および州の商標法および不正競争防止法違反となる場合があります。

本書に記載されているテクノロジーに関しては、明示または黙示を問わず、使用を許諾しません。本書に記載されているテクノロジーに関するすべての知的財産権は、Apple Inc. が保有しています。本書は、Apple ブランドのコンピュータ用のアプリケーション開発に使用を限定します。

本書には正確な情報を記載するように努めました。ただし、誤植や制作上の誤記がないことを保証するものではありません。

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
U.S.A.

アップルジャパン株式会社
〒163-1450 東京都新宿区西新宿
3丁目20番2号
東京オペラシティタワー
<http://www.apple.com/jp/>

App Store is a service mark of Apple Inc.

Apple, the Apple logo, Apple TV, Final Cut, Final Cut Studio, iPhone, iPod, iPod touch, Mac, Mac OS, QuickTime, Safari, and Xserve are trademarks of Apple Inc., registered in the United States and other countries.

iPad is a trademark of Apple Inc.

DEC is a trademark of Digital Equipment Corporation.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Intel and Intel Core are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Apple Inc. は本書の内容を確認しておりますが、本書に関して、明示的であるか黙示的であるかを問わず、その品質、正確さ、市場性、または特定の目的に対する適合性に関して何らかの保証または表明を

行うものではありません。その結果、本書は「現状有姿のまま」提供され、本書の品質または正確さに関連して発生するすべての損害は、購入者であるお客様が負うものとしします。

いかなる場合も、Apple Inc. は、本書の内容に含まれる瑕疵または不正確さによって生じる直接的、間接的、特殊的、偶発的、または結果的損害に対する賠償請求には一切応じません。そのような損害の可能性があらかじめ指摘されている場合においても同様です。

上記の損害に対する保証および救済は、口頭や書面によるか、または明示的や黙示的であるかを問わず、唯一のものであり、その他一切の保証にかわるものです。Apple Inc. の販売店、代理店、または従業員には、この保証に関する規定に何らかの変更、拡張、または追加を加える権限は与えられていません。

一部の国や地域では、黙示あるいは偶発的または結果的損害に対する賠償の免責または制限が認められていないため、上記の制限や免責がお客様に適用されない場合があります。この保証はお客様に特定の法的権利を与え、地域によってはその他の権利がお客様に与えられる場合もあります。