

#####

第一回 ドキュメント：ストリーミング 新ハウツー

このドキュメントは VNC Media Player を使ってストリームし、コンバートし、ストリームを保存する方法を説明します。

VLC Media Player:

https://wiki.videolan.org/VLC_media_player/

注：この作業は進行中です。完全を期すためには、時代遅れになったが VLC のストリーミングのドキュメントは旧ストリーミング・ハウツーをご覧ください。

旧ストリーミング・ハウツー：

https://wiki.videolan.org/Documentation:Streaming_HowTo/

1. イントロダクション
2. コンバート
 2. 1 GUI を使ってコンバートする
 2. 2 コマンドラインを使う
 2. 2. 1 ウィンドウズ
 2. 2. 2 UNIX ベースシステム
 2. 2. 3 コンバートのコマンドラインの文字列の構造
 2. 2. 4 コマンドラインの例：オーディオファイルを MP3 にコンバートする
 2. 2. 5 他のコマンドラインのコマンドの例
 - 3 ストリーミング
 3. 1 GUI を使ってストリーミングをする
 3. 2 コマンドライン・インターフェースを使ってストリーミングする
4. 参照

1. イントロダクション

VLC Media Player は様々なメディア・ストリームを配信し、コンバートし、保存することができます。サポートされるフォーマットやコーデックについてのさらなる情報については、VideoLAN ストリーミング機能リストをご覧ください。

2. コンバート

コンバートはメディアファイルやストリームを処理し、それを様々なフォーマットやビットレートにコンバートします。VLC でコンバートするにはふたつの方法があります。つまり GUI (VLC のデフォルト・インターフェース) かコマンドラインで。コンバートという用語自体はあるメディア・フォーマットを別のものに変え、ファイルに保存することです (つまりストリーミングは関係しません)。

2. 1 GUI を使ってコンバートする

変換方法のひとつは GUI を使って行います。次の通りアクセスしてください：

1. VLC で、「メディア>>変換/保存>>Save...」(Ctrl+R)を開く。
2. コンバートしたいインプットファイルあるいはストリームを追加します。「変換/保存」ボタンをクリック。
3. 「コンバート」ウィンドウで (2.1.2 では出力欄に) 保存ファイル名を(フィル拡張子を含め、すな

わちウィンドウズでは「C:\out.mp3」)指定します。

4. プロファイル・リストから望むコーデックを選択します。所与のプロファイルの設定をチェックし編集するため、「選択したプロファイルを編集する」ボタンをクリックします。プロファイルは追加またはさくができます。

5. 「Start」をクリックします。

プロファイル設定が互換性があり、コンバートが成功すれば、再生可能な出力ファイルが作られます。

Basic interface convert.png

2. 2 コマンドラインを使う

コンバートの別の方法はコマンドラインでできます。VLC で利用できるモジュールとオプションのすべての使用を可能にできるため、GUI より柔軟です。

VLC のコマンドラインを利用するためには、例えばシェルを通じてコマンドライン・インターンフェースにアクセスする必要があります。

2. 2. 1 ウィンドウズ

ウィンドウズでもコマンド・プロンプトは使えます。これはスタートメニューの「アクセサリ」フォルダーにあります。別の方法としては、「スタート>>実行」をするか、「Windows キー+R」を押して「cmd」をテキストフィールドに入力し「OK」またはエンターキーを押します。

コマンドライン・プロンプトを開いたら、VLC のあるフォルダー/ディレクトリに移動するのが、「vlc.exe」ないしは(.exe 部分が仮定されている場合)「vlc」とだけコマンドラインで実行するため最も簡単な方法です。VLC がデフォルトの場所(32bit システムの場合)にインストールされているなら、次の通り実行します：

```
cd "C:\Program Files\VideoLAN\VLC"  
訳注：64bit の場合  
cd "C:\Program Files(x86)\VideoLan\VLC"
```

必要なら vlc.exe のあるディレクトリに移動してください。

2. 2. 2 UNIX ベースシステム

Linux や Mac OS X のような Unix ベースのシステムでは端末ウィンドウを開くことができます。「vlc」を起動するのに必要なのは VLC を開くだけです：

```
../VLC.app/Contents/MacOS/VLC
```

2. 2. 3 コンバートのコマンドラインの文字列の構造

ファイルを他のフォーマットにコンバートするための典型的なコマンドラインの文字列は次のとおりです：

```
vlc --sout "#transcode{[TRANSCODE_OPTIONS]}:std{[OUTPUT_OPTIONS]}" INPUT
```

"INPUT"はインプット・ストリーム、[TRANSCODE_OPTIONS]はアウトプットしたいフォー

マット、コーデック、ビットレートなどのためのオプション、[OUTPUT_OPTIONS]はアプトプットのタイプのオプション、--sout はストリーム・アウトプットのコマンドライン・オプションです。

2. 2. 4 コマンドラインの例：オーディオファイルを MP3 にコンバートする

オーディオファイルの MP3 ファイルへのコンバートする典型的な使用法：

```
vlc --sout
"#transcode{acodec=mp3,ab=128,channels=2,samplerate=44100}:std{access=file,mux=raw,dst=OUTPUT}" INPUT
```

"INPUT"はインプットファイルで"OUTPUT"は出力ファイルです(例えば、ウィンドウズでは"C:\out.mp3"あるいは Unix ベースシステムでは"/home/username/out.mp3)。上の"sout"文字列は GUI でアクセス可能な「Audio - MP3」のデフォルトの設定です。この例ではオーディオインプットは 123kbps CBR MP3 ファイル(サンプルレート 44.1KHz、2 チャンネル)となります

2. 2. 5 他のコマンドラインのコマンドの例

オーディオ・インプットを WAVE(*.wav)ファイルに圧縮を戻す。

```
vlc --sout
"#transcode{acodec=s16l,channels=2,samplerate=44100}:std{access=file,mux=wav,dst=OUTPUT}" INPUT
```

オーディオ・インプットを AAC(MP4 形式)にコンバートする：

```
vlc --sout
"#transcode{acodec=mp4a,ab=128,channels=2,samplerate=44100}:std{access=file,mux=mp4,dst=OUTPUT}" INPUT
```

3 ストリーミング

3. 1 GUI を使ってストリーミングをする

GUI でのストリーミングは QT インターフェース上でのみ利用できます。GUI でストリームするにあ、VLC のメニューのメディア >> ストリームを開きます。：

ポップアップする「メディアを開く」で、(ファイル、ネットワーク・ストリーム、ディスク、キャプチャー・デバイスなど) ストリームを追加します。

項目を選択後、ウィザードの開始が準備されます。

次のストリーミング方法が VLC で利用可能です：

- ・ローカルで表示：ローカルのスクリーンでストリームを表示します。これはストリーミング中のストリームを表示させます。コンバートやリサイズの効果がこの機能でローカルでモニターできます。
- ・ファイル：ストリームをファイルに保存します。
- ・HTTP：HTTP ストリーミング方法を使います。どれを視聴するか TCP ポートを指定します。
- ・MS-WMSP(MMSSH)：このアクセス法は、マイクロソフト・ウィンドウズ・メディア・プレーヤーにストリームのために使います。どれを視聴するか IP アドレスと TCP ポート番号を指定します。注：

これは ASF カプセル化メソッドを使っただけのみ有効です。

- ・ UDP : 0.0.0.0~223.255.255.255 の範囲で提供されるユニキャスト、あるいは 244.0.0.0~239.255.255.255 の範囲で提供されるマルチキャストでストリームします。IPv6 アドレスにストリームすることも可能です。注：これは TS カプセル化メソッドでのみ有効です。
- ・ RTP:リアルタイム転送プロトコルを使用します。UDP のように、ユニキャストとマルチキャストのアドレスを利用できます。
- ・ IceCast : IceCast サーバーにストリームします。ストリームするため IceCast のアドレス、ポート、マウント・ポイント、認証を選択します。

一番普通の方法は HTTP です。HTTP を通じてストリームするためには、「次」の「ストリーム出力画面」で「新しい出力先」をドロップダウンから HTTP を選び、「追加」をクリックします。ビデオのストリーミングする場合、「/go.mpg.」のようにストリームを望むでしょう。ストリームが成功したら、クライアントとして VLC の別のインスタンスを使い、そのストリームを視聴できます。メニューの「メディア」>> 「ストリームを開く」>> 「http://localhost:8080/go/mpg」で、視聴できます。

いくつかの警告:

- ・ 「コンバートを可能にする」をクリックし、他のタイプを指定しても、".mp"のような拡張子を使わないなら、FLV 形式になります。

参照 : <http://forum.videolan.org/viewtopic.php?f=14&t=80722>

- ・ たとえそのスピードが適切でも(そして他のプレーヤーが受信できても)、VLC クライアントは 5fps 以下のストリームを受信できません。

参考 : <http://trac.videolan.org/vlc/ticket/214>

- ・ 注意！ストリームするため GUI を使うのは「多く」の場合、誤った選択になるでしょう。例えば、"/"のパスを使うと、それは (意図しない、そして非明示的にコンバート設定のファイル形式に上書きする) FLV 形式を使用することになります。そして"/go.mp4"とすると、非明示的に失敗します(エラーが何か見るとメッセージログを見ることができます。それは"mp4 type is not supported except for files"と表示されます)。そうセットアップするのはトリッキーです。これはほとんどの場合 GUI が間違っていて設定するようです。

ここに問題のリストがあります :

<https://forum.videolan.org/viewtopic.php?f=14&t=80722>

MPEG、TS、MPEG PS、MPEG 1、OGG、Raw、ASF、AVI、MP4、MOV などシステムのコーデックとアクセス法にあうコンバート・プロファイルを選びます。選択のコンボボックスの次の「選んだプロファイルを編集する」をクリックしてプロファイルを新たに作るか変更できます。

ストリームをアナウンスする方法を選択します。SAP(Service Announce Protocol)か SLP(Service Location Protocol)を使うことができます。またチャンネル名前を指定しなくてはなりません。Mac OS X インターフェイスでは、VLC の内部の HTTP か RTSP を使うか、ファイルとして、RTP セッションの記述ファイル (SDP) をエクスポートできます。これはアコーディオン・チェックボックスを使ってできます。SDP URL テキスト・ボックスで SDP ファイルが利用できる URL あるいは出力先を記入することができます。ストリーム・アウトプット MRL (Media Resources Locator) を表示するテキストボックスがあります。これはストリーム・アウトプット・ダイアログのオプションを変更すると更新され、必要なら、手動で調節できます。

「ストリーム」をクリックして、ストリーミングを開始する。

3. 2 コマンドライン・インターフェースを使ってストリーミングする

次も参照: [Documentation:Streaming_HowTo/Command_Line_Examples](https://wiki.videolan.org/Documentation:Streaming_HowTo/Command_Line_Examples/)

https://wiki.videolan.org/Documentation:Streaming_HowTo/Command_Line_Examples/

GUI の代わりにコマンドライン・インターフェースを使ってストリーミングする場合（さらにコントロールする。例えば、自分の形式を指定することができる）、GUI の場合、ロガーを開いたり（ツール・メニュー>>メッセージ）、冗長度 2 に設定してストリーミングするようなオプションを簡単に使うことができます。

デバッグメッセージの出力は基本的にコマンドライン・パラメーターのためのものです。そしてその中のパラメーターを読みコマンドラインで使うことができます(その場合パラメーターを引用符で囲んだり、"--"を加える必要があるかもしれません)。

ユニークなコマンドラインの例：

```
C:\Program Files\VideoLAN\VLC> vlc "C:\Users\Public\Videos\Sample Videos\Wildlife.wmv":  
sout=#transcode{vcodec=h264,acodec=mp3,ab=128,channels=2,samplerate=44100}:http{dst=:8090/go.mpg}
```

4. 参照

次も参照のこと：

Streaming HowTo - Outdated HowTo for streaming with VLC

[https://wiki.videolan.org/Documentation:Streaming_HowTo/](https://wiki.videolan.org/Documentation:Streaming_HowTo/Transcode)

Transcode - Further information about transcoding with VLC

<https://wiki.videolan.org/Transcode/>

VideoLAN documentation site, chapter 4

<http://www.videolan.org/doc/vlc-user-guide/en/ch04.html>

第二回 ストリーミング・ハウツー（1）イントロダクション

Documentation:Streaming HowTo
https://wiki.videolan.org/Documentation:Streaming_HowTo/

著者:
Alexis de Lattre
Johan Bilien
Anil Daoud
Clément Stenac
Antoine Cellier
Jean-Paul Saman

Copyright (c) 2002-2006 the VideoLAN project

Free Software Foundation が発行した GNU General Public License（あるいは License 第2版、それ以降）の定義の下、このドキュメントを複製し、配布し、改変する許可が与えられています。

このドキュメントは VideoLan を使ってストリームを配信し、変換し、保存する方法を説明します。

注：ストリーミングについての新しい情報はストリーミング・新ハウツーをご覧ください（WIP） *
訳注：上に訳出

- 目次
- 1) イントロダクション
 - 2) メイン
 - 3) VLM
 - 4) チュートリアルと例
 - 5) IPv6

1) イントロダクション
Streaming, Muxers and Codecs
https://wiki.videolan.org/Documentation:Streaming_HowTo/Streaming,_Muxers_and_Codecs/

- 目次
- 1. イントロダクション
 - 1. 1. 概要
 - 1. 2. VideoLan ソフトウェア
 - 1. 2. 1. VLC Media Player
 - 2. マルチプレクサとコーデック
 - 2. 1. コーデックとはなにか？
 - 2. 2. フォーマットとはなにか？
 - 2. 3. ビデオをエンコードする
 - 2. 4. ビデオを再生する

- 1. イントロダクション
 - 1. 1. 概要

VideoLan は、GNU General Public License の下、エコール・セントラル・パリの学生と世界中の開

発者によって開発されたビデオ・ストリーミングの完全なソリューションです。VideoLan は広帯域ネットワークで MPEG ビデオをストリームするよう作られています。

VideoLan は次のものを含みます :

- ・ V L S : MPEG-1、MPEG-2、MPEG-4、DVD、デジタル衛星チャンネル、地上波デジタルテレビ・チャンネル、ユニキャストまたはマルチキャストのネットワーク上のビデオをストリーミングする
- ・ V L C (元は VideoLan Client) : MPEG-1、MPEG-2、MPEG-4 ファイル、DVD、ユニキャストまたはマルチキャストのネットワークのライブ・ビデオのサーバーとしてもいることができます : また複数の OS で MPEG ストリームを受信し、デコードし、表示するクライアントとして使用します。

次が完全な VideoLAN ソリューションのイラストです :

Global VideoLAN solution

プロジェクトの詳細については VideoLan ウェブサイトで調べることができます :

<http://www.videolan.org/>

1. 2. VideoLan ソフトウェア

1. 2. 1. VLC Media Player

VLC は多くのプラットフォームで稼働します : Linux, Windows, Mac OS X, BeOS, *BSD, Solaris, Familiar Linux, Yopy/Linupy and QNX. VLC は次を再生できます :

- ・ ハードディスク、CD-ROM ドライブ上などの MPEG-1、MPEG-2、MPEG-4、DivX ファイル
- ・ DVD と VCD
- ・ 衛星カードから(DVB-S)
- ・ デジタルビデオカメラから(DV)
- ・ VLS または VLC でネットワークに送られた MPEG-1、MPEG-2、MPEG-4

VLC はストリーミングのためのサーバーとしても使えます :

- ・ MPEG-1、MPEG-2、MPEG-4、DivX ファイル
- ・ DVD
- ・ MPEG エンコーディング・カードから
- ・ デジタルカメラから

IPv4 または IPv6 で次へ :

- ・ 一台のマシンへ(つまりひとつの IP アドレスに) : これはユニキャストと呼ばれる
- ・ 動的なマシンのグループへ(つまりマルチキャスト IP アドレスに) : これはマルチキャストと呼ばれる

サポートされるそれぞれの VLC の可能性の完全なリストを得るには、VLC 機能ページを参照 :

<http://www.videolan.org/vlc/features.html>

注 : VLC は Mac OS 9 では動きません。

2. マルチプレクサとコーデック

2. 1. コーデックとはなにか?

VideoLan を完全に理解するには、コーデックとコンテナ・フォーマットとの違いを理解しなければ

なりません。

コーデックは圧縮アルゴリズムで、ストリームのサイズを小さくします。オーディオ・コーデックとビデオ・コーデックがあります。MPEG-1、MPEG-2、MPEG-4、Vorbis、DivXなどがコーデックです。

2. 2. コンテナ・フォーマットとはなにか

始めるにあたって、コンテナフォーマットを普通の輸送の箱と考えてみてください。箱が郵送されます。「中に何があるのか」と考えるでしょう。箱には注意を払わないでしょう。箱の中身にしか興味はない。何か問題でも？箱の中は見えません、。どうしたいのか。ナイフで切って開けないといけません。

コンテナ・フォーマットはこのような基本的アイデアに従います。それには、コーデックによってエンコード済みのひとつ以上のストリームが入っています。オーディオストリームやビデオストリームがあります。AVI、Ogg、MOV、ASF、MP4などがコンテナフォーマットです。ストリームは様々なコーデックでエンコードされます。完璧な世界では、どんなコンテナフォーマットにどんなコーデックでも入れられます。不幸にもいくつか互換性がないことがあります。次の機能のページで可能なコーデックとコンテナ・フォーマットの組み合わせを知ることができます。

機能のページ：

<http://www.videolan.org/vlc/features.html>

2. 3. ビデオをエンコードする

これは郵送箱を作る最初の段階です。それはオーディオでもビデオでも、ファイルは以前のフォーマットより物理的量の小さい別のフォーマットに圧縮することを意味します。よく使われるビデオ・エンコード法は、DivX、MPEG-1、MPEG-2、MPEG-4などがあり、最も使われるエンコード法はMP3かogg-vorvisです。

その後、マルチプレクスしなければなりません。これは基本的に分離したビデオ（あるいはストリーム）をひとつファイルに結合する処理です。

訳注：マルチプレクサ、デマルチプレクサは電子回路の専門用語、単純に言うと、信号の統合と分解。

2. 4. ビデオを再生する

箱があったとして、コンテンツを見る前に、箱を開ける必要があります。それがまさにVLCが行うことです。ストリームをデコードするため、VLCはデマルチプレクス（マルチプレクスされたものを分解する）をします。これはコンテナ・フォーマットを読み、オーディオ、ビデオ、字幕に分離します。デマルチプレクスはビデオもオーディオも質を落とさず、ストリームのデータになにもせず、単に別のファイルに分けるだけで、それぞれはオリジナルのファイルの部分が入っています。その後これらそれぞれはストリームを解凍する数学的処理をするデコーダーに渡されます。

ここでMPEGについて考えます：

- ・MPEGはコーデックです。MPEGには、MPEG-1、MPEG-2、MPEG-4といういくつかのバージョンがあります。

- ・MPEGはまたコンテナフォーマットです。しばしばMPEGシステムのことをいいます。

MPEGにはいくつかのタイプがあります：ES、PS、TS。

たとえばDVDからMPEGビデオを再生するとき、MPEGストリームは（エレメンタリー・システム、ESと呼ばれる）いくつかのシステムから実際構成されます：ビデオ、オーディオ、字幕などのストリームがあります。これらの異なるストリームがシングル・プログラム・システム（PS）にミックス

されます。しかしこのPSフォーマットは例えばネットワークや衛星通信ではビデオストリームは採用されていません。そのため、トランスポート・ストリーム(TS)はそのようなチャンネルを通じたMPEGビデオのため設計されています。

第三回 ストリーミング・ハウツー（2）メイン、VLM、チュートリアルと例、IPv6

2) メイン
・簡易ストリーミング

https://wiki.videolan.org/Documentation:Streaming_HowTo/Easy_Streaming/

このページは時代遅れです。

Documentation:Streaming HowTo New#Streaming using the GUI for updated streaming tutorials.

https://wiki.videolan.org/Documentation:Streaming_HowTo_New/#Streaming_using_the_GUI
をご覧ください。

(以下略)

新版 VLC での簡易ストリーム(未完成)

https://wiki.videolan.org/Documentation:Streaming_HowTo/Easy_Streaming_Newer_Versions/

* 訳注 :

このページの解説は、現行の VLC 2.* でもほぼ同じですが、GUI ページがことなるため、VLC Media Player 新バージョン 2.1.2 で、書き直しました。次を見てください

VLC Media Player 2.1.2 GUI での簡易ストリーム

<http://lunedì.sblo.jp/article/86061309.html>

(以下略)

3) VLM - マルチプル・ストリーミングとビデオ・オン・デマンド(Multiple Streaming and Video on Demand)

https://wiki.videolan.org/Documentation:Streaming_HowTo/VLM/

* 訳注 : VLM とは VideoLan Manager の頭文字です。

インターネットラジオのようなプログラムを組むための方法なので省略します

4) チュートリアルと例

ストリームの受信と保存

https://wiki.videolan.org/Documentation:Streaming_HowTo/Receive_and_Save_a_Stream/

目次

1. VLC でストリームを受信する
 1. 1. ユニキャスト・ストリームを受信する
 1. 2. マルチキャスト・ストリームを受信する
 1. 3. HTP/FTP/MMS ストリームを受信する
 1. 4. RTSP を次通じて利用可能にした RTP ストリームを受信する
 1. 5. SDP ファイルで記述されたストリームを受信する
 2. VLC でストリームを保存する
 3. セットトップボックスでストリームを受信する
-
1. VLC でストリームを受信する
 1. 1. ユニキャスト・ストリームを受信する

```
% vlc -vvv rtp : //
```

1. 2. マルチキャスト・ストリームを受信する

```
% vlc -vvv rtp : //@239.255.12.42
```

239.255.12.42 が受信したいマルチキャスト IP アドレスにあたります;

1. 3. HTTP/FTP/MMS ストリームを受信する

次のようにコマンドラインで実行します :

```
% vlc -vvv http : //example/stream.xyz
```

http : //example/stream.xyz がストリームの HTTP アドレスにあたります ;

```
% vlc -vvv ftp : //example/stream.xyz
```

ftp : //example/stream.xyz がストリームの FTP アドレスにあたります;

```
% vlc -vvv mms : //viptvr.yacast.fr/encoderfranceinfo
```

mms : //viptvr.yacast.fr/encoderfranceinfo がストリームの MMS アドレスにあたります;

1. 4. RTSP を次通じて利用可能にした RTP ストリームを受信する

```
% vlc -vvv rtsp : //www.hardradio.com/tonbeme.mov
```

rtsp : //www.hardradio.com/tonbeme.mov がストリームのアドレスにあたります。

1. 5. SDP ファイルで記述されたストリームを受信する

```
% vlc -vvv http : //server.example.org/stream.sdp
```

2. VLC でストリームを保存する

VLC はディスクにストリームを保存できます。このため、VLC ストリーム出力を使います : グラフィカル・インターフェース (メニュー : メディア→ストリーミング) またはレコードボタンを使うか、コマンドラインに次の引数を加えることでできます :

* 訳注 : 現行版 GUI ではストリーム表示を右クリック→保存

```
--sout file/muxer:stream.xyz
```

・ muxer (マルチプレクサ) は VLC ストリーム出力がサポートするフォーマットのひとつを示します。すなわち :

- ・ ogg : OGG format,
- ・ ps : MPEG2-PS format,
- ・ ts : MPEG2-TS format.
- ・ stream.xyz はストリーム保存したいファイル名です。正しい拡張子が必要です。

例:

```
vlc your_input_file_or_stream_here --sout=file/ps:go.mpg
```

これはより冗長なコマンドを省略したものです

```
vlc your_input_file_or_stream_here --sout="#std{access=file,mux=ps,dst=go.mpg}"
```

注意：ストリームのファイルタイプをサポートするマルチプレクサでなければなりません。
Transcode#Compatibility_issues を参照。
https://wiki.videolan.org/Transcode/#Compatibility_issues

レコードボタンを使いレコードするとき VLC が使う設定を見るのが便利です。例えば、ログで次のような記述を見つけることができかもしれません：

```
...: Using record output `std{access=file,mux='ps',dst='C:\vlc-record-2010__E-.mpg}'`
```

これは現在のストリームをレコードする方法のヒントになります。この場合、コマンドラインで「--sout"#std{access=file...」に翻訳できるでしょう。

3. セットトップボックスでストリームを受信する

(略) * 訳注：PC 以外の機器の使用

5) IPv6

(略)

第四回 ストリーミング・ハウツー（3）コマンドラインの例
Documentation:Streaming HowTo/Command Line Examples
コマンドラインの例
https://wiki.videolan.org/Documentation:Streaming_HowTo/Command_Line_Examples/

VLC ストリーム出力の上級使用例(変換、マルチストリーミングなど)

目次

1. 変換
 1. 1. より複雑な変換の例
 2. マルチ・ストリーミング
 3. 変換とマルチ・ストリーミング
 3. 1. より複雑なマルチ-変換の例
 4. HTTP ストリーミング
 5. RTSP ライブ・ストリーミング
 6. RTSP オンデマンド-ストリーミング
 7. ウィンドウズ・メディア・プレーヤーへの MMS/MMSH ストリーミング
 8. ES モジュールの利用
 9. ストリームをオープンのままにする
 10. VLC をリフレクターとして使用する

1. 変換

128kbps、44100HZ で2チャンネルに Ogg Vorbis ヘストリームを変換し、"foobar.ogg"として保存する：

```
% vlc -I dummy -vvv input_stream --sout  
"#transcode{vcodec=none,acodec=vorb,ab=128,channels=2,samplerate=44100}:file{dst=foobar.ogg}"
```

入力ストリームを変換し、関連する SAP アナウンスと共にマルチキャスト IP アドレスに送信する：

```
% vlc -vvv input_stream --sout  
'#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:  
rtp{mux=ts,dst=239.255.12.42,sdp=sap,name="TestStream"}'
```

インプットストリームを表示し、関連する SAP アナウンスと共にマルチキャスト IP アドレスに送信する：

```
% vlc -vvv input_stream --sout  
#duplicate{dst=display,dst="transcode{vcodec=mp4v,acodec=mpga,vb=800,  
ab=128,deinterlace}:rtp{mux=ts,dst=239.255.12.42,sdp=sap,name="TestStream"}"}
```

入力ストリームを変換し、変換ストリームを表示し、関連する SAP と共にマルチキャスト IP アドレスに送信する：

```
% vlc -vvv input_stream --sout  
'#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:  
duplicate{dst=display,dst=rtp{mux=ts,dst=239.255.12.42,sdp=sap,name="TestStream"}}'
```

クライアントで上記のマルチキャストされた入力ストリームを受信する：

```
vlc rtp : //239.255.12.42
```

1. 1. より複雑な変換の例

SDI カードでキャプチャーした入力を udp で H.264 と AAC の TS でストリーミングする

```
% cvlc -vvv --live-caching 2000 decklink : //  
--decklink-audio-connection embedded --decklink-aspect-ratio 16:9 --decklink-mode hp50  
--sout-x264-preset slow --sout-x264-tune film --sout-transcode-threads 8 --no-sout-x264-interlaced  
--sout-x264-keyint 50 --sout-x264-lookahead 100 --sout-x264-vbv-maxrate 6000 --sout-x264-vbv-  
bufsize 6000  
--sout  
'#transcode{vcodec=h264,vb=6000,acodec=mp4a,aenc=fdkaac,ab=256}:std{access=udp,mux=ts,ds  
t=192.168.2.1:1234}'
```

2. マルチストリーミング

ストリームをマルチキャスト IP アドレスとユニキャスト IP アドレスに送信する :

```
vlc -vvv input_stream  
--sout  
'#duplicate{dst=rtp{mux=ts,dst=239.255.12.42,sdp=sap,name="TestStream"},dst=rtp{mux=ts,dst=  
192.168.1.2}}'
```

ストリームを表示し、2つのユニキャスト IP アドレスに送信する :

```
vlc -vvv input_stream  
--sout  
'#duplicate{dst=display,dst=rtp{mux=ts,dst=192.168.1.12},dst=rtp{mux=ts,dst=192.168.1.42}}'
```

マルチ・プログラムのインプット・ストリームの一部を送信する :

```
vlc -vvv multiple_program_input_stream  
--  
sout'#duplicate{dst=rtp{mux=ts,dst=239.255.12.42},select="program=12345",dst=rtp{mux=ts,dst=  
239.255.12.43},select="video,program=1234-2345"}'
```

このコマンドは ID が 1 2 3 4 5 の入力ストリームのプログラムを 239.255.12.42 に送信し、ID が 1 2 3 4 から 2 3 4 5 のすべてのプログラムを 239.255.12.43 に送信する。

3. 変換とマルチ・ストリーミング

インプットストリームを変換し、変換されたストリームを表示し、関連する SAP アナウンスと共にマルチキャスト IP アドレスとユニキャスト IP アドレスに送信する :

```
% vlc -vvv input_stream --sout  
'#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:  
duplicate{dst=display,dst=rtp{mux=ts,dst=239.255.12.42,sdp=sap,name="TestStream"},  
dst=rtp{mux=ts,dst=192.168.1.2}}'
```

入力ストリームを表示し、変換し、2つのユニキャスト IP アドレスに送信する :

```
% vlc -vvv input_stream --sout
'#duplicate{dst=display,dst="transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:
duplicate{dst=rtp{mux=ts,dst=192.168.1.2},dst=rtp{mux=ts,dst=192.168.1.12}}'
```

入カストリームをマルチキャスト IP アドレスに送信し、変換したストリームを関連する SAP アナウンスと共に別のマルチキャスト IP アドレスに送信する：

```
% vlc -vvv input_stream --sout
'#duplicate{dst=rtp{mux=ts,dst=239.255.1.2,sdp=sap,name="OriginalStream"},
dst="transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:
rtp{mux=ts,dst=239.255.1.3,sdp=sap,name="TranscodedStream"}}'
```

3. 1. より複雑なマルチ変換の例

SDI 入力カードでキャプチャーした入力をひとつは HD、もうひとつは SD の 2 つに変換し、両方を udp に送信する。

```
% cvlc -vv --live-caching 2000
--decklink-audio-connection embedded --decklink-aspect-ratio 16:9 --decklink-mode hp50
decklink : //
--sout-x264-preset fast --sout-x264-tune film --sout-transcode-threads 24 --no-sout-x264-interlaced
--sout-x264-keyint 50 --sout-x264-lookahead 100 --sout-x264-vbv-maxrate 4000 --sout-x264-vbv-
bufsize 4000
--sout
'#duplicate{dst="transcode{vcodec=h264,vb=6000,acodec=mp4a,aenc=fdkaac,ab=256}:std{access
=udp,mux=ts,dst=192.168.1.2:4013}",
dst="transcode{height=576,vcodec=h264,vb=2000,acodec=mp4a,aenc=fdkaac,ab=128}:std{access
=udp,mux=ts,dst=192.168.1.2:4014}}'
```

SDI 入力カードでキャプチャーした入力を、無変換で一度再ストリーミングし、二度目は変換する：

```
% cvlc -vv --live-caching 2000
--decklink-audio-connection embedded --decklink-aspect-ratio 16:9 --decklink-mode hp50
decklink : //
--sout-x264-preset fast --sout-x264-tune film --sout-transcode-threads 24 --no-sout-x264-interlaced
--sout-x264-keyint 50 --sout-x264-lookahead 100 --sout-x264-vbv-maxrate 4000 --sout-x264-vbv-
bufsize 4000
--sout
'#duplicate{dst="transcode{vcodec=h264,vb=6000,acodec=mp4a,aenc=fdkaac,ab=256}:std{access
=udp,mux=ts,dst=192.168.1.2:4013}",
dst="std{access=udp,mux=ts,dst=192.168.1.2:4014}}'
```

4. HTTP ストリーミング

HTTP でストリームする：

サーバー側で次を実行する：

```
% vlc -vvv input_stream --sout '#standard{access=http,mux=ogg,dst=server.example.org:8080}'
```

クライアント側で次を実行する：

```
% vlc http://server.example.org:8080
```

変換し、HTTP でストリームする :

```
% vlc -vvv input_stream --sout '#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128}:standard{access=http,mux=ogg,dst=server.example.org:8080}'
```

ライブ・ビデオ・ストリームを録画する :

```
% vlc http://example.com/live.asf --sout="#duplicate{dst=std{access=file,mux=asf,dst='C:\test\test.asf'},dst=nodisplay}"
```

例えば HTTP で Ogg/Vorbis でオーディオをストリームしたい場合 :

```
% vlc -vvv cdda:/dev/cdrom
--sout
'#transcode{acodec=vorb,ab=128}:standard{access=http,mux=ogg,dst=server.example.org:8080}'
```

5. RTSP ライブ・ストリーミング

RTSP と RTP でストリームする :

サーバー側次を実行する :

```
vlc -vvv input_stream --sout
'#rtp{dst=192.168.0.12,port=1234,sdp=rtsp://server.example.org:8080/test.sdp}'
```

クライアント側で次を実行する:

```
vlc rtsp://server.example.org:8080/test.sdp
```

6. RTSP オンデマンド・ストリーミング

Documentation:Streaming HowTo/VLM を見よ。

https://wiki.videolan.org/Documentation:Streaming_HowTo/VLM/

7. ウィンドウズ・メディア・プレーヤーへの MMS/MMSH ストリーミング

```
% vlc -vvv input_stream --sout '#transcode{vcodec=DIV3,vb=256,scale=1,acodec=mp3,ab=32,channels=2}:std{access=mms, mux=asfh,dst=:8080}'
```

VLC メディア・プレーヤーは次の URL を使ってこの接続ができます :

mms://server_ip_address:8080。ウィンドウズ・メディア・プレーヤーは次の URL を使ってこの接続ができます : mms://server_ip_address:8080。

8. ES モジュールの利用

訳注 : ES=elementary Stream、要素ストリームと訳したもの

2つの PS ファイルでオーディオとビデオを分離します :

```
% vlc -vvv input_stream --sout '#es{access=file,mux=ps,url_audio=audio-%c.%m,url_video=video-
```



```
%c.%m}'
```

TS ファイルに入カストリームのオーディオトラックを抽出します :

```
% vlc -vvv input_stream --sout '#es{access_audio=file,mux_audio=ts,url_audio=audio-%c.%m}'
```

ひとつオーディオトラックとビデオトラックを別々のポートにユニキャストでストリームする(注 : これは VLC0.8.6 以前でしか有効ではありません。改善お願い) :

Stream in unicast the audio track on a port and the video track on another port (NOTE: This will not only work with VLC 0.8.6 or older - FIXME?):

サーバー側:

```
% vlc -vvv input_stream --sout '#es{access=rtp,mux=ts,url_audio=192.168.1.2:1212, url_video=192.168.1.2:1213}'
```

クライアント側 :

オーディオ受信 :

```
% vlc udp : //@:1212
```

ビデオ受信 :

```
% vlc udp : //@:1213
```

ビデオをマルチキャストし、オーディオをファイルにダンプする :

```
% vlc -vvv input_stream --sout '#es{access-video=udp,mux-video=ts,dst-video=239.255.12.42, access-audio=file,mux-audio=ps,dst-audio=audio-%c.%m}'
```

注 : 更に複雑な手法を構築するため ES モジュールと他のモジュールを結合することもできます。

9. ストリームをオープンのままにする

```
% % vlc -vvv input_stream -sout-keep  
-sout=#transcode{acodec=mp3}:duplicate{dst=display{delay=6000},  
dst=gather:std{mux=mpeg1,dst=:8080/stream.mp3,access=http},select="novideo"}
```

基本的な変換は選択したファイルから MP3 ストリームにです (ビデオ・ファイルの場合は、ビデオは無視されます)。HTTP を通じて localhost:8080/stream.mp3 似ストリームされます。
:sout-keep と d s t=gather:std の結合は、ストリームがオープンのままにされ、継続するアイテムが同じストリームを通じて再生されます。

10. VLC をリフレクターとして使用する

udp 入力を受け、一旦 ipv6 マルチキャストを通じて無変換にレンダリングし、更に HLS にレンダリングします。

```
% cvlc -vvv udp : //@:4013 --ttl 60  
--sout '#duplicate{dst=std{access=http,mux=ts,dst=[::]:3013}",  
dst=std{access=udp,mux=ts,dst=ffe2::1:2013},
```

```
dst=std{ access=livehttp{ seglen=5,delsegs=true,numsegs=5,index=/path/to/stream.m3u8,  
index-url=http : //example.org/stream-#####.ts},mux=ts {use-key-frames},dst=/path/to/stream-  
#####.ts}}
```


第五回 ストリーミング・ハウツー（４）ファイルをストリームする

ファイルをストリームする

https://wiki.videolan.org/Documentation:Streaming_HowTo/Stream_a_File/

VLC でファイルをストリームする

```
% vlc -vvv video1.xyz --sout udp:192.168.0.42 --ttl 12
```

つまり:

- ・ video1.xyz はストリームをしたいファイルです
- ・ 192.168.0.42 は次のいずれかです:
 - ・ ユニキャストをしたいマシンの IP アドレス;
 - ・ ユニキャストしたいマシンの DNS ネーム;
 - ・ マルチキャスト IP アドレス
- ・ 12 は IP パケットの TTL(Time To Live)の値(これはストリームが 11 のルーターを超えることができるを意味します)。

ファイルを継続的にストリームする場合は、--loop オプションを加えます。

もちろん更にオプションを加えることができます（変換、TCP ポートへのストリーミングなど）が、まずこれから始めてください。

* 訳注：次は略します。単にオタクっぽいので。

Stream a DVD

Stream a DVB Channel

Stream from Encoding Cards and Other Capture Devices

Stream from a DV Camcorder

Streaming a live video feed to Darwin Streaming Server for Mobile Phones

Streaming for the iPhone with live http

コマンドラインを使用する上級ストリーミング

https://wiki.videolan.org/Documentation:Streaming_HowTo/Advanced_Streaming_Using_the_Command_Line/

目次

- 1. ストリーム出力の構造
- 2. モジュールの記述
 - 2. 1. 標準(alias std)
 - 2. 1. 1. access
 - 2. 1. 2. mux
 - 2. 1. 3. dst
 - 2. 1. 4. sap
 - 2. 1. 5. group
 - 2. 1. 6. sap-ipv6
 - 2. 1. 7. slp
 - 2. 1. 8. name
 - 2. 2. ディスプレイ
 - 2. 2. 1. display
 - 2. 2. 2. noaudio
 - 2. 2. 3. delay
 - 2. 3. rtp
 - 2. 3. 1. dst
 - 2. 3. 2. pot
 - 2. 3. 3. pot-video
 - 2. 3. 4. pot-audio
 - 2. 3. 5. sdp
 - 2. 3. 6. ttl
 - 2. 3. 7. mux
 - 2. 3. 8. rtcp-mux
 - 2. 3. 9. proto=
 - 2. 3. 10. name
 - 2. 3. 11. description
 - 2. 3. 12. url
 - 2. 3. 13. email
 - 2. 4. es
 - 2. 4. 1. access-video
 - 2. 4. 2. access-audio
 - 2. 4. 3. access
 - 2. 4. 4. mux-video
 - 2. 4. 5. mux-audio
 - 2. 4. 6. mux
 - 2. 4. 7. dst-video
 - 2. 4. 8. dst-audio
 - 2. 4. 9. dst
 - 2. 5. 変換
 - 2. 5. 1. vcodec
 - 2. 5. 2. vb
 - 2. 5. 3. venc

- 2. 5. 4. fps
- 2. 5. 5. deinterlace
- 2. 5. 6. croptop
- 2. 5. 7. cropbottom
- 2. 5. 8. cropleft
- 2. 5. 9. cropright
- 2. 5. 10. scale
- 2. 5. 11. width
- 2. 5. 12 height
- 2. 5. 13. acodec
- 2. 5. 14. ab
- 2. 5. 15. aenc
- 2. 5. 16. samplerate
- 2. 5. 17. channels
- 2. 5. 18. scodec
- 2. 5. 19. senc
- 2. 5. 20. soverlay
- 2. 5. 21. sfilter
- 2. 5. 22. threads
- 2. 5. 23. audio-sync
- 2. 5. 24. vfilter
- 2. 6. duplicate
- 2. 6. 1. dst
- 2. 6. 2. select
- 2. 7. その他
- 2. 8. 単純なシンタックス
- 3. サンプル

1. ストリーム出力の構造

ストリーム出力は VLC が読み込んだどんなストリームもそれを表示する代わりに、ファイルやネットワークストリームに出力できる VLC の機能の名前です。様々な処理がこのプロセスの間、ストリームに適用することができます（変換、リサイズ、フィルター、再マルチプレクサなど）。ストリーム出力は様々なモジュールがあり、それぞれが様々な能力をもっています。可能性を高めるため、モジュールのチェーンを作ることができます。

・標準 出力モジュールへのアクセスを通じてストリームを送ることができます：例えば、UDP、ファイル、HTTP など。チェーンの最後にこのモジュールを使うでしょう。

・変換 インพุットストリームのオーディオとビデオを変換（異なるコーデックやビットレートを使ってストリームをデコードしたり、再エンコード）するのに使います。インพุットやアウトプットのアクセス方法はペース・コントロール(ネットワーク、キャプチャー・デバイス)を認めていないなら、リアルタイムで「急いで」行われる。これはパラメーターのセットに依存して、多くの CPU のパワーを必要とする。ファイルやディスクのような他のストリームはシステムが許す限り早く変換されます。

・デュプリケート ストリームが独立の方法で扱われる第二のチェーンを作ることができるようにします。

・ディスプレイ VLC が通常行うようにインพุット・ストリームを表示させます。デュプリケート・モジュールと共に使うことで処理中のストリームをモニターすることができます。

・ rtp RTP(それぞれの要素ストリームにひとつずつポートを割り振る)でストリームします。このモジュールは RTSP をサポートします。

・ es インプット・ストリームから要素ストリーム(ES:Elementary Streams)を分離できるようにします。これは例えば、オーディオとビデオのストリームを別のファイルに保存するのに用いることができます。

- ・ bridge-out 執筆予定
- ・ bridge-in 執筆予定
- ・ mosaic-bridge 執筆予定

それぞれのモジュールにはオプションがあります。次のようなシンタックスに従わなくてはなりません：

```
% vlc input_stream --sout "#module1{option1=parameter1{parameter-option1},option2=parameter2}:module2{option1=...,option2=...}:..."
```

注：いくつかのモジュール・オプション(例の中の option1)は設定しなくてはなりません、他は任意です。

オプションのパラメータ(例の中の parameta-option1)は常に任意です。これらのオプション・パラメータもまたしばしば上級の設定です。それらの記述を理解できない場合は、おそらくそれは必要ではないことを意味します。

次のようなシンタックスを使うこともできます：

```
% vlc input_stream --sout-module1-option1=... --sout-module1-option2=... --sout-module2-option1=... --sout-module2-option2=... ...
```

例えば、ストリームを変換し、それを送信する場合は次のとおりです：

```
% vlc input_stream --sout '#transcode{options}:standard{options}'
```

2. モジュールの記述

現在利用可能なモジュールのリストです：

2. 1. スタンダード(alias std)

このモジュールは、マルチプレクサ(信号の複数の要素をカプセル化)したあと、ストリームをファイルに保存するか、ネットワークに送ります。

利用可能なオプション：

2. 1. 1. 標準(alias std) - access

このオプションはストリームを保存するか送信するのに用いられるメディアを設定します。これは必須のオプションです。利用可能なオプションは：

- ・ file : ストリームをファイルに保存する

append オプションは、ファイルを置き換える代わりに、既存のファイルにストリームを付加するのに

用います :

```
standard{ ... ,access=file{append}, ... }
```

- ・ udp: UDP ユニキャストまたはマルチキャスト・アドレスにストリームする

アイテムのオプションは :

- ・ caching=< time in ms> : 送信する前に VLC がデータをバッファする時間をセットします;
 - ・ ttl=< ttl> : 送信する udp パケットの ttl を設定します;
 - ・ group=< amount of packets> : ひとつひとつではなく一気にパケットを送信します;
 - ・ late=< time in ms> : このチェーンの段階で到着が遅すぎるパケットを落とします;
 - ・ raw : パケットを送信する前に MTU が満杯になるまで待てない場合;
- ・ http: streams over HTTP.

アイテムのオプション :

- ・ user=< user name> : HTTP ベーシック認証を可能にし、ユーザーを設定します;
 - ・ pwd=< password> : ベーシック認証のパスワードを設定します;
 - ・ mime=< mime type> : サーバー返す MIME タイプを設定します;
- ・ https: セキュアな SSL コネクションを使って HTTP でストリームします。

アイテムのオプションは HTTP と同じでそれと :

- ・ cert=< path to certificate> : 使用するための認証を設定します;
 - ・ key=< path to key> : サーバーが SSL 接続のため使うプライベートのキー・ファイルを設定します;
 - ・ ca=< path to certificate> : SSL のため使用するルートの CA 認証へのパスを設定します;
 - ・ crl=< path to certificate> : SSL 接続のため使用する取り消し認証を設定します。
- ・ mmsh: マイクロソフト MMS プロトコルを使用するストリーム。このプロトコルは多くのマイクロソフトのソフトウェアによる転送方法として用いられます。

MMS プロトコルの僅かな部分しかサポートされないことに注意(HTTP でカプセル化される MMS) 。アイテムのオプションは HTTP モジュールと同じです。

- ・ rtp: RTP によるストリーム。これはプレーンの RTP を通じて MPEG-TS をストリームする場合にだけ使うことができます。

このオプションのサポートは VLC0.9.0 以降削除されました。その代わりに rtp ストリームモジュールを使用しなくてはなりません。オプションは udo 設定と同じです。

2. 1. 2. 標準(alias std) - mux(マルチプレクサ)

このオプションで、ストリームするため使用されるカプセル化の方法を設定します。これは設定しなければなりません。

利用可能なオプションは次のとおりです :

- ・ ts: MPEG2/TS マルチプレクサ。これは MPEG-2 をストリームする時用いられる標準的なマルチプ

レクサです。このマルチプレクサはどのアクセス方法でも利用できます。サポートされるコーデックは、MPEG1/2/4、MJPEG、H264、WMV 1/2 & theora for video、MPEG audio、AAC & a52 for audio stream。

アイテムのオプションは：

- ・ pid-video=< pid> ビデオ・トラックのPIDを設定する;
- ・ pid-audio=< pid> オーディオトラックのPIDを設定する;
- ・ pid-spu=< pid> 字幕トラックのPIDを設定する;
- ・ pid-pmt=< pid> PMT(プログラム・マップ・テーブル)のPIDを設定する;
- ・ tsid=< id> 出力したTSストリームのIDを設定する;
- ・ shaping=< shaping delay in ms> 様々なビットレートのためストリームのビットレートを維持する最小のインターバルを設定する;
- ・ use-key-frames シェイピング・インターバルの制限としてIフレームを使用する
- ・ pcr=< PCR interval in ms> どのインターバルでプログラム・クロック・リファレンスを送信するか設定することができる;
- ・ dts-delay=< delay in ms> 所与の時点からDTSからPTS(プレゼンテーション・タイム・スタンプ)を遅らせることができます;
- ・ crypt-audio CSA アルゴリズムを使ってオーディオの暗号化を可能にすることができます;
- ・ csa-ck=< key as a 16 character word> CSA 暗号化のため使うキーを設定することができます。

・ ps: the MPEG2/PS マルチプレクサ。これはMPEG 2ファイル(.mpg)標準マルチプレクサです。これはファイルとHTTP出力方法を使用できます。サポートするコーデックはMPEG 1/2、MJPEG & MJPEG for VIDEO、MPEG audio & a52 for audio streams です。

使用できるオプションはつぎだけです：

・ dst-delay=< delay in ms> ; これは所与の時間のDTS (デコード・タイム・スタンプ)からPTS (プレゼンテーション・タイム・スタンプ)に遅らせることができます。

・ mpeg1: 標準的なMPEG 1マルチプレクサ。このマルチプレクサは、ファイルを保存するか、HTTPでストリーミングした時、MPEG 1ビデオストリームでpsの代わりに用いられるこのマルチプレクサを使います。サポートするコーデックはMPEG 1とMPEG audioです。

アイテムのオプションはPSマルチプレクサと同じです。

・ ogg: oggマルチプレクサ。これはXiphプロジェクトに由来するマルチプレクサです。HTTPとファイル出力方法と共に使うことができます。サポートするコーデックはMPEG 1/2/4、MJPEG WMV 1/2 & Theoraです。オーディオストリームにはvorbis、flac、speex、a52、MPEG audioが使えます。

このマルチプレクサにはアイテム・オプションはありません。

・ asf: マイクロソフトASFマルチプレクサ。これはマイクロソフトのソフトウェアがストリーミングで使う標準的なマルチプレクサです。またWMAオーディオ・ファイルのコンテナとして使われます。このマルチプレクサはファイルでもHTTPアウトプット方法でも使うことができます、サポートするコーデックはMPEG4、MJPEG、WMV 1/2 for VIDEO、MPEG audio、a52 for audio streamです。

アイテムのオプションは：

- ・ title=< title> ;
- ・ autor=< author> ;
- ・ copyright=< copyright message> ;

- ・ comment=< comments> ;
- ・ rating=< rating> ; これは何がストリーム・コンテンツの一致するフィールドで表示されるかを設定することができます;
- ・ asfh: これは ASF マルチプレクサの特殊なバージョンで MMSH のため用いられます。MMSH は出力方法だけサポートします。サポートするコーデックは ASF と同じです。アイテムのオプションは ASF と同じです。
- ・ avi: マイクロソフト AVI マルチプレクサ。これは MPEG 4 ファイルの非常に普及したカプセル化フォーマットです。サポートするアウトプット方法はファイルだけです。サポートするコーデックは MPEG 1/2/4、H263、H264&I263 for video、MPEG audio と a52 for audio stream です。このマルチプレクサにはアイテムオプションがありません。注：VLC の AVI マルチプレクサは壊れたファイルしかできないと知られています。
- ・ mpjpeg: マルチパート jpeg マルチプレクサー。
このカプセル化フォーマットは統合されたウェブ・サーバーとともに監視ビデオ・カメラにほとんど用います。このようなストリームは通常ウェブ・ページに埋め込まれ、jpeg の連続イメージとして見ることができるよう、標準的なインターネット・ブラウザで見られます。サポートする出力方法は HTTP だけです。使用できるコーデックは MJPEG だけです。このストリームではサウンドトラックはマルチプレクサできません。このマルチプレクサには利用できるアイテム・オプションはありません。

2. 1. 3. 標準(alias std) - dst

このオプションはストリームが実際保存されるか、送信される場所についての情報を与えることができます。

アクセス・オプションのため使われるパラメーターに依存することに dst オプションの意味がありません。

- ・ ファイル出力方法が用いられたら、dst はファイルが保存されるパスです。
- ・ udp ないし rtp 出力方法が用いられたら、dst は、アドレス：ポート形式で、ユニキャストあるいはマルチキャストの出力アドレスと、オプションで UDP ポートポートです。
- ・ http、https、mmsch 出力方法を選んだ場合、dst は、サーバーがリクエストを受けるローカル・ネットワーク・インターフェースのアドレス、ポート、パスです。アドレスが与えられない場合は、VLC はすべてのネットワーク・インターフェースのリクエストを受けます。このような情報はアドレス：ポート/パスのシンタックスを使って提供されなければなりません。

2. 1. 4. 標準(alias std) - sap

VLC に SAP(Session Announcement Protocol)アナウンスを送るようにしたい場合このオプションを使います。SAP は service discovery protocol で、サーバーで利用可能なストリームのリストを送る特別なマルチキャストアドレスを使用します。

このオプションは udp 出力方法とともにしか使用できません。

2. 1. 5. 標準(alias std) - group

このオプションでストリームのグループのオプションの名前をつけることができます。VLC をクライアントとして使う場合、ストリームを分類するのにこのフィールドを使用します。

このオプションは SAP プロトコルのプライベートな拡張機能を使います。VLC ではクライアントだけがこのフィールドを読むことができます。

このオプションは sap オプションが可能な場合に限り使用することができます。

2. 1. 6. 標準(alias std) - sap-ipv6

SAP アナウンスを IPv4 ではなく、IPv6 を使って送る場合このオプションを使います。

このオプションは sap オプションが可能な場合に限り使用することができます。

2. 1. 7. 標準(alias std) - slp

SLP は Service Location Protocol のことです。これはセッション・アナウンスで SAP の代わりとなるものです。このオプションはそのようなアナウンスを送りたい場合使います。

2. 1. 8. 標準(alias std) - name

このオプションは SAP や SLP アナウンスで送るストリームの名前をつけるために使います。これは SAP や SLP オプションが利用可能な場合だけ使うことができます。

2. 2. ディスプレイ

2. 2. 1. ディスプレイ - display

このモジュールはストリームを表示するために使うことができます。これは特に保存やストリームの際、ストリームをモニターするため、デュプリケート・チェーンで役に立ちます。

利用可能なオプションは次のとおりです：

2. 2. 2. ディスプレイ - noaudio

表示されたストリームでオーディオをオフにするためこのオプションを使うことができます。

2. 2. 3. ディスプレイ - delay

ストリームの表示でディレイを導入するためこのオプションを使うことができます。ディレイは分秒で使われなくてはなりません。

2. 3. rtp

このモジュールは RTP(Real Time Protocol)プロトコル (RFC3550 を見よ)を使うストリームを送信するため使うことができます。

このモジュールで RTSP が利用可能ですが、ビデオ・オン・デマンドはできません。この点については VLM モジュールの記述をご覧ください。

様々な利用可能なオプションが次の通りあります：

2. 3. 1. rtp - dst

このオプションは出力する UDP アドレスを指定することができます。これはホストあるいはマルチキャスト・グループのアドレスになります。sdp=rtsp://オプション (次を参照) をつけないなら、このオプションは必須です。sdp=rtsp://オプションの場合、ストリームは RTSP リクエストをするホストに送信されます。

2. 3. 2. rtp - port

このオプションで最初の要素ストリームを送信するのに使う UDP ポートを設定することができます。このポートは偶数でなくてはなりません。他のストリームは直接この上の偶数ポートを使ってストリームされます。

2. 3. 3. rtp - port-video

これで最初のビデオ要素ストリームを送信するのに使用する UDP ポートを設定することができます。このポートは偶数でなければなりません。

2. 3. 4. rtp - port-audio

このオプションで、最初のオーディオ要素ストリームを送信する UDP ポートを設定できます。このポートは偶数でなければなりません。

2. 3. 5. rtp - sdp

このオプションで、ストリームに一致する SDP(Session Description Protocol)ファイルを利用できるようにする方法を設定できます。オプションは次のとおりです：

- ・ file : //< path to the file> , ローカルファイルとして SDP をエクスポートします。
- ・ http : //< local interface IP:port/path> , ファイルを VLC に統合された HTTP サーバーで使用できるようにします。

注：ローカル・インターフェース IP の引数はオプションです。与えられなければ、VLC はすべての利用できる・インターフェースを使用します。

- ・ rtsp : //< local interface IP:port/path> , RTSP プロトコルを使うように SDP ファイルを利用できるようにします。 RFC 2326 を参照
<https://tools.ietf.org/html/rfc2326>

Note: ローカルインターフェース IP はオプションです。与えられなければ VLC はすべての利用できるインターフェースで使用できるようにします。

- ・ sap, SAP(Session Announcement Protocol, RFC 2974 を参照)を使用して SDP をエクスポートします。
RFC 2974
<https://tools.ietf.org/html/rfc2974>

2. 3. 6. rtp - ttl

これで送信した UDP パケットの TTL(Time to Live)を設定するため使用できるようにします。

2. 3. 7. rtp - mux(マルチプレクサ)

このオプションで、ストリームに送信するのに使用されるカプセル化方法を設定できます。利用できる方法の記述は標準モジュールのマルチプレクサ・オプションを参照してください。

ts だけが RTP ストリームで使用可能です。デフォルトでは、各要素ストリームは分離された RTP メディアとして送信されます。つまりカプセル化は行われません。

2. 3. 8. rtp - rtcp-mux(rtcp-マルチプレクサ)

このオプションは RTP/RTCP マルチプレクサを可能にします(draft-ietf-avt-rtp-and-rtcp-mux を参照)、すなわち RTP パケットと同じポート番号で RTCP パケットを送受信します。デフォルトでは、RTCP パケットは隣のポートで送受信されます。

2. 3. 9. rtp - proto=

これは RTP パケットを運ぶ転送プロトコルを選択します。

とり得る値は次です：

- ・ dccp, 指定された IP アドレス(dst=)の DCCP 接続の入力を受信します
- ・ sctp, 指定された IP アドレス(dst=)の SCTP 接続を受信します(まだ実装されていません)
- ・ tcp, 指定された IP アドレス(dst=)の TCP 接続を受信し、RFC4571 RTP フレーミングを使用します(まだ実装されていません)
- ・ udp, 指定された出力に(ユニキャストかマルチキャストで)UDP パケットを送信します；これがデフォルトの値です
- ・ udplite, 指定された出力に(ユニキャストかマルチキャストで)UDP-Lite パケットを送信します
このオプションは RTP と RTCP パケットの転送プロトコルに UDP の代わりに UDP-Lite を使用します。

2. 3. 10. rtp - name

このオプションでストリームを受信するクライアントに表示される名前を設定することができます。

2. 3. 11. rtp - description

このオプションでストリームの付随的記述を与えることができます。

2. 3. 12. rtp - url

このオプションでストリームに関する付随的な情報とウェブサイトのアドレスを与えることができます。

2. 3. 13. rtp - email

このオプションでコンタクトのメールアドレスを与えることができます。

2. 4. es

es モジュールは、ストリームから異なる要素ストリームを分離し、異なるファイルにそれぞれを保存し、別々の出力に送信するのに用いることができます。

利用できるパラメーターは次のとおりです：

2. 4. 1. es - access-video

これはビデオ要素ストリームを保存し、あるいは送信するのに用いられるメディアを設定するオプションです。可能な値とアイテムのオプションは標準モジュール(上記を参照)のアクセスのオプションと同じです。

2. 4. 2. es - access-audio

このオプションは、オーディオ要素ストリームを保存し、送信するために使用するメディアを設定するのに使います。可能な値とアイテムのオプションは標準モジュール(上記を参照)のアクセス・オプションと同じです。

2. 4. 3. es - access

このオプションは、同じ設定を共有する場合、access-video と access-audio の代わりに使用できます。

2. 4. 4. es - mux-video

このオプションはビデオ要素ストリームのため用いられるカプセル化方法を設定するのに使用します。可能な値とアイテムのオプションは、標準モジュール(上記を参照)のマルチプレクサのオプションと同じです。

2. 4. 5. es - mux-audio

このオプションはオーディオ要素ストリームのため用いられるカプセル化の方法を設定するのに使用します。可能な値とアイテムのオプションは、標準モジュール(上記を参照)のマルチプレクサのオプションと同じです。

2. 4. 6. es - mux

このオプションは、同じ設定を用いる場合、mux-video と mux-audio オプションの代わりに使用できます。

2. 4. 7. es - dst-video

このオプションはビデオ要素ストリームが保存、送信、あるいは利用可能にする場所を設定するため使用します。このオプションの正確な意味は access-video の値に依存し、標準モジュール(上記を参照)の url オプションの場合と同じです。

注：url フィールドで文字列%n を使用すれば、VLC はそれをオーディオかビデオトラックとみなされる数で置き換えます。文字列%c はトラックのコーデックの名前 (FOURCC)で置き換えます。%a はアクセス出力に、%m はマルチプレクサに用いられます。

2. 4. 8. es - dst-audio

このオプションはオーディオ要素ストリームを保存、送信、あるいは利用可能にする場所を設定します。このオプションの正確な意味は access-audio オプションの値に依存し、標準モジュール(上記を参照)の url オプションの場合と同じです。

注：文字列%n を url フィールドで使用すれば、VLC はそれをオーディオ化ビデオ・トラックとみなされる数で置き換えます。文字列%c はトラックのコーデックの名前(FOURCC)で置き換えます。%a はアクセスアウトプットに、%m はマルチプレクサに用いられます。

2. 4. 9. es - dst

このオプションは、同じ設定を共有する場合、dst-video と dstaudio オプションの両方の代わりに用いることができます。

2. 5. 変換

ストリームを変換するため、つまりコーデックを変更したり、ビットレートをエンコードするためこのモジュールを使用することができます。リスケール、インターレース解除、リサンプリングなどいくつかの追加的処理がこのプロセスで行うことができます。

注：オリジナル・ストリームと選択したオプションのビットレートに依存して、変換は高い負荷を CPU にかけることがあります。結果的に、リソースが枯渇すると、リアル・タイム変換ストリームのストリーミングはコマ落ちや、場合によっては画質や音質の劣化につながります。

利用可能なオプションは :

2. 5. 1. 変換 - vcodec

このオプションでインプット・ストリームのビデオ・トラックを変換するコーデックを指定することができます。

可能なコーデックのリストはストリーミング機能のページで調べることができます。

streaming feature page:

<http://www.videolan.org/streaming/features.html>

2. 5. 2. 変換 - vb

このオプションで kbit/s で変換するビデオ・ストリームのビットレートを設定できます。

2. 5. 3. 変換 - venc

これで、ビデオ・ストリームをエンコードするのに使うエンコーダーを設定できます。利用可能なオプションは :

・ `ffmpeg`: これは `libavcodec` エンコード・モジュールです。これは非常に多くの様々なコーデックを扱います(リストはストリーミング機能のページで調べることができます)。
this is the libavcodec encoding module. It handles a large variety of different codecs (the list can be found on the streaming features page).

アイテムのオプションは :

- ・ `keyint=<number of frames>`; 2つのキーフレーム間フレームの最大値を設定します ;
- ・ `hurry-up` CPU がエンコード・レートを維持できない場合、エンコーダーがストリームの質を落とすため設定できます ;
- ・ `interlace` インターレースのエンコードの質を改善することができます ;
- ・ `noise-reduction=<noise reduction factor>`; ノイズ低減アルゴリズムを可能にします(画像の細部を犠牲にして必要なビットレートを低減します);
- ・ `vt=<bitrate tolerance in kbit/s>`; 出力のビデオ・ストリームのための耐用度を設定できます ;
- ・ `bframes=<amount of frames>`; 2つのキーフレーム間の B フレームの数を設定します ;
- ・ `qmin=<quantizer>`; 最小の量子化スケールを設定できます ;
- ・ `qmax=<quantizer>`; 最大の量子化スケールを設定できます ;
- ・ `qscale=<quantizer scale>`; VBR エンコードのための固定量子化スケールを設定します ;
- ・ `i-quant-factor=<quantization factor>`; P フレームと比較した I フレームの量子化ファクターを設定します ;
- ・ `hq=<quality>`; モーション・ベクターのエンコードの室のレベルを選択できます(引数は `simple`, `rd`, `bits` で、デフォルトは `simple` です *改善お願い*) ;
- ・ `strict=<level of compliance>`; より厳しい標準的規格を矯正することができます(可能な値は -1、0、1 です。デフォルトは 0 です);
- ・ `strict-rc` 厳しいレート・コントロール・アルゴリズムを可能にします ;
- ・ `rc-buffer-size=<size of the buffer in bits>`; レート・コントロールに使うバッファのサイズを選択できます(より大きければ効果的なレート・コントロールできます);
- ・ `rc-buffer-aggressivity=<float representing the aggressiveness>`; レート・コントロールのバッファの `aggressiveness` を設定できます *改善お願い* ;
- ・ `pre-me` プレ・モーション評価を可能にします ;

- ・ mpeg4-matrix enable use of the MPEG4 MPEG2 ストリームとの量子化マトリクス。MPEG2 デコーダーとの互換性を維持しながら質を改善します;
- ・ trellis enables 格子量子化(質を改善しますが、処理が遅くなります)。

- ・ theora: Xiph.org theora エンコーダー。モジュールは theora ストリームを作るのに使用されます。theora は特許フリーで使用料フリーのビデオ・コーデックです。

利用できるアイテム・オプションは次のとおりです :

- ・ quality=< quality level> . このオプションで、vb 設定に上書きする VBR ストリームを作ることができます。質のレベルは 1 と 10 の間の整数でなければなりません。高いほうが質がよいです。

- ・ x264. x264 はフリーでオープンソースの h264 エンコーダーです。h264 (あるいは MPEG4-AVC) は全く最近の高画質ビデオコーデックです。

項目のオプションは次のとおりです :

- ・ keyint=< number of frames> 2 つのキーフレーム間のフレームの最大数を設定できます;
- ・ idrint=< number of frames> 2 つの IDR フレーム間のフレームの最大数を設定できます;
- ・ bframes=< amount of frames> I フレームと P フレーム間の B フレームの数を設定できます;
- ・ qp=< quantizer parameter> 固定量子化(1 から 5)を設定することができます;
- ・ qp-max=< quantizer parameter> 最大の量子化値を設定できます;
- ・ qp-min=< quantizer parameter> 最小の量子化値を設定できます;
- ・ cabac CABAC(コンテキスト調整バイナリー数値コーディング)アルゴリズム(遅いが、高品質)を可能にします;
- ・ loopfilter デブロッキング・ループ・フィルターを可能にします;
- ・ analyse 分析モードを可能にします;
- ・ frameref=< amount of frames> 予測に用いられる事前のフレーム数を設定できます;
- ・ scenecut=< sensibility> 能動的に、場面転換で、追加の I フレームをエンコーダーが挿入する方法をコントロールできます;

2. 5. 4. 変換 - fps

このオプションで一秒あたりのフレーム数で変換されるビデオのフレーム率を設定し、ビデオのフレームレートを低減してビットレートを減らすことができます。

2. 5. 5. 変換 - deinterlace

このオプションでエンコーディングの前にインターレースのあるビデオストリームからインタレースを取り除くことができます。

2. 5. 6. 変換 - croptop

このオプションで変換中、元のビデオの上部を切り取ることができます。引数は切り取るビデオの行数です。

2. 5. 7. 変換 - cropbottom

このオプションで元のビデオの下部を切り取ることができます。引数は切り取る最初のラインの Y 調整です。

2. 5. 8. 変換 - cropleft

このオプションで変換中の元のビデオの左側を切り取ることができます。引数は切り取るビデオの列数です。

2. 5. 9. 変換 - cropright

このオプションで、元のビデオの左側を切る取ることができます。引数は切り取る最初の列の X 調整です。

2. 5. 10. 変換 - scale

このオプションは変換中ビデオをリスケールする比率を設定できます。このオプションはストリームのビットレートを低減するのに特に有効になることができます。

2. 5. 11. 変換 - width

このオプションでピクセル単位で変換されるビデオの幅を設定することができます。

2. 5. 12. 変換 - height

このオプションでピクセル単位で変換するビデオの高さを設定することができます。

2. 5. 13. 変換 - acodec

このオプションは変換する入力ストリームのオーディオ・トラックのコーデックを設定します。利用可能なコーデックのリストはストリーミング機能のページで調べることができます。

streaming features page:

[http : //www.videolan.org/streaming/features.html](http://www.videolan.org/streaming/features.html)

2. 5. 14. 変換 - ab

このオプションで kbit/s 単位で変換するオーディオストリームのビットレートを設定できます。

2. 5. 15. 変換 - aenc

これで、オーディオ・ストリームをエンコードするのに用いるエンコーダーを設定できます。利用可能なオプションは次のとおりです :

- ・ `ffmpeg`: これは `libavcodec` エンコード・モジュールです。大変幅広いさまざまなコーデックを扱います(そのリストはストリーミング機能のページで調べることができます)。

streaming features page:

[http : //www.videolan.org/streaming/features.html](http://www.videolan.org/streaming/features.html)

- ・ `vorbis`. このモジュールは Xiph.org プロジェクトの `vorbis` エンコーダーを使用します。vorbis は無料でオープンソースのライセンスフリーの非可逆オーディオコーデックです。

アイテムのオプションは次のとおりです :

- ・ `quality=< quality level>`; デフォルトの CBR(固定ビットレート)でエンコードする代わりに VBR(可変ビットレート)を使うことができ、そして

音質レベル(1から10、高いほうが高音質)を設定します :

- ・ max-bitrate=< bitrate in kbit/s> vbr エンコードの最大ビットレートを設定できます;
- ・ min-bitrate=< bitrate in kbit/s> vbr エンコードの最小ビットレートを設定できます;
- ・ cbr cbr エンコードを強制します。
- ・ speex. これで Xiph.org の soeex エンコーダーを使用します。Speex は非可逆オーディオ・コーデックで、非常に低いビットレート(約10kbit/s)、特にビデオ会議に最適です。

2. 5. 16. 変換 - samplerate

このオプションで Hz 単位で変換するオーディオ・ストリームのサンプルレートを設定できます。サンプルレートの低減は、出力するオーディオ・ストリームのビットレートを小さくする方法となります。

2. 5. 17. 変換 - channels

このオプションで、出力するオーディオ・ストリームのチャンネル数を設定できます。これは2チャンネル以上をサポートしない低ビットレートのオーディオ・ストリームのコーデックに役立ちます。

2. 5. 18. 変換 - scodec

このオプションで、変換する入力ストリームの字幕トラックのフォーマットを設定します。利用可能なコーデックのリストはストリーミング機能のページで調べることができます。

streaming features page:

<http://www.videolan.org/streaming/features.html>

2. 5. 19. 変換 - senc

これで、字幕ストリームをエンコードするのに使うコンバーターを設定します。現在われわれが使用できる字幕エンコーダーは dvdsub だけです。

2. 5. 20. 変換 - soverlay

このオプションで変換中、ビデオに直接字幕をレンダリングできます。字幕の変換とそのストリームを行う senc/scode とこのオプションを混同しないでください。

2. 5. 21. 変換 - sfilter

このオプションでビデオの上に、いわゆる字幕フィルター(すなわち、ロゴ、テキスト文章など)で生成される画像をレンダリングすることができます。

利用可能な字幕フィルターのリストは[<http://www.videolan.org/streaming/features.html> のストリーミング機能のページ]で調べることができます(改善希望?)。このモジュールのアイテムのぷしょんはつぎのコマンドを使って端末で調べることができます :

```
% vlc -p --advanced &lt; module name&gt;
```

2. 5. 22. 変換 - threads

このオプションで、ストリーミングをエンコードするのに使うスレッド数を設定できます。コンピュータのプロセッサ数に合わせることで(あるいは Intel P4 HT プロセッサではこの数を倍にして)、変換のパフォーマンスを改善できます。

2. 5. 23. 変換 - audio-sync

このオプションが可能な場合、VLC はオーディオ・トラックにビデオトラックを同期するためビデオ・フレームを落としたり、重複させたりします。

2. 5. 24. transcode - vfilter

変換処理中、ビデオフィルターを使用します。vfilter のパラメーターは VLC フィルターの上級使用で調べることができます。

Advanced Use of VLC Filters :

http://wiki.videolan.org/Documentation:Play_HowTo/Advanced_Use_of_VLC#Filters

使用例

vlc input_file

```
--sout="#transcode{vfilter=adjust{gamma=1.5},vcodec=theo,vb=2000,scale=0.67,acodec=vorb,ab=128,channels=2}:standard{access=file,mux=ogg,dst="output_file.ogg"}"
```

これは"input_file"のガンマを 1.5 に調整し、ビデオサイズ(解像度)を 0.67 にリサイズし(例えば 1080x720 を 720x480 に)、ビットレート 2000k/s で Theora コーデックを使いビデオを、128kb/s で Vorbis コーデックを使ってコンバートし、Ogg コンテナにビデオとオーディオをカプセル化し、"output_file.ogg"に保存します。

2. 6. デュプリケート

このモジュールはストリームを重複させるのに使うことができ、いくつかの異なるチェーンで処理するように出来ます。

利用可能なオプションは次のとおりです :

2. 6. 1. デュプリケート - dst

このオプションで、多重化されたストリームを処理するためのチェーンを作ります。

注 : dst オプションはストリームを実際多重化するのと同じ重複ブロックで使用しなければなりません。以前に記述したストリーム出力モジュールのいずれもこのオプションのパラメーターとしても使用できます。

2. 6. 2. デュプリケート - select

このオプションが全ストリームの一部の要素ストリームだけを多重化するのに使用できます。コマンドで区切ることで、いくつかの基準を与えることができます。パラメーターを必要とする基準には、es や program のように、criteria=num_start-num_end のシンタックスを使って、範囲を設定することができます。

利用可能なパラメーターは次のとおりです :

- ・program=: 選択したプログラム(あるいは SID)に属する要素ストリームだけ多重化します。これは MPEG2/TS ストリームの場合だけ有効です。
- ・noprogram=: 選択したプログラム(あるいは PID)に属する要素ストリームを多重化しません。このオプションは MPEG2/TS ストリームの場合だけ有効です。
- ・es=: 選択した id の要素ストリーム(es)だけ多重化します。
- ・noes=: 選択した id の要素ストリーム(es)を多重化しません。

- ・ video: ビデオの要素ストリームだけを多重化します。
- ・ novideo: ビデオ要素ストリームを多重化しません。
- ・ audio: オーディオ要素ストリームだけを多重化します。
- ・ noaudio: オーディオ要素ストリームを多重化しません。
- ・ spu: 字幕要素ストリームだけを多重化します。
- ・ nospu: 字幕要素ストリームを多重化しません。

使用例:

```
#duplicate{dst=std{...},select="program=100-200,novideo"}
```

この多重化チェーンはPIDが100から200までのプログラムに属する非ビデオ要素ストリームだけを出力します。

2. 7. その他

いくつかの追加的汎用オプションは次のとおりです:

- ・ --sout-all, --no-sout-all: すべてのESのストリーミングができるようにします(デフォルトでは不能)。デフォルトでは、(最初の)VLCはひとつのオーディオESとひとつのビデオのストリーミングESだけをストリームします。sout-allを可能にすると、すべてのES(audio、video、SPU)がストリームされます。
- ・ --sout-keep, --no-sout-keep: soutをオープンのままにします(デフォルトでは不能): 可能なら、さまざまなプレイリストのアイテムで同じsoutインスタンスを使用します。
- ・ --no-sout-audio: このオプションで、出力されたストリームのオーディオを不能にすることができます。
- ・ --no-sout-video: このオプションで、出力されたストリームのビデオを不能にすることができます。

2. 8. 単純なシンタックス

ストリーム出力は単純化されたシンタックスを提供します。それで標準モジュールの主要なオプションだけを使用するようにすることができます:

```
% vlc input_stream --sout access/mux : //url
```

access、mux、urlは標準モジュールのオプションで規定されます。

3. サンプル

VLCのストリーム出力の複雑なシンタックスを完全に理解するには、次節のサンプルを見てください。

第七回 ストリーミングの上級サンプル、マルチ・ファイル・ストリーミング、ストリーミングでのマルチキャストの使用

ストリーミングの上級サンプル、マルチ・ファイル・ストリーミング、ストリーミングでのマルチキャストの使用

Advanced streaming with samples, multiple files streaming, using multicast in streaming
Documentation:Streaming HowTo/Advanced streaming with samples, multiple files streaming, using multicast in streaming

著者:wwwital

このドキュメントは VideoLAN を使用してストリーム、マルチ・ファイル・ストリーム、マルチキャストなどの方法をサンプルとともに説明します。

目次

1. UDP ストリーミングのサンプル
2. マルチキャスト RTP ストリーミングのサンプル
3. マルチファイルでのマルチキャスト RTP ストリーミング、サンプル付き
4. 特別なマルチファイル記述の設定ファイル

1. UDP ストリーミングのサンプル

標準的な UDP ストリーミング

```
% vlc -vvv file:///home/vlc/2007.avi --sout #std{access=udp,mux=ts,dst=:1234}'
```

難しい点はありません。ファイル 2007.avi ファイルを、/home/vlc/から、udp ポート 1234 にストリーミングします。

2. マルチキャスト RTP ストリーミングのサンプル

```
% vlc -vvv file:///home/vlc/Jumper.avi --sout  
'#rtp{access=udp,mux=ts,dst=224.255.1.1,port=1234,sap,group="Video",name=Jumper Movie}':sout-all
```

難しいですか？そんなことはありません。これはすでに見たキーのファイルです。UDP のサンプルと同じく、キー --sout は出力を始めます。次いで RTP で、access のタイプは udp、muxser(マルチプレクサ)のタイプは ts に設定します。そして出力先(dst)としてマルチキャスト IP アドレス 224.255.1.1、ポート 1234 を指定します。そしていくつかのキーがあります。VLC は SAP(service advertisements protocol を見よ)を使って、このストリームの告知をし、ストリーミンググループに「Video」と設定し、このストリームに「Jumper Movie」と名づけます。

3. マルチファイルでのマルチキャスト RTP ストリーミング、サンプル付き

注意：これを始めると停止できません。
わたしは数時間かけてこの解決策を見つけました。

```
% vlc -vvv --color -I telnet --telnet-password "i_dont_know_this_password" --vlm-conf=/home/vlc/vlc.streaming.conf
```

VLC はキー --color で出力をカラー化しなくてはなりません。telnet サーバーを開きます。われわれは

本当にそれをコントロールしなくてはならないのでしょうか？これが -I tenet キーです。コンソールにアクセスするため "i_dont_know_this_password" を設定します。標準 telnet ポート 4212 を使用します。変える必要がある場合は --telnet-port xxx を使います。VLC が複数のファイルを記述した特別なファイルをスタート時に開くよう指示するのに --vlm-conf=/home/vlc/vlc.streaming.conf を使います。

4. 特別なマルチファイル記述の設定ファイル

・ vlc.streaming.conf

この設定ファイルを使用して、2007.avi と Jumper.avi の2つのビデオをストリームします。これを行うためには、2つのチャンネルを記述しなければなりません。channel1 と channel2 が入力と出力のフォーマットを設定します(これをマルチキャストします)：

```
-----
new channel1 broadcast enabled
setup channel1 input file:///home/vlc/2007.avi loop
setup channel1 output
#rtp{access=udp,mux=ts,dst=224.255.1.1,port=1234,sdp=sap,sap,group="Video",name="2007
Movie"}
-----
new channel2 broadcast enabled
setup channel2 input file:///home/vlc/Jumper.avi loop
setup channel2 output
#rtp{access=udp,mux=ts,dst=224.255.1.2,port=1234,sdp=sap,sap,group="Video",name="Jumper
Movie"}
-----
control channel1 play
control channel2 play
-----
```


第八回 変換 - VLC を使った変換についての詳細情報

変換 - VLC を使った変換についての詳細情報
<https://wiki.videolan.org/Transcode/>

変換はビデオ・ファイルを別のフォーマットやビットレートに変更するプロセスです。

VLC の変換は動画を別のフォーマットの新しいファイルにコピーします。そのため、オリジナルと新しいファイルの両方を得ることになります。これはこの追加されるファイルを保存する十分なハードディスクのスペースが必要だということを意味します。また変換が非常に遅くなることがあるため、ある程度高速なコンピュータが必要です。

VLC では、変換はネットワークではなく、ファイルに保存する場合を除いて、ネットワークでストリームするのと全く同じです。

目次

1. 互換性の問題
2. ウィザードを使った変換
3. コマンドラインによる変換
 3. 1. MS ウィンドウズの場合
 3. 2. 完全な非対話的変換
4. キャンバスとパディング

- 5. 非対話的バッチ変換 (ウィンドウズ)
 - 5. 1. バッチファイル
 - 5. 2. VB スクリプト
- 6. 非対話的バッチ変換(Linux)
 - 6. 1. 非対話的マージ/変換 (ウィンドウズ)
 - 6. 2. サンプル・スクリプト

1. 互換性の問題

入力ストリームがすでに出力ストリームのエンコードと同じなら変換の必要は全くありません(コマンドラインの変換設定をしないだけで、それゆえ変換はしません)。GUI ではこれを「オリジナルトラックのままにする」と呼びます (と思う)。

変換設定を決定した場合、あるコンテナ・フォーマット (マルチプレクサ) はいくつかのコーデックだけをサポートすることに注意してください—どれが何をサポートするかを調べるにはストリーミング機能の互換性を見てください

streaming feature:

<http://www.videolan.org/streaming/features.html>

現在のフォーマットが望むマルチプレクサと互換性がない場合、いずれか互換性のあるものに変換するか、異なるマルチプレクサを選ぶかしなければなりません。マルチプレクサと互換性のないストリームを選ぶと、何も反応なく失敗します。エラーのデバッグに役立てるためメッセージ・ウィンドウのエラーメッセージを調べることができます。

2. ウィザードを使った変換

VLC には変換とストリーミングのウィザードがあります。ファイルを変換するためには、変換オプションを選択するだけです。変換するフォーマットを尋ねられたら、ビデオ・コーデック、オーディオ・コーデック、コンテナ・フォーマットを指定できます。

3. コマンドプロンプトによる変換

VLC を使ったより柔軟な変換方法は、VLC を起動するためコマンド・プロンプトを使うことです。変換はストリーミングと同じ作業をします。例えば、次のコマンドは asf ファイルを MPEG-2 ファイルに変更します：

```
vlc "C:\Movies\Your File.asf"  
:sout=#transcode{vcodec=mp2v,vb=4096,acodec=mp2a,ab=192,scale=1,channels=2,deinterlace,audio-sync}:std{access=file,mux=ps,dst="C:\Movies\Your File Output.ps.mpg"}
```

access=file はファイルに(ストリームではなく)出力を保存するよう VLC に支持し、dst は新しい(出力)ファイルの場所です。

注1：MAC OS X の場合、BUS エラーを回避するため VLC の代わりに clivlc を使うべきです。

注2：古い VLC(0.8.6 以前)では"dst"の代わりに"url"を使ってください。

変換にはしばらく時間がかかります。そのため、最初の30秒だけエンコードするよう --stop-time=30 のようなオプションを使うのが望ましい。これはファイルが正しく変換されたことをチェックできること、また出力が適切な質であるかをチェックできることを意味します(注：Mac OS X では、この方法は有効でないようです。少し経ったら、VLC はすでに変換された部分を適切に書き出します)。

変換文にはビデオ・コーデックと acodec を変更するため vcodec と vb が、オーディオ・コーデックを変更するため ab を挿入することができます。vcodec がない場合ビデオ・コーデックは同じままになります(acodec と同じ)。

利用できる共通の追加オプションには(確実にオーディオを正確に同期させるための)audio-sync と(イ

インターレースビデオの質をわずかに高める)deinterlace があります。

3. 1. MS ウィンドウズの場合

ウィンドウズのシンタックスはわずかな違いがあります：

```
C:\Program Files\VideoLAN\VLC>vlc -vvv "D:\688497.flv"  
--sout=#transcode{vcodec=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:standard{access=file,  
mux=ts,dst="D:\asd.mpg"}
```

(注：おそらく 0.8.6 以前の古い VLC は"dst"の代わりに"url"を使用します)

(注：コマンドラインの解釈のため、場合により、特にシングルクォーテーション"とダブルクォーテーション"のブロック内で、バックスラッシュ\ダブルバックスラッシュを使ってエスケープしなければならないでしょう。そのためファイル名は D:\path\to\file.mpg となります)

3. 2. 完全な非対話的変換

完全な非対話的変換のためには(Mac OS X で行う場合必要なケース)、上記のサンプルは次のように書き換えます

```
vlc -I dummy "C:\Movies\Your File.asf"  
:sout=#transcode{vcodec=mp2v,vb=4096,acodec=mp2a,ab=192,scale=1,channels=2,deinterlace,au  
dio-sync}:std{access=file, mux=ps,dst="C:\Movies\Your File Output.ps.mpg"} vlc : //quit
```

ふたつの追加のオプションは

```
-I dummy  
  Disables the graphical interface  
vlc : //quit  
  Quit VLC after transcoding
```

DVD から変換する場合、DVD のメニューシステムの対話画面を避けるため、dvd : // の表記の代わりに、dvdsimple : // の表記を使用する必要もあります。ディスクを開くとき、'no DVD menus'オプションを選んで GUI でこれを使用することができます。

ウィンドウズで、"-I dummy"オプションは'vbw buffer overflow'のエラーメッセージがコマンドプロンプトに出力される場合、テキスト・インターフェースを開く代わりに、"-I rc"を試すことができますが、ウィンドウズではダミーがすることを正確にします。

詳細は VideoLAN's Official Documentation を参照
<http://www.videolan.org/doc/streaming-howto/en/ch03.html>

4. キャンバスとパディング

キャンバスとパディング・オプションはバージョン 0.9.*以降変更されました。wiki のほとんどのサンプルは通用しません。

```
transcode{vcodec=mp2v,vfilter=canvas{width=640,height=480}}
```

他のオプション：

- ・ aspect=4:3 : Aspect ratio

- ・ padd:
 - ・ true: Pad with black bars
 - ・ false: crop video

```
transcode{vcodec=mp2v,vfilter=croppadd{cropttop=20,cropbottom=30,paddleft=100}}
```

top、left、right、bottom をピクセル単位でカットまたは余白挿入をします。

width、height パラメーターは使わないでください！vfilter 以前に適用すると望まない結果が生じます。

5. 非対話的バッチ変換 (ウィンドウズ)

(略)

* 訳注：インデントができないため、原文をご覧ください

6. 非対話的バッチ変換(Linux)

対話画面を使わずにいくつかのファイルを変換するためには、ファイルがあるフォルダーにバッチスクリプトを作り次の bash スクリプトにつぎのコマンドを加えます：

* 訳注：テストはしていません。

```
#!/bin/bash
vcodec="VIDEO_CODEC"
acodec="AUDIO_CODEC"
bitrate="VIDEO_BITRATE"
arate="AUDIO_BITRATE"
ext="OUTPUT_EXT"
mux="MUXER"
vlc="PATH_TO_VLC"
fmt="INPUT_EXT"

for a in *$fmt; do
$vlc -I dummy -vvv "$a" --sout
"#transcode{vcodec=$vcodec,vb=$bitrate,acodec=$acodec,ab=$arate,channels=6}:standard{mux=$
mux,dst=\"$a.$ext\",access=file}" vlc ://quit
done
```

6. 1. 非対話的マージ/変換 (ウィンドウズ)

How_to_Merge_and_Transcode_Multiple_Videos を参照：

https://wiki.videolan.org/How_to_Merge_and_Transcode_Multiple_Videos/

6. 2. サンプル・スクリプト

(略)

訳注：プレイリストファイルの問題

オーディオファイルを連続再生するためにはプレイリストを作成する必要があります。
ローカルでは単に複数のファイルを開く（メディア）、ディレクトリを開く（メディア）で自動的にプレイリストが作成されます。

開いたあとプレイリストファイルを保存する（メディア）とプレイリストファイルが拡張子.xspfのXMLファイルとして保存されます（テキストファイルです）。
プレイリストファイルはエディターで開くことができます。またエディターでプレイリストを作成編集ができます。

これを訳出するのは、連続ストリームをすると曲の頭で変換が行われるため、頭が切れる場合が生じる不具合に対応するためです。ここまで訳出してきた「変換しない」ようコマンドラインを編集したらいいのじゃないかと思ってますが、ついでなので訳出します。う～ん？

XSPF

<https://wiki.videolan.org/XSPF/>

VLC プレイリスト

スピフと発音する XSPF は XML 形式のプレイリストです。これは Xiph によってサポートされます。これはフリーでオープンフォーマットなので簡単に、自由にプレイリストを共有するのに使うことができます。

目次

- 1. サンプル
- 2. 互換性
- 3. VLC 拡張
 - 3. 1. vlc:node
 - 3. 2. vlc:item
 - 3. 3. vlc:id
 - 3. 4. vlc:option
 - 3. 5. VLC 拡張をとまなう XSPF のサンプル
- 4. 参照
- 5. 安全でない操作

1. サンプル

非常に単純な例は次のようなものです：

*訳注：“__”はインデントのために加えました。空白に置き換えてください。

```
<?xml version="1.0" encoding="UTF-8"?>
<playlist version="1" xmlns="http://xspf.org/ns/0/">
  __<trackList>
    _____<track><location>file:///mp3s/song_1.mp3</location></track>
    _____<track><location>file:///mp3s/song_2.mp3</location></track>
    _____<track><location>file:///mp3s/song_3.mp3</location></track>
  __</trackList>
</playlist>
```

2. 互換性

XSPF の互換性について知りたい場合は、次に追加してください

<https://wiki.videolan.org/index.php?title=XSPF&action=edit>

3. VLC 拡張

XSPF は特別なデータをアプリケーションに追加できる拡張をサポートしています。この拡張は次の項目にあります：

- ・ playlist
- ・ track

拡張フォーマットはネームスペース `xmlns:vlc="http://www.videolan.org/vlc/playlist/ns0/"` で規定されています：

```
&lt;playlist version="1" xmlns="http://xspf.org/ns/0/"
xmlns:vlc="http://www.videolan.org/vlc/playlist/ns/0/"&gt;
...
&lt;extension application="http://www.videolan.org/vlc/playlist/0"&gt;
...
&lt;/extension&gt;
&lt;/playlist&gt;
```

現在、拡張は次の要素をサポートします：

- ・ vlc:node
- ・ vlc:item
- ・ vlc:id
- ・ vlc:option

拡張 `vlc:node` と `vlc:item` はプレイリストのツリーを表示する方法を設定するのに用いられます。それは標準 XSPF ではサポートされていません。

3. 1. vlc:node

この要素はプレイリストのノードとして表示されます。それはプレイリスト・ブロック (`playlist/extension` の下)に見ることができます。その名前だけが設定できます：

```
&lt;vlc:node title="Node title"&gt;
[list of vlc:item or vlc:node]
&lt;/vlc:node&gt;
```

3. 2. vlc:item

この要素はプレイリストの項目（ノードではなく）を表現します。それはプレイリスト・ブロック (`playlist/extension` の下)に見ることができます。それはトラック ID だけを格納します(下記参照、`vlc:id`):

```
&lt;vlc:item tid="42" /&gt;
```

3. 3. vlc:id

この要素はトラック ID を設定します。トラック・ブロックの拡張(`playlist/trackList/track/extension` の下)に見ることができます。

<vlc:id>42</vlc:id>;

id の値は vlc:item の属性 tid の値に一致します。それぞれの vlc:id はエントリーごとに異ならなければならないことに注意。そうでないとすべて表示されません。

3. 4. vlc:option

この要素で入力項目にオプションを加えることができます。それはトラック・ブロックの拡張 (playlist/trackList/track/extension の下)に見ることができます。

<vlc:option>option-name</vlc:option>;

あるいはオプションがひとつの値を持つ場合は :

<vlc:option>option-name=option-value</vlc:option>;

3. 5. VLC 拡張をともなう XSPF のサンプル

サンプルとしてこれらを要約してみます。これはプレイリストの個々のエントリーごとに VLC オプションを付与することを示しています。

* 訳注 : インデントができず、見難くてすみません。オリジナルのページをご覧ください。

```
<?xml version="1.0" encoding="UTF-8"?>
<playlist version="1" xmlns="http://xspf.org/ns/0/"
xmlns:vlc="http://www.videolan.org/vlc/playlist/ns/0/">
  <title>Playlist</title>
  <location>D:/media/example.xspf</location>
  <trackList>
    <track>
      <title>DVD seconds 42 to 45 muted</title>
      <location>dvd : //d:\@1</location>
      <extension application="http://www.videolan.org/vlc/playlist/0">
        <vlc:id>3</vlc:id>
        <vlc:option>start-time=42</vlc:option>
        <vlc:option>stop-time=45</vlc:option>
        <vlc:option>no-audio</vlc:option>
        <vlc:option>some-option=100</vlc:option>
      </extension>
    </track>
    <track>
      <title>DVD seconds 49-55 unmuted (normal)</title>
      <location>dvd : //d:\@1</location>
      <extension application="http://www.videolan.org/vlc/playlist/0">
        <vlc:id>4</vlc:id>
        <vlc:option>start-time=49</vlc:option>
        <vlc:option>stop-time=55</vlc:option>
      </extension>
    </track>
  </trackList>
</playlist>
```

次はプレイリスト内のネストされたエントリーを表示する方法を示すサンプルです：
* 訳注：インデントができず、見難くてすみません。オリジナルのページをご覧ください。

```
&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;playlist version="1" xmlns="http://xspf.org/ns/0/"
xmlns:vlc="http://www.videolan.org/vlc/playlist/ns/0/"&gt;
  &lt;title&gt;Playlist&lt;/title&gt;
  &lt;location&gt;D:/media/example.xspf&lt;/location&gt;
  &lt;trackList&gt;
    &lt;track&gt;
      &lt;title&gt;Track 1&lt;/title&gt;
    ...
    &lt;extension application="http://www.videolan.org/vlc/playlist/0"&gt;
      &lt;vlc:id&gt;0&lt;/vlc:id&gt;
      &lt;/extension&gt;

    &lt;/track&gt;
    &lt;track&gt;
      &lt;title&gt;Track 2&lt;/title&gt;
    ...
    &lt;extension application="http://www.videolan.org/vlc/playlist/0"&gt;
      &lt;vlc:id&gt;1&lt;/vlc:id&gt;
      &lt;/extension&gt;
    &lt;/track&gt;
    &lt;track&gt;
      &lt;title&gt;Track 3&lt;/title&gt;
    ...
    &lt;extension application="http://www.videolan.org/vlc/playlist/0"&gt;
      &lt;vlc:id&gt;2&lt;/vlc:id&gt;
      &lt;/extension&gt;
    &lt;/track&gt;
    &lt;track&gt;
      &lt;title&gt;Track 4&lt;/title&gt;
      &lt;location&gt;dvd://e:\@1&lt;/location&gt;
      &lt;extension application="http://www.videolan.org/vlc/playlist/0"&gt;
        &lt;vlc:id&gt;3&lt;/vlc:id&gt;
        &lt;vlc:option&gt;my-option=100&lt;/vlc:option&gt;
        &lt;vlc:option&gt;start-time=42&lt;/vlc:option&gt;
        &lt;vlc:option&gt;stop-time=45&lt;/vlc:option&gt;
        &lt;vlc:option&gt;no-audio&lt;/vlc:option&gt;
        &lt;/extension&gt;
      &lt;/track&gt;
    &lt;/trackList&gt;
    &lt;extension application="http://www.videolan.org/vlc/playlist/0"&gt;
      &lt;vlc:node title="Node 1"&gt;
        &lt;vlc:item tid="0" /&gt;
        &lt;vlc:item tid="1" /&gt;
      &lt;vlc:node title="Node 2"&gt;
        &lt;vlc:item tid="2" /&gt;
        &lt;vlc:item tid="3" /&gt;
      &lt;/vlc:node&gt;
    &lt;/extension&gt;
  &lt;/playlist&gt;
```

</vlc:node>
</extension>
</playlist>

このプレイリスト・サンプルは次のように示すことができます :

```
Playlist  
|- Node 1  
|---- Track 1  
|---- Track 2  
-----> | Node 2  
-----|-- Track 3  
-----|-- Track 4
```

4. 参照

Media Library : VLC Media Player に組み込まれた機能
https://wiki.videolan.org/Media_Library/

フォーマットとして XML を使用するのには多くの長所があります。詳細は次で見ることができます

- ・ Xiph.org
<http://www.xiph.org/>
- ・ XSPF project website.
<http://www.xspf.org/quickstart/>

5. 安全でない操作

ボリュームの設定、sout 出力先の設定などの操作はプレイリストでは「危険」と考えられます。そのためその設定はコマンドラインで追加すべきです。エラーメッセージ「unsafe option"sout-audio" has been ignored for security reasons、のようなエラーメッセージが出るでしょう。

<http://forum.videolan.org/viewtopic.php?f=14&t=78945>

#####