

# AquesTalk10 Linux マニュアル

株式会社 アクエスト  
www.a-quest.com

## 概要

本文書は、音声合成ライブラリ AquesTalk10 Linux をアプリケーションに組み込んで使用するためのプログラミングの方法、注意点を示したものです。

AquesTalk10 は、かな表記の音声記号列から WAV 音声データを生成するライブラリです。

AquesTalk10 は、声質パラメータを指定することで様々な声質の音声を生成できる特徴があります。

本ライブラリを使用するには、開発ライセンスキーの設定が必要です。このライセンスキーを設定しない場合は、評価版として動作し、以下の制限があります。

評価版の制限      「ナ行、マ行」を指定すると、すべて「ヌ」と発声します

また、本ライブラリをアプリケーションに組み込んで使用する際には**使用ライセンス**、配布には**頒布ライセンス**が必要です。ライセンスの種類や購入方法は、弊社サイトのライセンスのページを参照してください。

## 仕様

ライブラリ形式	so 形式 共有ライブラリ
対応 OS	Linux x86-64(64bit 版)/ IA-32(32bit 版)
入力データ形式	かな表記音声記号列(Shift JIS/UTF8/UTF16)
出力データ形式	WAV フォーマット(16KHz サンプリング,16bitPCM,モノラル)データ *サンプリング周波数は声質パラメータにより変化
声種	ベース音素片 3 種    パラメータでユーザカスタマイズ可
関数 I/F	C 関数呼び出し
マルチスレッド	対応
ライブラリサイズ	約 400KByte
外部依存ライブラリ	libc.so.6, libgcc_s.so.1

## ビルド・実行

### ライブラリ配置

AquesTalk10 Linux のライブラリは共有ライブラリとなっています。リンク時、および実行時に本ライブラリが必要になります。

以下に一例として、`/usr/lib` に本ライブラリを配置する方法を示します。これによりリンク時および実行時にライブラリにアクセスできるようになります。配置ディレクトリはお使いの環境に応じて変更してください(`/usr/lib64/` や `/usr/lib32` など)。また、バージョン番号もライブラリの実際のファイル名に合わせてください。パッケージ内の `lib32` フォルダが 32bit 版、`lib64` フォルダが 64bit 版となります。

```
# cp libAquesTalk10.so.1.0 /usr/lib
# ln -sf /usr/lib/libAquesTalk10.so.1.0 /usr/lib/libAquesTalk10.so.1
# ln -sf /usr/lib/libAquesTalk10.so.1 /usr/lib/libAquesTalk10.so
# /sbin/ldconfig -n /usr/lib
```

### コンパイル・リンク

ライブラリを呼び出すソースコードにヘッダファイル(`AquesTalk.h`)をインクルードしてコンパイルします。リンク時には、`-lAquesTalk10` を指定してリンクしてください。

次に、サンプルプログラム `HelloTalk.c` をコンパイル・リンクする方法を示します。

```
$ g++ -o HelloTalk HelloTalk.c -lAquesTalk10
```

g++の代わりに gcc でリンクするときに、`undefined reference to `__gxx_personality_v0'` などのエラーが出る場合は、`-lstdc++` を追加して指定してください。

### サンプルプログラムの実行

次のコマンドで、`test.wav` に音声データが生成できれば OK です。エラーの場合には、文字コードやライブラリの配置が正しく行われているか確認してください。

```
echo "これわ、ごーせー/お'んせーです。" | ./HelloTalk > test.wav
```

音声記号列にシェルの特許文字が含まれているので、ダブルクォーテーション(")で囲むなど、適当にエスケープする必要があります。

文字コードは UTF8 になっています。文字コードを変更する場合は、`AquesTalk_Synthe_Utf80` 部分を文字コードに応じて変更してください。

なお、開発ライセンスキーを設定していないので、評価版の制限で「ナ行、マ行」が、すべて「ヌ」になります。

### プログラミングの注意

`AquesTalk_Synthe0`関数で返された音声データは、使用後に `AquesTalk_FreeWav0`で解放してください。アプリ側で `free0` などを使用して解放すると、異なる C ランタイムの場合にハングします。

## 関数 API

### AquesTalk\_Synthe

AquesTalk.h

説明	かな表記音声記号列(SJIS)から音声波形を生成します
構文	unsigned char * <b>AquesTalk_Synthe</b> (const AQT_VOICE * <i>pParam</i> , const char * <i>koe</i> , int * <i>size</i> )
引数	
<i>pParam</i>	声質パラメータ(構造体 AQT_VOICE のアドレス)を指定
<i>koe</i>	音声記号列(SJIS NULL 終端)を指定
<i>size</i>	生成した音声データのサイズが返る[byte](エラーの場合はエラーコードが返る)
戻り値	WAV フォーマットの音声データを返す。 ヒープ領域を関数内部で確保するので、解放は AquesTalk_FreeWave()で行う。 エラー時は、NULL を返す。このとき size にエラーコードが設定される。

### AquesTalk\_Synthe\_Utf8

AquesTalk.h

説明	かな表記音声記号列(UTF-8)から音声波形を生成します
構文	unsigned char * <b>AquesTalk_Synthe_Utf8</b> (const AQT_VOICE * <i>pParam</i> , const char * <i>koe</i> , int * <i>size</i> )
引数	
<i>pParam</i>	声質パラメータ(構造体 AQT_VOICE のアドレス)を指定
<i>koe</i>	音声記号列(UTF-8 NULL 終端)を指定
<i>size</i>	生成した音声データのサイズが返る[byte](エラーの場合はエラーコードが返る)
戻り値	WAV フォーマットの音声データを返す。 ヒープ領域を関数内部で確保するので、解放は AquesTalk_FreeWave()で行う。 エラー時は、NULL を返す。このとき size にエラーコードが設定される。

### AquesTalk\_Synthe\_Utf16

AquesTalk.h

説明	かな表記音声記号列(UTF-16)から音声波形を生成します
構文	unsigned char * <b>AquesTalk_Synthe_Utf16</b> (const AQT_VOICE * <i>pParam</i> , const unsigned short * <i>koe</i> , int * <i>size</i> )
引数	
<i>pParam</i>	声質パラメータ(構造体 AQT_VOICE のアドレス)を指定
<i>koe</i>	音声記号列(UTF-16 NULL 終端 BOM の有無は問わない エンディアンは実行環境に従う)を指定

<i>size</i>	生成した音声データのサイズが返る[byte](エラーの場合はエラーコードが返る)
<b>戻り値</b>	WAV フォーマットの音声データを返す。 ヒープ領域を関数内部で確保するので、解放は <code>AquesTalk_FreeWave()</code> で行う。 エラー時は、NULL を返す。このとき <i>size</i> にエラーコードが設定される。

**AquesTalk\_FreeWave**

AquesTalk.h

<b>説明</b>	音声データの領域を開放
<b>構文</b>	<code>void AquesTalk_FreeWave(unsigned char *wav)</code>
<b>引数</b>	
<i>wav</i>	WAV フォーマットのデータ( <code>AquesTalk_Synthe()</code> 等で生成した音声データ)
<b>戻り値</b>	なし

**AquesTalk\_SetDevKey**

AquesTalk.h

<b>説明</b>	開発ライセンスキーを設定。音声波形を生成する前に一度呼び出すことで、以降、製品版とし動作し、評価版の制限がなくなる。
<b>構文</b>	<code>int AquesTalk_SetDevKey(const char *key)</code>
<b>引数</b>	
<i>key</i>	開発ライセンスキー文字列(半角英数)
<b>戻り値</b>	ライセンスキーが正しければ 0、正しくなければ 1 が返る。 不正なキーでも 0 を返す場合がある。このとき制限は解除されない。

**AquesTalk\_SetUsrKey**

AquesTalk.h

<b>説明</b>	使用ライセンスキーを設定。音声波形を生成する前に一度呼び出すことで、以降、合成音声データに含まれる透かしが使用ライセンス無しから取得済みに変化する。
<b>構文</b>	<code>int AquesTalk_SetUsrKey(const char *key)</code>
<b>引数</b>	
<i>key</i>	使用ライセンスキー、または頒布ライセンスキーの文字列(半角英数)
<b>戻り値</b>	ライセンスキーが正しければ 0、正しくなければ 1 が返る。 不正なキーでも、まれに 0 を返す場合がある。このときはライセンス無しのままである。

## 音声記号列

AquesTalk10 は、かな表記の音声記号列から音声を合成します。漢字を含んだテキスト文字列から音声を合成するときは、別途、言語処理ライブラリ AqKanji2Koe を用いて漢字仮名交じりテキストから音声記号列に変換する必要があります。

音声記号列の詳細は、付属の音声記号列仕様書を参照してください。

## 声質パラメータ

AquesTalk10 は声質パラメータの値を変更することで、様々な声種で合成できます。声質パラメータの種類と効果を以下に示します。

<b>bas</b>	声色のベースとなる基本素片 F1E,F2E,M1E のいずれかを指定
<b>spd</b>	話速。値が大きいほど発話速度が速い
<b>vol</b>	音量。デフォルトは 100。値が大きいほど音量が大きくなる。100以下を指定の場合は、比例して音量が変化する。100以上を指定した場合は、コンプレッサーが機能する。
<b>pit</b>	声の高さがパラメータに比例する。
<b>acc</b>	アクセントの強さ。値が大きいほどアクセントによるピッチの高低が大きくなる。
<b>lmd</b>	主に声質の高低を表現するが、より複雑な声質の変化がある。
<b>fsc</b>	声質の高低を表現。デフォルトは 100。サンプリング周波数を変化するだけなので、これに応じて話速や声の高さも同時に変化する。

AquesTalk.h には、声質パラメータの値をセットしたプリセット声種を複数用意しています。簡単に使う場合は、これを AquesTalk\_Synthe() の声質パラメータの引数にすることができます。

## エラーコード表

関数が返すエラーコードの内容は、次の通りです。

値	内容
100	その他のエラー
101	メモリ不足
103	音声記号列指定エラー(語頭の長音、促音の連続など)
104	音声記号列に有効な読みがない
105	音声記号列に未定義の読み記号が指定された
106	音声記号列のタグの指定が正しくない

107	タグの長さが制限を越えている(または[>]が見つからない)
108	タグ内の値の指定が正しくない
120	音声記号列が長すぎる
121	1つのフレーズ中の読み記号が多すぎる
122	音声記号列が長い(内部バッファオーバー1)

## サンプルプログラム

次に示すコードは、音声記号列から音声データを生成し、WAV ファイルとして出力する最も単純なプログラムです(同等のソースが、パッケージに含まれています。samples/HelloTalk.cpp)。

7 行目ではローカル変数 `voice` に、プリセット声種 `g_VoiceF1` をコピーして、8 行目で話速を 120 と少し早めに設定しています。

11 行目の"こんにちわ。" の部分を、任意の音声記号列に変更することで、異なるメッセージを生成可能です。

```
#include <stdio.h>
#include <AquesTalk.h> // AquesTalk クラスのヘッダ

int main(int ac, char **av)
{
    // 声質パラメータ
    AQT_VOICE voice = gVoice_F1;
    voice.spd = 120;

    int size;
    unsigned char *wav = AquesTalk_Synthe_Utf8(&voice, "こんにちわ。", &size);
    if(wav==0) {
        fprintf(stderr, "ERR %d", size); // エラー時は size にエラーコードが返る
        return -1;
    }

    // 生成した音声データをファイルに保存
    FILE *fp = fopen("ZZZ.wav", "w");
    fwrite(wav, 1, size, fp);
    fclose(fp);

    // Synthe()で生成した音声データは、使用後に呼び出し側で解放する
    AquesTalk_FreeWave (wav);

    return 0;
}
```

## アプリ開発ガイドライン

アプリケーションの開発(評価での使用を除く)は、以下のガイドラインに従ってください。

### ライセンスキー

本ライブラリの動作は、開発ライセンスキーと使用ライセンスキー、頒布ライセンスキーの3種類の関連キーに依存します。これらのキーは、各ライセンス購入時に発行されるライセンス証に記載されています。

`AquesTalk_SetDevKey()` をアプリケーションの起動初期に一度呼び出します。引数には開発ライセンスキーを指定します。これにより製品版として動作し、評価版の制限がなくなります。

`AquesTalk_SetUsrKey()` をアプリケーションの起動初期に一度呼び出します。`AquesTalk_SetDevKey()` との呼び出し順序は任意です。引数には、使用ライセンスキー、または頒布ライセンスキーを指定します。この指定により、合成音声データに含まれる透かしが、使用ライセンス無しの状態から取得済みに変化します。この変化による聴感上の違いはありません。

頒布ライセンスによりアプリを配布する場合は、頒布ライセンスキーを指定して呼び出します。

それ以外の場合は、エンドユーザが使用ライセンスキーを指定できるようにします。なお、エンドユーザが個人かつ非営利の利用の場合は使用ライセンスが不要なので、使用ライセンスキーが未指定の場合は、この関数の呼び出しをスキップして構いません。

関数の戻り値のチェックは必ず行い、エラーの場合はエンドユーザにその旨を通知してください。

### 声質パラメータ

エンドユーザによる声質パラメータの変更機能の有無は任意です。

変更可能にする場合、アプリケーション間での声質パラメータの値を共有するために、声質パラメータの値を写像しないでください。

例えば、話速のパラメータの範囲は 50 から 300 ですが、これを 0 から 100 の範囲にマッピングしてエンドユーザに提示しないでください。

エンドユーザに提示する各パラメータの名称には、`AquesTalk.h` に記載の漢字名称(話速など)や、3文字の構造体変数名(`spd` など)を用いるのが望ましいです。

## 文書履歴

日付	版	変更箇所	更新内容	更新者
2017/10/12	1.0		新規作成	N.Y