

AquesTalk Mac マニュアル

2010/1/6 (株)アクエスト

<http://www.a-quest.com/>

1. 概要

本文書は、規則音声合成ライブラリ AquesTalk Mac(以下、「AquesTalk」)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に関して、方法および注意点を示したものです。

AquesTalk Mac は Win 版の AquesTalk とは異なり、DA(サウンド出力)機能はありません。本ライブラリを用いて生成した音声波形データは、ファイルに出力、サウンドデバイスへ出力、またはネットワークを通じて転送するなど、アプリケーションの要求に応じた処理を別途実装する必要があります*¹。なお、生成する波形データのフォーマットは WAV 形式です。

AquesTalk の入力である音声記号列の文字コードは ShiftJIS となっています。実装環境によっては文字コード変換が必要になります。

なお、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、事前にパッケージに同梱の「ソフトウェア使用許諾契約書」(licence_aqtk_mac.txt)をご確認ください。配布にはライセンス契約が必要です。

*¹ AquesTalk の出力する音声波形データを Mac 本体から出力するためのオプションライブラリもあります。お問い合わせください。

2. 仕様

- OS Mac OS X 10.4 以降
- CPU Intel / PowerPC (Universal Binary)
- 入力 かな表記音声記号列 (ShiftJIS)
- 出力 WAV フォーマット(8KHz サンプリング, 16bitPCM, モノラル)
- 声種 女声 / 男声 2種切替
- 形式 Mac framework (ダイナミックライブラリ*)
- * リンク時、および実行時、共に本ライブラリが必要になります。

3. 関数 API

AquesTalk_Synthe

AquesTalkF.h

説明	音声記号列から音声波形を生成します 生成した音声データは、使用后、呼び出し側で AquesTalk_FreeWave を呼び出して開放してください。
構文	unsigned char * AquesTalk_Synthe (const char *koe, int iSpeed, int *size)
引数	
koe	音声記号列(NULL 終端 Shift JIS)
iSpeed	発話速度[%] 50-300 の間で指定 デフォルト:100 値を大きく設定するほど、速くなる
size	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 フォーマットのデータ(内部で領域確保、解放は呼び出し側で AquesTalk_FreeWave()で行う)の先頭アドレスを返す。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_SyntheMV

AquesTalkF.h

説明	音声記号列から音声波形を生成します AquesTalk_Synthe()に声種切替を追加したもの
構文	unsigned char * AquesTalk_SyntheMV (int idVoice, const char *koe, int iSpeed, int *size)
引数	
<i>idVoice</i>	声種 (0:女声 f1, 1:男声 m1) 以下、AquesTalk_Synthe()と同じ

SetSpeed AquesTalk_FreeWave

AquesTalkF.h

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

4. 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
102	音声記号列に未定義の読み記号が指定された
103	韻律データの時間長がマイナスになっている
104	内部エラー(未定義の区切りコード検出)
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
109	WAVE 再生ができない(サウンドドライバ関連の問題)
110	WAVE 再生ができない(サウンドドライバ関連の問題 非同期再生)
111	発声すべきデータがない
200	音声記号列が長すぎる
201	1つのフレーズ中の読み記号が多すぎる
202	音声記号列が長い(内部バッファオーバー)
203	ヒープメモリ不足
204	音声記号列が長い(内部バッファオーバー)
205	ライセンスキーが正しくない。または、設定されていない。
上記以外	音声記号列エラー(音声記号列上でエラーの位置を返す)

5. 音声データ形式

本ライブラリで生成する音声データは、次の形式となります。

AquesTalk_Synthe()等で生成する音声データには、先頭部に WAV ヘッダが付与されています。ストレート PCM データが必要な場合は、別途ヘッダを除いて使用してください(先頭の44バイト)。ほかのフォーマットをご希望の場合はカスタマイズを承りますので、お問い合わせください。

サンプリング周波数	8KHz
量子化 bit 数	16bit
チャンネル数	モノラル
エンコード	リニア PCM
フォーマット	WAV 形式

6. 音声記号列

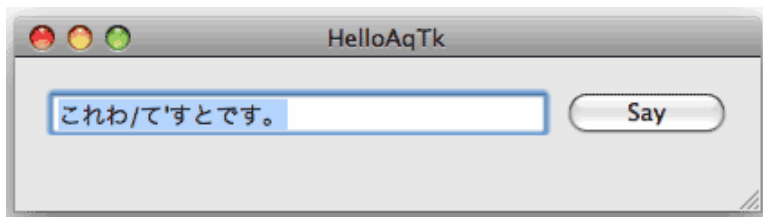
音声記号列の書き方、仕様については、

<http://www.a-quest.com/download/> の「音声記号列仕様」を参照ください。

7. サンプルプログラム

AquesTalk Mac ライブラリパッケージにサンプルプログラムのプロジェクト一式が入っています。

HelloAqTk は、任意の音声記号列指定して Say ボタンを押下すると、合成音声を発声するアプリケーションです。



7.1. ビルド方法

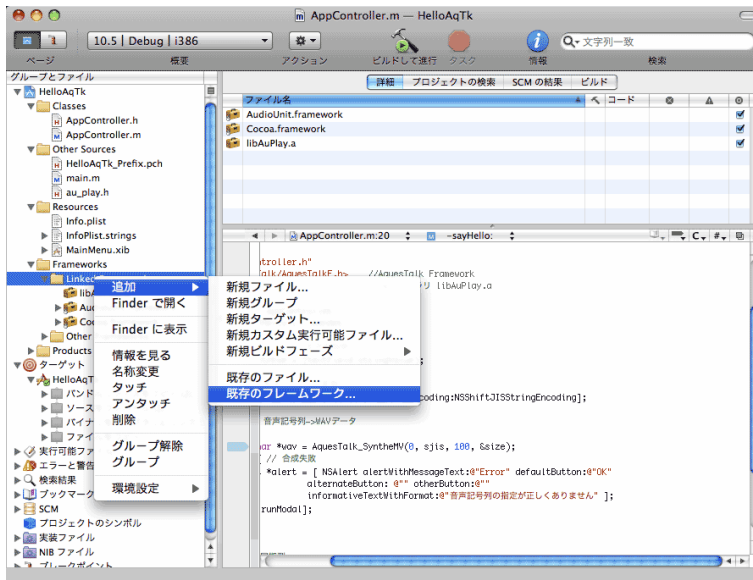
1. アプリケーションプロジェクトを開く

HelloAqTk.xcodeproj をダブルクリックして xcode でプロジェクトを開きます。

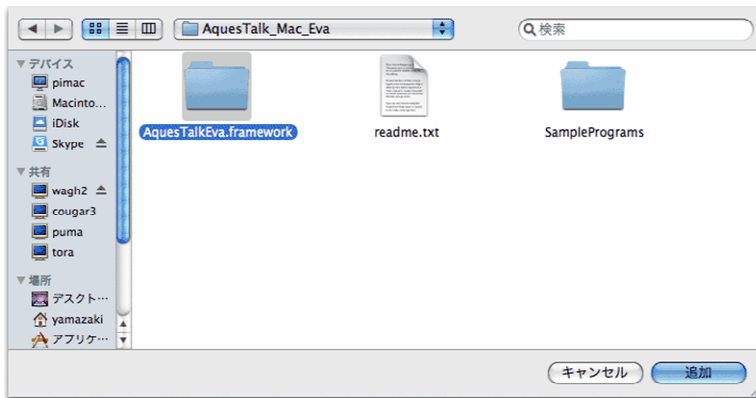
2. プロジェクトにフレームワークの追加

パッケージの初期状態では、AquesTalk フレームワークがプロジェクトに含まれていませんので追加します。

グループとファイルのコンテキストメニューの[Frameworks]/[Linked Frameworks]の右クリックから[追加]/[既存のフレームワーク]を選択します。



次に、AquesTalk パッケージ内の AquesTalk.framework を選択し、[追加]をクリックします。

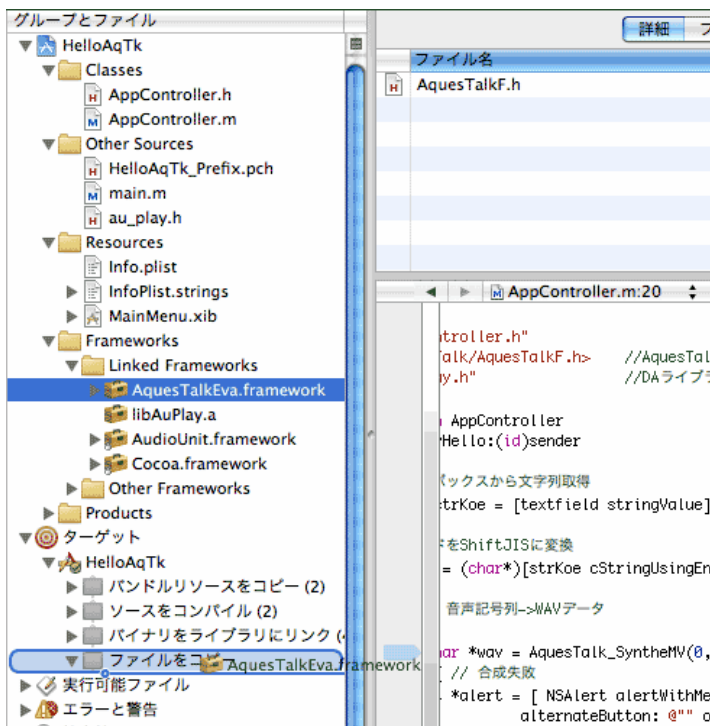


なお、次に表示されるコピー方法の指定のダイアログは、変更せずに[次へ]をクリックします。これで、[Linked Frameworks]に AquesTalk.framework が追加されたのが確認できます。

3. ターゲットにフレームワークを追加

AquesTalk framework は、動的ライブラリなので、実行時にも必要になります。そこで、実行モジュール HelloAqTk.app に配置されるように設定します。直接フレームワークをコピーしてもよいのですが、ここでは、ビルド時にコピーされる方法を用います。

グループとファイルのコンテキストメニューの[Linked Frameworks]の[AquesTalk.framework]を[ターゲット]/[HelloAqTk]/[ファイルをコピー]の部分にドラッグ&ドロップします。



4. ビルド

以上の方法で、フレームワークをプロジェクトに取り込むことができたので、ビルドができます。

ビルドでエラーが無ければ実行してみてください。

テキストボックスに任意の音声記号列を入力して[Say]ボタンのクリックで音声が入れば OK です。

7.2. コード説明

次に示すコードは、ボタンが押されたときに呼ばれる関数で、テキストボックスから文字列を取得し、AquesTalk で音声合成して、DA 出力する一連の処理が書かれています。

AquesTalk フレームワークの関数ヘッダをインポートします(2行目)。

AquesTalk Mac の文字コードは Shift JIS のため、文字コードを変換しています(12行目)。

関数 AquesTalk_SyntheMV() の最初の引数は、声種です。ここでは0の女声を指定しています。男声の場合は1を指定します(16行目)。

音声合成に成功すると戻り値に WAV 音声データの先頭アドレスが帰ります。指定した音声記号列が正しくない場合などのエラー時には0が返ります(17行目)。

生成した音声データを Mac 本体から再生します。ここでは、別途当社製の DA ライブラリ libAuPlay.a を使っています。このライブラリの入力は、16Bit/StraightPCM なので、WAV フォーマットを StraightPCM に変換してから与えるのですが、ここでは簡単に WAV ヘッダ(44バイト)の次のアドレスを指定しています。(27行目)。

音声出力の終了後、生成した WAV 波形データバッファを AquesTalk_FreeWave() で開放します(30行目)。

AppController.m

```
#import "AppController.h"
#import <AquesTalk/AquesTalkF.h> //AquesTalk Framework
#import "au_play.h" //DA ライブラリ libAuPlay.a

@implementation AppController
- (IBAction)sayHello:(id)sender
{
    // テキストボックスから文字列取得
    NSString *strKoe = [textfield stringValue];

    // 文字コードを ShiftJIS に変換
    char *sjis = (char*)[strKoe cStringUsingEncoding:NSUTF8StringEncoding];

    // 音声合成 音声記号列->WAV データ
    int size;
    unsigned char *wav = AquesTalk_SyntheMV(0, sjis, 100, &size);
    if(wav==0){ // 合成失敗
        UIAlertView *alert = [UIAlertView alertWithMessageText:@"Error" defaultButton:@"OK"
            alternateButton: @""" otherButton:@"""
            informativeTextWithFormat:@"音声記号列の指定が正しくありません"];

        [alert runModal];
        return;
    }

    // 音声出力 同期型
    // WAV ヘッダー(先頭の 44byte)を除いて、StraightPCM とする
    PlaySound((short*)(wav+44), (size-44)/2); // libAuPlay.a

    // 生成した波形バッファの解放
    AquesTalk_FreeWave(wav);
}

@end
```

8. 履歴

日付	版	変更箇所	更新内容	更新者
2007/1/7	1.0	新規作成	Win 版から加筆修正	N. Y
2010/1/6	2.3	Ver. up	ライセンスキー不要版 Hel loAqTk 説明追加	N. Y