AquesTalk for WinCE プログラミングガイド

(株)アクエスト

1. 概要

本文書は、規則音声合成ライブラリ AquesTalk for WinCE (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 には 2 種類のライブラリがあります。音声データをメモリ上に生成するものと、サウンドデバイスに出力する2種類があります。使用するアプリケーションに応じて選択してください。

最も簡単な使用方法は、次の1行のコードで実現できます（VC++）。

1. 音声を生成して、サウンドデバイスに出力します

```
AquesTalkDa_PlaySync_Utf16 ("こんにちわ。"); //< 引数に音声記号列の文字列を指定
```

2. ライブラリ構成

ご利用の OS に応じて 2 種類のライブラリが含まれています。

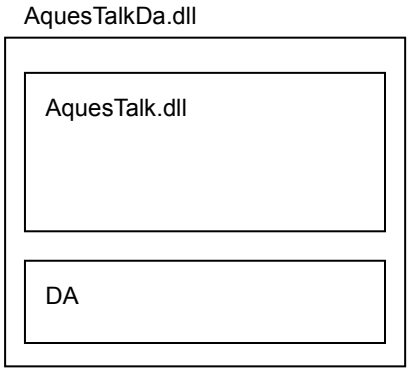
1. **PC2003**
   Windows Pocket PC 2003 上でご利用のときのライブラリが含まれています。

2. **WM5**
   Windows Mobile 5.0 用のライブラリが含まれています。Windows Mobile 6 でもご利用いただけます。

また、それぞれ次の 2 種類のライブラリがあります。必要に応じてどちらか、あるいは両方を選択して、アプリモジュールとおなじディレクトリにコピーして用いてください。

1. **AquesTalk.dll**
   音声記号列から音声データ(WAV フォーマット)を生成します。
   音声データをメモリ上に生成します。
   生成した音声データになんらかの処理を施す場合には、こちらを用います。

2. **AquesTalkDa.dll**
   AquesTalk.dll に、DA(サウンド出力機能)を含んだもので、音声記号列から音声を生成し、サウンドデバイスに出力します。
   AquesTalkDa.dll の実行に際し AquesTalk.dll は不要です。
   同期と非同期の 2 種類があります。同期タイプは発声を終了するまで関数から戻らないもので、非同期タイプは、発声の終了を待たずに関数から戻るもので、発声の終了はメッセージで通知することが可能です。



3. コンパイル･リンク

3.1. ヘッダ、ライブラリ

DLL を使用するには 対応するヘッダファイル(.h)をインクルードし、 リンク時に 対応する lib ファイルをリンクすることが必要です。各 DLL に対応するヘッダ等は下表を参照してください

DLL	ヘッダ	lib
AquesTalk.dll	AquesTalk.h	AquesTalk.lib
AquesTalkDa.dll	AquesTalkDa.h	AquesTalk.lib

4. 関数 API

4. 1. AquesTalk.dll

(※WindowCE では通常文字コードは UNICODE ですので、通常は AquesTalk_Synthe_Utf16()を用います)

AquesTalk_Synthe

AquesTalk.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、使用後、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 **シフト 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_Synthe_Euc

AquesTalk.h

説明 音声記号列から音声波形を生成します

AquesTalk_Synthe()の EUC 文字コード版

構文 unsigned char * **AquesTalk_Synthe_Euc**(const char **koe*, int *iSpeed*, int **size*)

引数

koe 音声記号列(NULL 終端 EUC)

以下、 AquesTalk_Synthe()と同じ

AquesTalk_Synthe_Utf8

AquesTalk.h

説明 音声記号列から音声波形を生成します

AquesTalk_Synthe()の UTF-8 文字コード版

構文 unsigned char * **AquesTalk_Synthe_Utf8**(const char **koe*, int *iSpeed*, int **size*)

引数

koe 音声記号列(NULL 終端 UTF-8 BOM は無し)

以下、 AquesTalk_Synthe()と同じ

AquesTalk_Synthe_Utf16

AquesTalk.h

説明 音声記号列から音声波形を生成します

	AquesTalk_Synthe()の UTF-16 文字コード版
構文	unsigned char * **AquesTalk_Synthe_Utf16**(const unsigned short *w*koe*, int *iSpeed*, int *size*)
引数	
wkoe	音声記号列(NULL 終端 UTF-16 BOM 指定は任意 エンディアンは実行環境に依存)
	以下、 AquesTalk_Synthe()と同じ

AquesTalk_Synthe_Roman

AquesTalk.h

説明	音声記号列から音声波形を生成します
	AquesTalk_Synthe()のローマ字(7bitASCII)文字コード版
構文	unsigned char * **AquesTalk_Synthe_Roman**(const char *koe, int *iSpeed*, int *size*)
引数	
koe	音声記号列(NULL 終端 ASCII 表記方法はホームページ上の音声記号列仕様を参照)
	以下、 AquesTalk_Synthe()と同じ

SetSpeed AquesTalk_FreeWave

AquesTalk.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
wav	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. 2. AquesTalkDa.dll

AquesTalkDa_PlaySync

AquesTalkDa.h

説明	同期タイプの音声合成。発声が終了するまで戻らない。
構文	int **AquesTalkDa_PlaySync**(const char *koe, int *iSpeed=100*)
引数	
koe	音声記号列(NULL 終端 **シフト 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
戻り値	0:正常終了 それ以外:エラー

AquesTalkDa_PlaySync_Euc

AquesTalkDa.h

説明	同期タイプの音声合成。発声が終了するまで戻らない。
構文	int **AquesTalkDa_PlaySync_Euc**(const char *koe, int *iSpeed=100*)
引数	
koe	音声記号列(NULL 終端 EUC)
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
戻り値	0:正常終了 それ以外:エラー

AquesTalkDa_PlaySync_Utf8

AquesTalkDa.h

説明 同期タイプの音声合成。発声が終了するまで戻らない。

構文 int **AquesTalkDa_PlaySync_Utf8**(const char **koe*, int *iSpeed=100*)

引数

koe 音声記号列(NULL 終端 UTF-8 BOM は無し)

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

戻り値 0:正常終了 それ以外:エラー

AquesTalkDa_PlaySync_Utf16

AquesTalkDa.h

説明 同期タイプの音声合成。発声が終了するまで戻らない。

構文 int **AquesTalkDa_PlaySync_Utf16**(const unsigned short **koe*, int *iSpeed=100*)

引数

koe 音声記号列(NULL 終端 UTF-16 BOM 指定は任意 エンディアンは実行環境に依存)

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

戻り値 0:正常終了 それ以外:エラー

AquesTalkDa_PlaySync_Roman

AquesTalkDa.h

説明 同期タイプの音声合成。発声が終了するまで戻らない。

構文 int **AquesTalkDa_PlaySync_Roman**(const char **koe*, int *iSpeed=100*)

引数

koe 音声記号列(NULL 終端 ASCII 表記方法はホームページ上の音声記号列仕様を参照)

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

戻り値 0:正常終了 それ以外:エラー

AquesTalkDa_Create

AquesTalkDa.h

説明 音声合成エンジンのインスタンスを生成(非同期タイプ)

構文 H_AQTKDA **AquesTalkDa_Create**()

引数

戻り値 音声合成エンジンのハンドル

AquesTalkDa_Release

AquesTalkDa.h

説明 音声合成エンジンのインスタンスを解放(非同期タイプ)

発声終了前にこの関数でインスタンス解放すると、その時点で発声が終了してしまうので注意

構文 void **AquesTalkDa_Release**(H_AQTKDA *hMe*)

引数

hMe 音声合成エンジンのハンドル

戻り値 なし

AquesTalkDa_Play

AquesTalkDa.h

説明 非同期タイプの音声合成。発声終了を待たずに戻る。

	発声終了時に hWnd に指定したウィンドウにメッセージが送出(post)される。 再生終了前に AquesTalkDa_Play()を再度呼び出すことで、連続再生可能。また、このとき、hWnd 等を変更して異なるメッセージを設定することも可能
構文	int **AquesTalkDa_Play**(H_AQTKDA *hMe,* const char *koe, int *iSpeed=100*, HWND *hWnd*=0, unsigned long *msg*=0, unsigned long dwUser=0)
引数	
hMe	音声合成エンジンのハンドル
koe	音声記号列
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
hWnd	発声終了時のメッセージの送り先 Window ハンドル(NULL を指定すると終了メッセージは送られない)
msg	発声終了時のメッセージ ID を指定する。hWnd=NULL の時は無効
dwUser	任意。発生終了時のメッセージの lParam(第 2 引数)に渡される
戻り値	0:正常終了 それ以外:エラー

AquesTalkDa_Stop

AquesTalkDa.h

説明	発声の停止。Play()で発声中に使用する。 Stop()によって発声が終了した場合も、Play()で *hWnd* が指定されていたならメッセージが送出される。
構文	void **AquesTalkDa_Stop**(H_AQTKDA *hMe*)
引数	
hMe	音声合成エンジンのハンドル
戻り値	なし

AquesTalkDa_IsPlay

AquesTalkDa.h

説明	再生中か否か
構文	int **AquesTalkDa_IsPlay**(H_AQTKDA *hMe*)
引数	
hMe	音声合成エンジンのハンドル
戻り値	1:再生中 0:再生中でない

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()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。
また、AquesTalkDa.dll では、お使いの環境のサウンドドライバが以下の形式の音声を再生できる必要があります。

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. サンプルコード

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(HelloTalk.cpp)。
12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、異なるメッセーを生成可能です。
なお、このプログラムで出力した WAV ファイルは、メディアプレイヤーなどで再生することができます。
非同期に音声出力を行う、再生を停止する、発声速度を変更するなどのより高度なプログラミング方法は、付属の MFC アプリ AqTkApp のソースコードを参考にしてください。
AquesTalk　http://www.a-quest.com/aquestalk/

```
#include "stdafx.h"
#include "AquesTalk.h"    //←このソースと同じところにコピーしておく


int _tmain(int argc, _TCHAR* argv[])
{
    int iret;
    const _TCHAR *koe = _T("こんにちわ。");//発声させる音声記号列を指定


    // 音声合成(UNICODE の音声記号列を、音声波形データ(wav フォーマット)に変換)
    int size;
    unsigned char *wav = AquesTalk_Synthe_Utf16((const unsigned short*)koe, 100, &size);
    if(wav==0){ // エラー時は、size にエラーコードがセットされ、0が返る
        fwprintf(stderr, _T("ERR:Synthe():%d¥n"),size);
        return -1;
    }

    // サウンド出力(メモリ上のデータを同期再生)
    iret = PlaySound((LPCWSTR)wav, NULL, SND_MEMORY|SND_SYNC);
    if(iret==0){
        fwprintf(stderr, _T("ERR:PlaySound()¥n"));
        return -1;
    }

    // 波形バッファの開放
    AquesTalk_FreeWave(wav);

    return 0;
}
```

7. 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[>]がみつからない)

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1）
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1）
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）

8. 履歴

日付	版	変更箇所	更新内容	更新者
2008/09/12	1.0	新規作成	新規 （Win,Linux 版のマージ、修正）	N.Y