

AquesTalk2 Win 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 AquesTalk2 Win 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。

AquesTalk2 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。

AquesTalk2 は、声質ファイル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。

本ライブラリの評価版は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	DLL(ダイナミックライブラリ) i386/x64
対応 OS	Windows 11/10/8/7/Vista/XP, 32bit/64bit 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 (8KHz サンプリング, 16bitPCM, モノラル) データ
声種	14 種
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 130KByte

ビルド・実行

ヘッダ、ライブラリ

プログラムのコンパイル時にはヘッダファイル(AquesTalk2.h)をインクルードします。必要に応じてインクルードファイルのパスを設定しておきます。リンク時には、AquesTalk2.lib ファイルをリンクするファイルに追加しておきます。なお、LoadLibrary()などで実行時に動的にリンクすることも可能です。

パッケージ内の lib フォルダが 32bit 版、lib64 フォルダが 64bit 版となります。

実行

アプリの実行の際には、AquesTalk2.dll が必要です。通常、このファイルはアプリの exe ファイルと同じディレクトリに配置します。

依存ライブラリ

AquesTalk2.dll は、コンパイルオプション/MT でビルドしています。したがって、AquesTalk2.dll が使用する C ランタイム ライブラリ (CRT) は DLL 内に含まれています (静的リンク)。実行時に必要なライブラリは KERNEL32.dll だけで、これは Windows のシステムディレクトリに含まれています。アプリの開発において依存ライブラリを意識する必要はありません。

AquesTalk2_Synthe()関数で返された音声データは、使用後に AquesTalk2_FreeWav()で解放してください。アプリ側で free() などを使用して解放すると、異なる CRT の場合にハングします。

関数 API

AquesTalk2_Synthe

AquesTalk2.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 * AquesTalk2_Synthe (const char *koe, int *size ,void *phontDat=0)
引数	
koe	音声記号列 (SJIS NULL 終端) を指定
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)
phontDat	phont データの先頭アドレスを指定します。 この DLL のデフォルト Phont を用いるときは0を指定します。
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 AquesTalk2_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。

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 * AquesTalk2_Synthe_Utf8 (const char *koe, int *size, void *phontDat=0)
引数	
<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)
<i>phontDat</i>	phont データの先頭アドレスを指定します。 この DLL のデフォルト Phont を用いるときは0を指定します。
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 AquesTalk2_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。

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 * AquesTalk2_Synthe_Utf16 (const unsigned short *koe, int *size, void *phontDat=0)
引数	
<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)
<i>phontDat</i>	phont データの先頭アドレスを指定します。 この DLL のデフォルト Phont を用いるときは0を指定します。
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 AquesTalk2_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。

説明	音声データの領域を開放
構文	void AquesTalk2_FreeWave (unsigned char *wav)
引数	
<i>wav</i>	WAV フォーマットのデータ(AquesTalk2_Synthe()等で生成した音声データ)

戻り値

なし

音声記号列

AquesTalk2 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。

PHONT ファイル

AquesTalk2 は声質データ（Phont データ）を指定することで、様々な声種で合成できます。各 Phont データは、パッケージの **phont** フォルダにある ***.phont** ファイルです。音声合成の実行時にこのファイルを読み込んで(バイナリ)、その先頭アドレスを指定します。なお、Phont データを指定しない場合のデフォルトは、**ar_f4.phont** と同じです。

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
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
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)
1000-1008	Phont データが正しくない

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/HelloTalk.cpp)。

9 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 <AquesTalk2.h>
#pragma comment(lib, "AquesTalk2.lib")

int main(int ac, char **av)
{
    int size;
    unsigned char *wav = AquesTalk2_Synthe("こんにちわ。", 100, &size, 0); // Phont:Default
    if(wav==0){
        fprintf(stderr, "ERR:%d\n",size);
        return -1;
    }

    FILE *fp=fopen("\\zzz.wav", "wb");
    fwrite(wav, 1, size, fp);
    fclose(fp);

    AquesTalk2_FreeWave(wav);

    return 0;
}
```

次に示すコードは、先のコードに外部の Phont ファイルを指定できるようにしたものです。Phont ファイルをバイナリで読み込んで、先頭アドレスを合成時に指定しています。

```
#include <stdio.h>
#include <memory.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <AquesTalk2.h>
void * file_load(const char * file, int * psize);
int main(int ac, char **av)
```

```

{
int size;
// Phont ファイルの読み込み
int size;
void *pPhont = file_load("¥¥aq_f1b.phont", &size); // Phont ファイルをここで指定
if(pPhont ==0) return-2;
// メモリ上に音声データを生成
unsigned char *wav = AquesTalk2_Synthe("こんにちわ。", 100, &size, pPhont);
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;
}
// Phont データの開放（音声合成が終わったら開放できる）
free(pPhont);
// ルートディレクトリに生成した音声データを保存
FILE *fp = fopen("¥¥ZZZ.wav", "wb");
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
AquesTalk2_FreeWave (wav);
return 0;
}
// ファイルの読み込み
void * file_load(const char * file, int * psize)
{
FILE *fp;
char *data;
struct _stat st;

*psize = 0;
if( _stat(file, &st)!=0) return NULL;

if((data=(char *)malloc(st.st_size))==NULL){
    fprintf(stderr, "can not alloc memory(file_load)¥n");
    return NULL;
}
if((fp=fopen(file, "rb"))==NULL) {
    free(data);
    perror(file);
    return NULL;
}
if(fread(data, 1, st.st_size, fp)<(unsigned)st.st_size) {
    fprintf(stderr, "can not read data (file_load)¥n");
    free(data);
    fclose(fp);
    return NULL;
}
fclose(fp);
*psize = st.st_size;
return data;
}

```

C や C++ 以外の言語環境(たとえば C#, VB など) での使用方法はここでは示しません。データ型の変換等の処理が必要になりますが、各言語の DLL を呼び出す一般的な方法を参照してください。なお、AquesTalk2_Synthe()関数はポインタを返しますので、VB ではラッパーが必要になるかもしれません。

文書履歴

日付	版	変更箇所	更新内容	更新者
2009/12/26	1.0		新規作成	N.Y
2011/01/28	2.1		パッケージ同梱用に修正	N.Y
2022/08/16	2.4		関数追加	N.Y