

AquesTalk10 Mac 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 Mac 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。

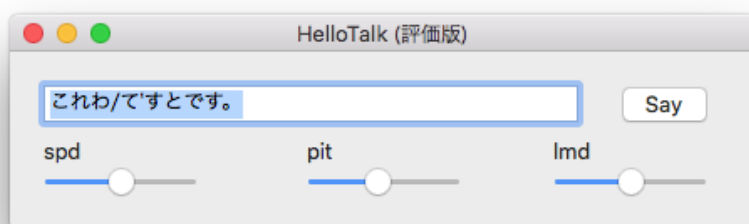
仕様

ライブラリ形式	macOS framework(共有ライブラリ dylib)
対応 OS	Mac OS X 10.3 以降 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 __stdcall
マルチスレッド	対応
ライブラリサイズ	約 650KByte (i386/x86_64 universal binary)
外部依存ライブラリ	libc++.so.1

サンプルプログラム

AquesTalk10 Mac のライブラリは共有ライブラリです。ビルド（リンク）時だけでなく、実行時にも本ライブラリが必要です。アプリケーションに組み込む際のポイントは、AquesTalk.framework をプロジェクトに適切にインポートすることと、アプリ実行モジュール(app)内の適切な位置に AquesTalk.framework をコピーすることです。

以下に、パッケージ内のサンプルプログラム HelloTalk をつかって、framework のインポートから実行までの手順を示します。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。開発言語は Objective-C、動作環境は macOS 10.11、xcode 8.2.1 で確認しています。なお、パッケージ内にビルド済みの HelloTalk.app も含まれています。



ビルド

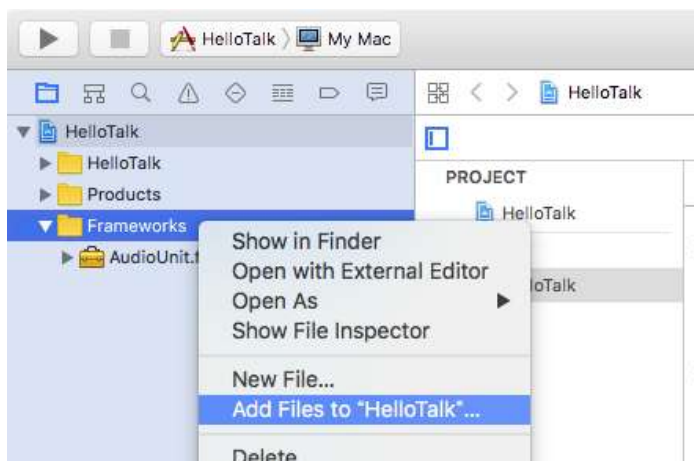
1. アプリケーションプロジェクトを開く

samples/HelloTalk/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。

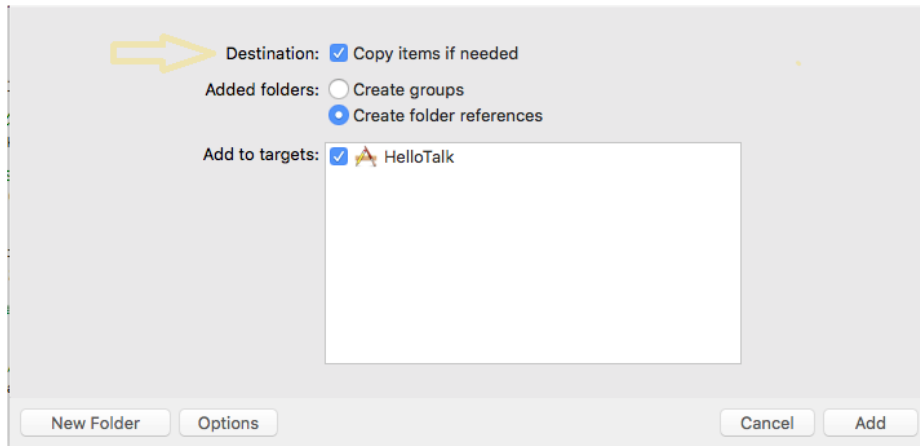
2. プロジェクトにフレームワークの追加

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。

ファイル一覧の [Frameworks] を右クリックし、[Add Filesto "HelloTalk"...] を選択します。



ここに加えるファイルとして、AquesTalk10 パッケージ内の AquesTalk.framework を選択します。
[Options]をクリックし、[Destination]の[Copy items if needed]をチェックして、AquesTalk.framework
をプロジェクトフォルダにコピーするようにします。その後に、[Add]をクリックします。

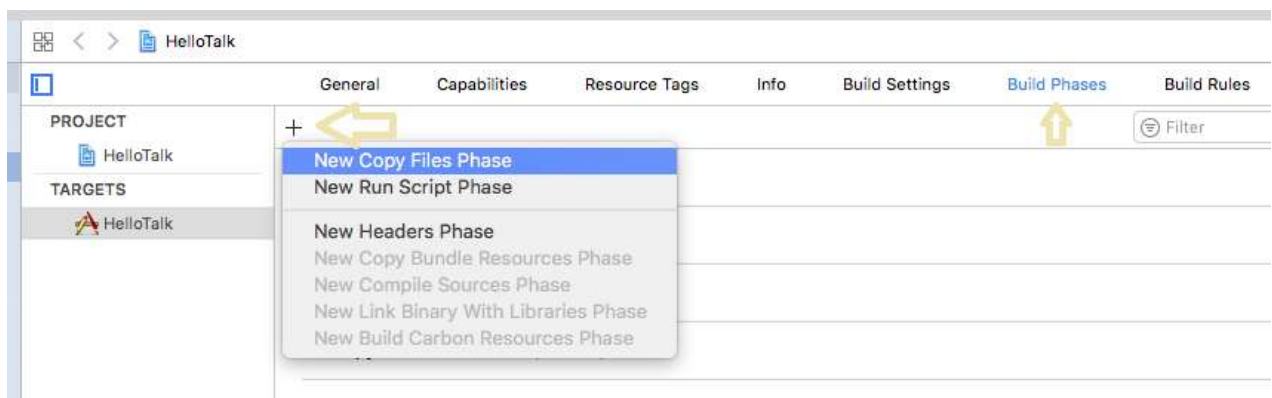


ファイル一覧の[Frameworks]に AquesTalk.framework が追加されたことを確認します。

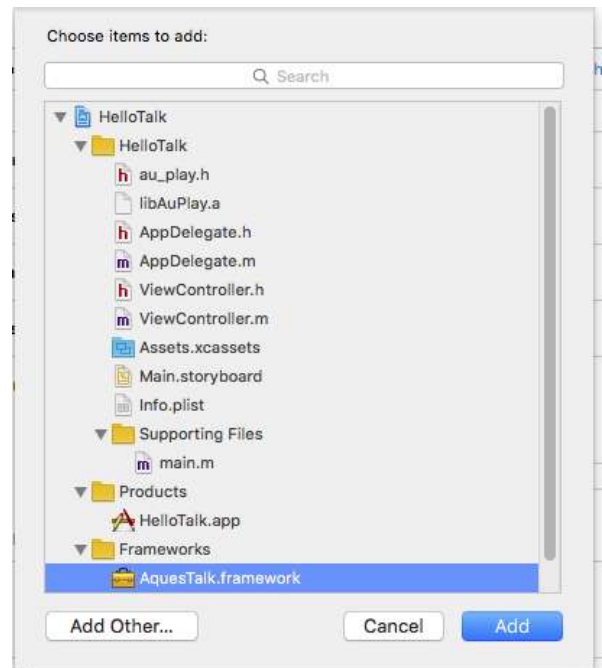
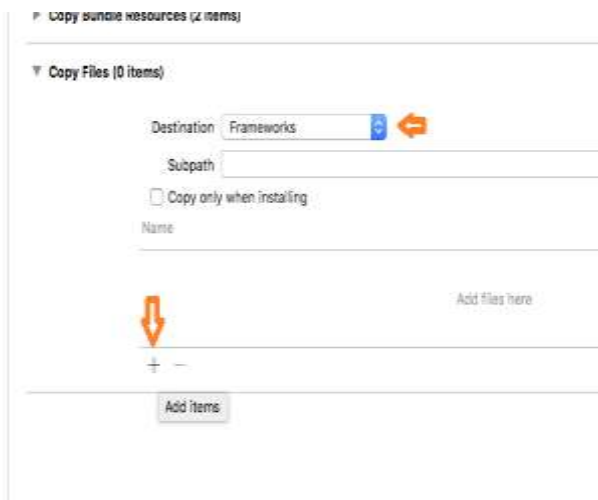
ターゲットにフレームワークをコピーする設定

AquesTalk framework は、ダイナミックライブラリなので、実行時にも必要です。そこで、ビルド時に実行モジュール HelloTalk.app 内にこれが配置されるように設定します。なお、ビルド後に手動で framework をコピーする方法もあります。

[Build Phases]タブ内の[+]をクリックして[New Copy Phase]を選択します。

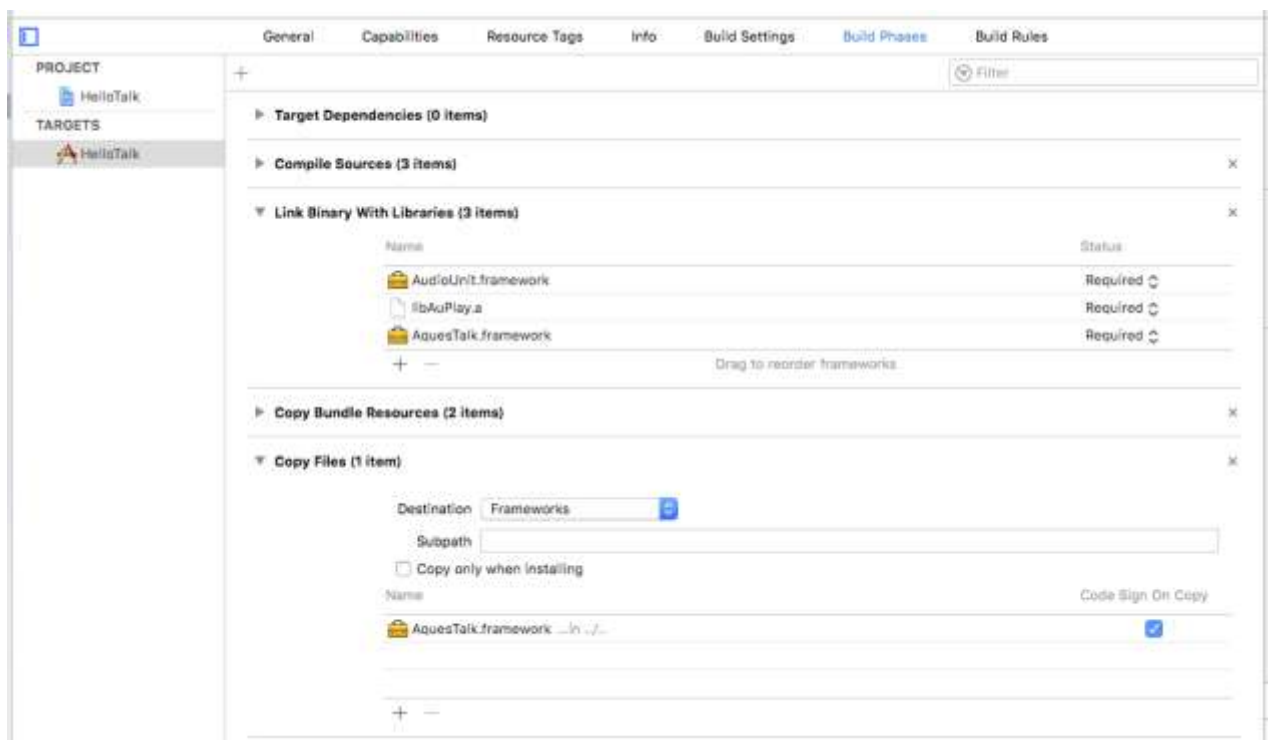


[Copy Files]ができるので、[Destination]に[Frameworks]を選択して、[+]をクリックします。



AquesTalk.framework を選択して[Add]します。

以下のようにになっているはずです。これで、ビルド時に AquesTalk.framework が app 内にコピーされます。



ビルド・実行

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、ビルドします。メニューから[Run]を行い、ビルド・実行してみてください。エラーが無ければ HelloTalk のウィンドウが開くので、テキストボックスに任意の音声記号列を入力して [Say] ボタンのクリックします。音声聞こえれば OK です。

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。

コード説明

次に示すコードは、HelloTalk の ViewController.m です。[Say] 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、AquesTalk10 で音声合成して、音声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)。
テキストボックスから取得した文字列を Shift JIS へ文字コードを変換しています (13 行目)。
AquesTalk には、ShiftJIS 以外に、UTF-8、UTF16 の文字コードの関数も用意されています。

変数 voice に声質パラメータを設定しています。この例では、プリセット声種[F1]の値をコピーしてから(16 行目)、スライダから取得した値を話速、ピッチ、音程 1 の各声質パラメータにセットしています(17-19 行目)。これによりスライダの操作で声質を変えられます。声質パラメータは他にもあり、適宜追加できます。動作の詳細は「声質パラメータ」の章を参照ください。

関数 AquesTalk_Synthe の最初の引数は、声質パラメータの構造体のアドレスをしています。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 (23 行目)。

生成した音声データを Mac 本体から再生します (32 行目)。ここでは、別途当社製の音声出力ライブラリ libAuPlay.a を使っています(AudioUnit.framework を使用)。このライブラリは WAV データを与えると同期的に再生出力するものです。このライブラリは AquesTalk の音声データ用であり、すべての WAV データに対応しているわけではありません。

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 (35 行目)。

```
1      #import "ViewController.h"
2      #import <AquesTalk/AquesTalk.h> //AquesTalk10 Framework
3      #import "au_play.h"           //DA ライブラリ libAuPlay.a
4
5      @implementation ViewController
6
7      - (IBAction)sayHello:(id)sender
8      {
9          // テキストボックスから文字列取得
10         NSString *strKoe = [textfield stringValue];
11
12         // 文字コードを ShiftJIS に変換
13         char *sjis = (char*)[strKoe cStringUsingEncoding:NSUTF8StringEncoding];
14
```

```

15 // 声質設定
16 AQTK_VOICE voice =gVoice_F1; //プリセット声種の F1 をベースにする
17 voice.spd = (int)[spd integerValue]; // スライダーの値から話速をセット
18 voice.pit = (int)[pit integerValue]; // 高さ
19 voice.lmd = (int)[lmd integerValue]; // 音程 1
20
21 // 音声合成 音声記号列->WAV データ
22 int size;
23 unsigned char *wav = AquesTalk_Synthe(&voice, sjis, &size);
24 if(wav==0){ // 合成失敗
25     UIAlertView *alert = [[ UIAlertView alloc] init ];
26     [ alert setMessageText:@"音声記号列の指定が正しくありません" ];
27     [alert runModal];
28     return;
29 }
30
31 // 音声出力 同期型
32 PlaySound(wav); // libAuPlay.a
33
34 // 生成した波形バッファの解放
35 AquesTalk_FreeWave(wav);
36 }
37
38 @end

```

プログラミングの注意

AquesTalk_Synthe()関数で返された音声データは、使用後に AquesTalk_FreeWave()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 * AquesTalk_Synthe (const AQTK_VOICE *pParam, const char *koe, int *size)
引数	
pParam	声質パラメータ(構造体 AQTK_VOICE のアドレス)を指定
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)
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 * AquesTalk_Synthe_Utf8 (const AQTK_VOICE *pParam, const char *koe, int *size)
引数	
pParam	声質パラメータ(構造体 AQTK_VOICE のアドレス)を指定
koe	音声記号列(UTF-8 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)
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 * AquesTalk_Synthe_Utf16 (const AQTK_VOICE *pParam, const unsigned short *koe, int *size)
引数	
pParam	声質パラメータ(構造体 AQTK_VOICE のアドレス)を指定
koe	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
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_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	
wav	WAV フォーマットのデータ(AquesTalk_Synthe()等で生成した音声データ)
戻り値	なし

説明	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。
構文	int AquesTalk_SetDevKey (const char *key)
引数	
key	開発ライセンスキー文字列(半角英数)
戻り値	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。

説明	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。
構文	int AquesTalk_SetUsrKey (const char *key)
引数	
key	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)
戻り値	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みである。

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。

変数名	範囲	デフォルト	説明
bas	0 - 2	-	声色のベースとなる基本素片 F1E,F2E,M1E のいずれかを指定
spd	50 - 300	100	話速。値が大きいほど発話速度が速い。
vol	0 - 300	100	音量。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。
pit	20 - 200	-	声の高さがパラメータに比例する。デフォルトは基本素片毎に異なる。
acc	0 - 200	-	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。デフォルトは基本素片毎に異なる。
lmd	0 - 200	100	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。
fsc	50 - 200	100	声質の高低を表現。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_Synthe0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)

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
2017/10/14	1.0		新規作成	N.Y