AqKanji2Koe 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、言語処理ライブラリ AqKanji2Koe 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。

AqKanji2Koe は、漢字かな混じり文のテキスト情報を AquesTalk 用のアクセント付きの音声記号列に変換する macOS 用のライブラリ(Framework)です。

このライブラリと音声合成ライブラリ AquesTalk を使えば、動的に変化する様々なテキストから、リアルタイムに音声メッセージを生成できるようになります。

特長
・簡単に組み込み可能
　- テキスト文字列を入力すると音声記号列を返す、シンプルな API
・高速な変換処理
　- 約 20 万字/秒の高速な変換処理
・登録単語の追加が可能
　- ユーザ辞書ライブリを用いて、アプリで単語の追加や、編集、削除が可能
・高精度な読み・アクセント付与
　- 最大 50 万語の単語辞書とアクセントルールにより、正確な読みとアクセントを生成

本ライブラリを使ったプログラミングに先立って動作を確認される場合は、弊社サイトのオンラインデモや、言語処理エンジンのページからデモアプリ AqK2kDemo をダウンロードしてお試しください。漢字を含んだ文を音声記号列に変換や、ユーザ辞書を編集する機能を確認できます。

オンラインデモ	https://www.a-quest.com/demo/
デモアプリ ダウンロード	https://www.a-quest.com/products/aqkanji2koe.html

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」は、すべて「ヌ」と出力されます**

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。

本文書は、言語処理ライブラリ AqKanji2Koe 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に関しての方法および注意点を示したものです。

仕様

ライブラリ形式	macOS 用 framework(ダイナミックリンクライブラリ) 32bit / 64bit
対応 OS	Mac OS X 10.3 以降
入力データ形式	漢字かな混じり文テキスト(UTF8)
出力データ形式	かな表記音声記号列(UTF8)
CPU	32bit または 64bit (i386/x84_64 Universal Binary)
ビルド環境	Xcode 9.0, LLVM 9.0
ライブラリサイズ	約 150KByte
辞書サイズ	標準:約7MB(約36万語)、スモール:約5MB(約25万語) ラージ:約13MB(約50万語)
依存ライブラリ	libc++.1.dylib, libSystem.B.dylib
その他	ユーザ辞書用ライブラリ(AqUsrDic) 同梱

ビルド・実行

ヘッダ、ライブラリ

アプリ作成の際には、プロジェクトに AqKanji2Koe.framework を追加します。また、ソースコードにヘッダファイル(AqKanji2Koe.h)をインポートします。

実行

アプリの実行の際には、AqKanji2Koe.framework のファイルが必要です。アプリをビルドしたときに、これらの framework がアプリに含まれるようにします。

また、実行時には辞書ファイルも必要です。必要に応じて、3 種類のサイズの中から選択して、アプリに含めるようにします。

ユーザ辞書ライブラリ

ユーザ辞書に単語の追加や削除、編集を行う場合には、別途、ユーザ辞書ライブラリ AqUsrDic,framework を用います。アプリに組み込む際は、AqKanji2Ko.framework と同様の方法で追加します。

関数 API

AqKanji2Koe.framework

AqKanji2Koe_Create　　AqKanji2Koe.h

説明　言語処理モジュールのインスタンス生成と内部データの初期化
生成したインスタンスは、使用後、AqKanji2Koe_Release で解放してください。

構文　void * **AqKanji2Koe_Create** (const char **pathDic*, int **pErr*)

引数

pathDic　辞書のディレクトリを指定。通常、"<app dir>/aq_dic"。指定した内容は内部で保存される。
文字列最後の、/　の付与は任意

pErr　エラー時にはエラーコードが入る 正常終了時は不定値

戻り値　インスタンスハンドル エラーの時は0が返る。このとき pErr にエラーコードが設定される。

AqKanji2Koe_Create_Ptr　　AqKanji2Koe.h

説明　言語処理モジュールのインスタンス生成と内部データの初期化
生成したインスタンスは、使用後、AqKanji2Koe_Release で解放してください。

構文　void * **AqKanji2Koe_Create_Ptr** (const void **pSysDic*, const void **pUserDic*, int **pErr*)

引数

pSysDic　システム辞書(通常 aqdic.bin) をメモリ上に読み込んだ先頭アドレスを指定

pUserDic　ユーザ辞書(通常 aq_user.dic) をメモリ上に読み込んだ先頭アドレスを指定
ユーザ辞書を使用しない場合は NULL を指定する

pErr　エラー時にはエラーコードが入る 正常終了時は不定値

戻り値　インスタンスハンドル エラーの時は0が返る。このとき pErr にエラーコードが設定される。

AqKanji2Koe_Release　　AqKanji2Koe.h

説明　言語処理モジュールのインスタンスを開放

構文　void **AqKanji2Koe_Release** (void * *hAqKanji2Koe*)

引数

hAqKanji2Koe　AqKanji2Koe_Create() / AqKanji2Koe_Create_Ptr()で返されたハンドルを指定

戻り値　なし

AqKanji2Koe_Convert　AqKanji2Koe.h

説明　漢字かな混じりのテキストを音声記号列に変換(UTF8 版)

構文　int **AqKanji2Koe_Convert** (void * *hAqKanji2Koe*, const char **kanji*, char **koe*, int *nBufKoe*)

引数

hAqKanji2Koe　AqKanji2Koe_Create() / AqKanji2Koe_Create_Ptr()で返されたハンドルを指定

kanji　入力漢字かな混じり文テキスト文字列（UTF8, NULL 終端)

koe　出力バッファ。音声記号列文字列が返る（UTF8, NULL 終端)

nBufKoe　koe バッファのサイズ[byte] 256 以上を指定。バッファサイズ以上の音声記号列は切り捨てられますので入力テキストの数倍のサイズを指定することを推薦。

戻り値　0:正常終了　それ以外：エラーコード

AqKanji2Koe_ConvertW　AqKanji2Koe.h

説明　漢字かな混じりのテキストを音声記号列に変換(UTF32 版）

構文　int **AqKanji2Koe_Convert** (void * *hAqKanji2Koe*, const wchar_t **kanji*, wchar_t **koe*, int *nBufKoe*)

引数

hAqKanji2Koe　AqKanji2Koe_Create() / AqKanji2Koe_Create_Ptr()で返されたハンドルを指定

kanji　入力漢字かな混じり文テキスト文字列（Unicode(UTF32), NULL 終端)

koe　出力バッファ。音声記号列文字列が返る（Unicode(UTF32), NULL 終端)

nBufKoe　koe バッファのサイズ[byte] 256 以上を指定。バッファサイズ以上の音声記号列は切り捨てられますので入力テキストの数倍のサイズを指定することを推薦。

戻り値　0:正常終了　それ以外：エラーコード

AqKanji2Koe_SetDevKey　　AqKanji2Koe.h

説明　開発ライセンスキーを設定。アプリ起動後、他の関数を呼び出す前に呼び出すことで、以降、製品版とし動作し、評価版の制限がなくなる。

構文　int **AqKanji2Koe_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。

AqUsrDic.framework（ユーザ辞書ライブラリ）

AqUsrDic ライブラリは、個々の単語の登録を行うインターフェースはありません。アプリケーションでユーザ辞書を変更するときは、CSV 形式の単語リストファイルを介して行います。

AqUsrDic_Import　　AqUsrDic.h

説明　CSV 形式の単語リストからユーザ辞書(aq_usr.dic)を生成
生成する aq_usr.dic と同じディレクトリに、システム辞書（aqdic.bin）が必要。
aq_usr.dic がすでにある場合は上書きする(Append しない)。
【注意】ユーザ辞書は生成時のシステム辞書に依存するため、異なる(サイズの)システム辞書とは使用できない。

構文　int **AqUsrDic_Import**(const char * *pathUserDic*, const char * *pathDicCsv*)

引数

pathUserDic　出力するユーザ辞書(aq_user.dic)ファイルを指定

pathDicCsv　CSV 単語リストファイルファイルを指定(入力データ)

戻り値　0:正常終了　それ以外：エラーコード(詳細は AqUsrDic_GetLastError()で取得)

AqUsrDic_Export　　AqUsrDic.h

説明　ユーザ辞書(aq_usr.dic)から CSV 形式の単語リストを生成
aq_usr.dic と同じディレクトリに、システム辞書（aqdic.bin）が必要。

構文　int **AqUsrDic_Export**(const char * *pathUserDic*, const char * *pathDicCsv*)

引数

pathUserDic	ユーザ辞書(aq_user.dic)ファイルを指定
pathDicCsv	出力する CSV 単語リストファイルファイルを指定
戻り値	0:正常終了　それ以外:エラーコード(詳細は AqUsrDic_GetLastError()で取得)

AqUsrDic_Check　　AqUsrDic.h

説明	単語表記、読みの書式や品詞コードが正しいかチェックする
構文	int **AqUsrDic_Check**(const char * *surface*, const char * *yomi*, int *posCode*)
引数	
surface	単語の表記の文字列を指定(UTF8)
yomi	単語の読み(アクセント付き音声記号列)の文字列を指定(UTF8)
posCode	単語の品詞コード(下記参照)を指定
戻り値	0:チェック OK　それ以外:NG(詳細は AqUsrDic_GetLastError()で取得)

AqUsrDic_GetLastError　　AqUsrDic.h

説明	最後のエラーの詳細メッセージを返す
構文	const char * **AqUsrDic_GetLastError**()
引数	なし
戻り値	エラーメッセージ（UTF8, NULL 終端)

エラーコード表

AqKanji2Koe ライブラリの関数が返すエラーコードの内容は、次の通りです。
AqUsrDic ライブラリに関しては、AqUsrDic_GetLastError()関数で内容を取得してください。

値	内容
100	その他のエラー
101	関数呼び出し時の引数が NULL になっている。
104	初期化されていない(初期化ルーチンが呼ばれていない)

105	入力テキストが長すぎる
106	システム辞書データが指定されていない
107	変換できない文字コードが含まれている
200 番台	システム辞書(aqdic.bin)が不正
300 番台	ユーザ辞書(aq_user.dic)が不正

辞書サイズ

この SDK には、サイズの異なる 3 種類のシステム辞書が含まれています。
利用の目的に応じて選択してお使いください。

辞書名	フォルダ	サイズ	見出し語数
標準辞書	aq_dic	約7MB	約 36 万語
ラージ辞書	aq_dic_large	約 13MB	約 50 万語
スモール辞書	aq_dic_small	約 5MB	約 25 万語

CSV 単語リストファイル

ユーザ辞書の編集で使用する CSV 単語リストファイルの例を下に示します。
1 列目：単語の表記を記述します。文字コードは UTF8 です。すべて全角で記述します。空白は指定できません。
2 列目：単語の読みを音声記号列で指定します(UTF8)。アクセントも ' 記号で指定できます。詳細は音声記号列仕様書の読み記号とアクセント記号の項を参照ください。区切記号は指定できません。
3 列目：単語の品詞コード(下記参照)を半角数値で指定します。

CSV 単語リストファイルの例

```
大江戸線,オーエドセン,5
ＧＤＧＤ,グ'ダグダ,1
天王洲,テンノーズ,7
Ｊアラート,ジェイアラ'ート,5
就活,シューカツ,1
魔理沙,マ'リサ,4
どや顔,ドヤガオ,1
アラフォー,アラフォー,0
```

品詞コード一覧

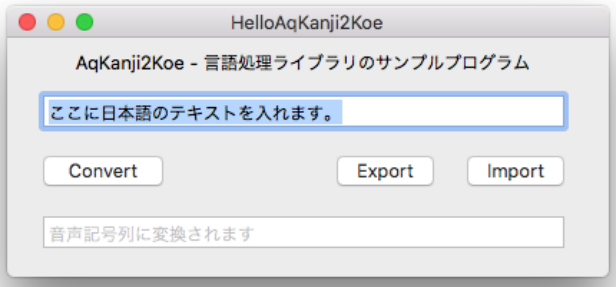
0	名詞
1	名詞(サ変)

2	人名
3	人名(姓)
4	人名(名)
5	固有名詞
6	固有名詞(組織)
7	固有名詞(地域)
8	固有名詞(国)
9	代名詞
10	代名詞(縮約)
11	名詞(副詞可能)
12	名詞(接続詞的)
13	名詞(形容動詞語幹)
14	名詞(ナイ形容詞語幹)
15	形容詞
16	副詞
17	副詞(助詞類接続)
18	接頭詞(名詞接続)
19	接頭詞(動詞接続)
20	接頭詞(数接続)
21	接頭詞(形容詞接続)
22	接続詞
23	連体詞
24	記号
25	記号(アルファベット)
26	感動詞
27	間投詞

サンプルプログラム

AqKanji2Koe 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。

HelloAqKanji2Koe は、任意の漢字かな混じり文を音声記号に変換するアプリケーションです。



ビルド方法

1．アプリケーションプロジェクトを開く

HelloAqKanji2Koe.xcodeproj をダブルクリックして Xcode でプロジェクトを開きます。

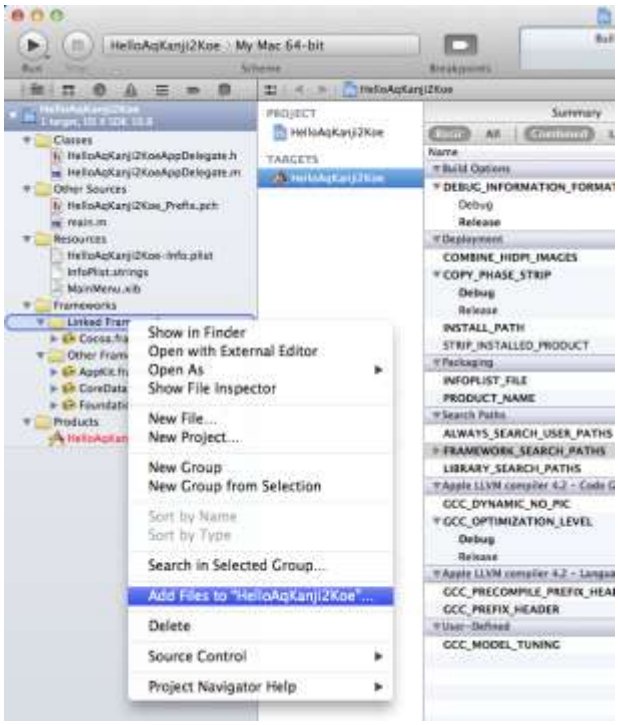
2. プロジェクトにフレームワークの追加]

初期状態では、AqKanji2Koe 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。

「プロジェクトナビゲーター」の[Frameworks]/[Linked Frameworks]の右クリックから[Add Files To “HelloAqKanji2Koe”…]を選択します。

次に、AqKanji2Koe パッケージ内の AqKanji2Koe.framework を選択し、[Add]をクリックします（[Copy items…]のチェックは外したまま）。これで、[Linked Frameworks]に AqKanji2Koe.framework が追加されたのが確認できます。

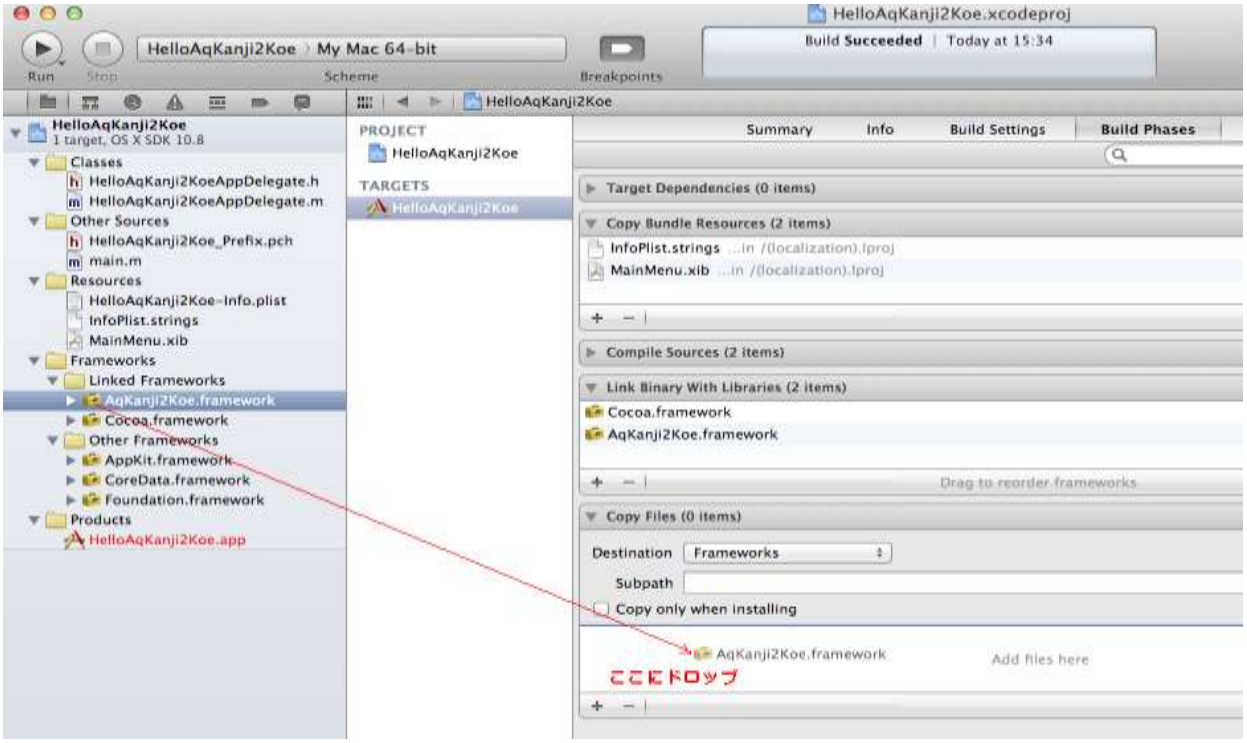
同様の方法で、AqUsrDic.framework も追加します。



3. ターゲットにフレームワークを追加

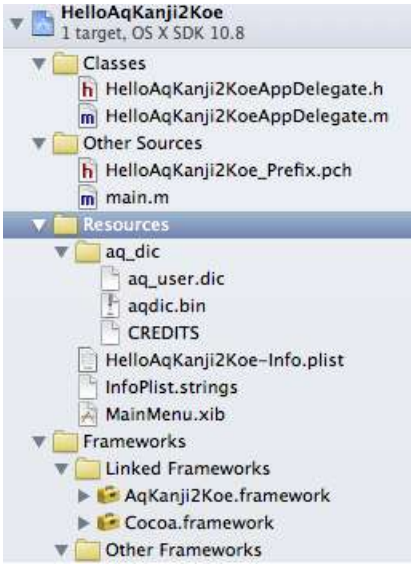
AqKanji2Koe や AqUsrDic は、ダイナミックライブラリなので、実行時にも必要です。そこで、ビルド操作で、実行モジュール HelloAqKanji2Koe.app 内にこれらの framework がコピーされるように設定します。

プロジェクトの Build Phases を開いて、Destination が[Frameworks]になっていることを確認してから、「プロジェクトナビゲーター」の[Frameworks]/[Linked Frameworks]の[AqKanji2Koe.framework]を[Copy Files]の部分にドラッグ＆ドロップします。AqUsrDic.framework も同様に追加します。



4. 辞書データの追加

実行時には辞書データも必要です。このサンプルアプリは、実行時にリソースディレクトリにある辞書ファイルを読み込むようにコーディングされています。そこで、SDK 内の辞書データファイル（aq_dic 以下のファイル）を Resources に追加します。



「プロジェクトナビゲーター」の[Resources]の右クリックから[Add Files to "HelloAqKanji2Koe"]を選択し、SDK の aq_dic フォルダを指定して追加します。このとき、[Options]で、[Copy items if needed]が OFF、[Add folders:]は[Create groups]を選択してください。

5. 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。

実行時に、AqKanji2Koe.framework が見つからないとエラーになる場合は、HelloAqKanji2Koe.app 内の Frameworks フォルダに AqKanji2Koe.framework がコピーされているか確認してください。または、Build Settings > Linking > LD_RUNPATH_SEATCH_PATHS に @loader_path/../Frameworks/ の指定があるか確認してください。

テキストボックスに任意の漢字かな混じり文を入力して、[Convert]ボタンの押下で、下のテキストボックスに音声記号列が表示されれば OK です。なお、開発ライセンスキーを指定していないため、評価版の制限によりナ行マ行がすべてヌになります。

[Export]ボタンを押下し、出力ファイルを指定して[Save]の押下で、ユーザ辞書 aq_user.dic の内容が CSV ファイルとして出力されます。
[Import]ボタンを押下し、CSV 単語リストファイルを指定して[Open]の押下で、CSV ファイルからユーザ辞書 aq_user.dic が構築されます。
ユーザ辞書に単語を追加したり編集を行うには、この CSV 単語リストファイル介して行います。アプリにユーザ辞書の編集機能を付与する場合は、この CSV 単語リストファイルを編集する機能を実装します。

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、AqKanji2Koe で音声記号列に変換する一連の処理が書かれています。

AqKanji2Koe 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）。

applicationDidFinishLaunching()内で、開発ライセンスキーを指定します(13 行目)。正しいキーを指定することで製品版として動作します。なお、ライセンスキーは、ライセンス購入後に発行される開発ライセンス証に記載されています。

AqKanji2Koe の初期化で辞書ファイルのパスを指定する必要がありますので、バンドルの Resource ディレクトリのパスを取得します（20 行目）。

AqKanji2Koe のインスタンスを生成します（26 行目）。

テキストボックスから漢字を含んだテキストを取得し（29 行目）、UTF-8 の C 言語文字列に変換します（32 行目）。

関数 AqKanji2Koe_Convert（）で、音声記号列に変換します。変換結果は koe 配列に戻ります。配列の大きさは変換に十分な大きさを用意してください。このコードではエラー処理を省略していますが、エラー時は関数の戻り値が 0 以外になります。

変換した音声記号列の文字コードは UTF-8 なので、NSString に変換して、テキストボックスへセットします(39 行目)。

最後に、AqKanji2Koe のインスタンスを解放します（42 行目）。

なお、このコードでは、変換の都度 AqKanji2Koe のインスタンスを生成/解放していますが、アプリの起動時に生成し、終了時に解放するようにしたほうが、複数の変換処理を行う場合は、高速に変換することができます。

HelloAqKanji2KoeAppDelegate.m から抜粋

```
1   #import "HelloAqKanji2KoeAppDelegate.h"
2   #import <AqKanji2Koe/AqKanji2Koe.h>
3   #import <AqUsrDic/AqUsrDic.h>
4
5   @implementation HelloAqKanji2KoeAppDelegate
6
7   @synthesize window;
8
9   - (void)applicationDidFinishLaunching:(NSNotification *)aNotification {
10      // Insert code here to initialize your application
11
12      // 開発ライセンスキーの指定。評価版の制限が無くなり、製品版として動作する
13      int iret = AqKanji2Koe_SetDevKey("XXX-XXX-XXX");
14  }
15
16
17  - (IBAction)convKanji2Koe:(id)sender
18  {
19      // 辞書データのディレクトリパス取得(HelloAqKanji2Koe.app/Resources に辞書ファイルをおいた場合)
20      NSBundle *bundle = [NSBundle mainBundle];
21      NSString *path= [bundle resourcePath];
22      const char *pathDic = [path UTF8String];
23
24      // AqKAnji2Koe インスタンス生成
25      int iErr;
26      void *pAqK2K = AqKanji2Koe_Create(pathDic, &iErr);
27
28      // テキストボックスから文字列取得
29      NSString *strKoe = [textfieldKanji stringValue];
30
31      // NSString 文字列を C 言語文字列(Utf8)に変換
32      const char *kanji = [strKoe UTF8String];
33
34      // 漢字かな交じり文を音声記号列に変換
```

```
35      char koe[4096];
36      AqKanji2Koe_Convert(pAqK2K, kanji, koe, 4096);
37
38      // C 言語文字列(Utf8)を NSString に変換してテキストボックスへセット
39      [textfieldKoe setStringValue:[NSString stringWithCString:koe encoding:NSUTF8StringEncoding]];
40
41      // AqKAnji2Koe インスタンス解放
42      if(pAqK2K) AqKanji2Koe_Release(pAqK2K);
43  }
```

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に依存します。このキーは、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。

AqKanji2Koe_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。

CREDITS

本ライブラリは、BSD ライセンスに基づいてライセンスされている下記のオープンソースソフトウェアを使用しています。このライセンスの表示は SDK 付属の CREDITS ファイルを参照ください。また、本ライブラリを含んだアプリを配布する場合は、CREDITS ファイルまたはその内容が含まれるようにしてください。

- MARISA: Matching Algorithm with Recursively Implemented StorAge
- NAIST Japanese Dictionary：形態素解析用辞書

文書履歴

日付	版	変更箇所	更新内容	更新者
2011/01/17	1.0	新規作成		N.Y
2013/07/02	2.0		Ver.2 用に書き換え	N.Y
2017/11/25	3.0		Ver.3 用に書き換え	N.Y