

# SpeechRecSDK for docomo Developer support

開発ガイド for Android

(第 1.0.1 版)

エヌ・ティ・ティ アイティ株式会社

Copyright© 2014 エヌ・ティ・ティ アイティ株式会社

#### 商標

- ・ SpeechRec はエヌ・ティ・ティアイティ株式会社の登録商標です。
- ・ iPhone、iPad、Mac、Apple、iTunes、MacOS、iOS は、米国 Apple Inc.の米国およびその他の国における登録商標または商標です。
- ・ Android は Google Inc.の商標または登録商標です。
- ・ その他、本文中の製品名は各社の商標または登録商標です。

#### 注意

- 1.エヌ・ティ・ティアイティ株式会社からの書面による許諾を得ずに、いかなる方法においても本書の内容の一部または全部を無断で複製、複写、転載、翻訳、頒布することを禁じます。
- 2.本書の内容は予告なく変更する場合があります。
- 3.本書の商品性、特定目的に対する適合性に関して、エヌ・ティ・ティアイティ株式会社は保証いたしません。
- 4.本書の内容については万全を期しておりますが、万一記載内容の誤りなどにお気づきの点がございましたら、エヌ・ティ・ティアイティ株式会社までご連絡ください。
- 5.運用した結果の影響については、4 項にかかわらず、エヌ・ティ・ティアイティ株式会社は一切の責任を負いかねますのでご了承ください。

## 改版履歴

版数	日付	変更点
第 1.0.0 版	2014/4/24	新規作成
第 1.0.1 版	2014/9/26	Android 4.4 (KitKat)での不具合対応

# 目次

<b>目次</b> .....	<b>II</b>
<b>はじめに</b> .....	<b>1</b>
1.1 本書の目的 .....	1
1.2 対象となる読者 .....	1
1.3 用語の定義 .....	2
<b>SPEECHRECSDK の構成</b> .....	<b>4</b>
<b>音声認識の流れ</b> .....	<b>6</b>
3.1 システム構成 .....	6
3.2 動作の流れ .....	7
<b>開発環境の準備</b> .....	<b>9</b>
4.1 ビルド環境の準備 .....	9
4.1.1 JDK (Java Development Kit) .....	9
4.1.2 ADT Bundle .....	9
4.1.3 Pleiades プラグイン .....	10

4.2 ライブラリの配置 .....	10
4.3 ASSETS ファイルの配置 .....	11
<b>API の利用方法.....</b>	<b>12</b>
5.1 処理概要.....	12
5.1.1 初期化.....	12
5.1.2 オプションの設定.....	13
5.1.3 認識の開始.....	14
5.1.4 発話の終了/中止 .....	15
5.1.5 認識結果の取得 .....	15
5.1.6 終了 .....	17
5.2 状態遷移.....	18
<b>API リファレンス .....</b>	<b>19</b>
6.1 音声認識サービスヘルパークラス.....	19
SpeechRecServiceHelper .....	19
6.1.1 音声認識サービスとの接続 .....	19
6.1.2 音声認識の開始 .....	20
6.1.3 音声認識の停止 .....	21
6.1.4 音声認識サービスとの接続の切断 .....	21
6.2 音声認識イベントリスナー.....	22
VoiceRecognitionEventListener .....	22
6.2.1 音声認識サービス接続通知 .....	22
6.2.2 音声録音通知 .....	22
6.2.3 音声録音終了通知.....	23
6.2.4 認識結果通知 .....	23
6.2.5 音声認識完了通知.....	23
6.3 候補クラス .....	24
Nbest.....	24
6.3.1 文候補クラスのリスト取得 .....	24

6.4 文候補クラス .....	25
Sentence.....	25
6.4.1 単語クラスのリスト取得.....	25
6.5 単語クラス.....	26
Word.....	26
6.5.1 単語表記取得.....	26
6.5.2 単語の開始位置取得 .....	26
6.5.3 単語の終了位置取得 .....	27
6.6 DIVIDE ファイル管理クラス.....	27
DivideFileManager .....	27
6.6.1 コンストラクタ .....	27
6.6.2 音声モデルパス取得 .....	28
6.6.3 ファイル展開状態取得 .....	28
6.6.4 ファイル展開 .....	29
<b>サンプルアプリケーション .....</b>	<b>30</b>
7.1 サンプルアプリ概要 .....	30
7.2 サンプルアプリビルド方法.....	30
7.3 サンプルアプリ操作方法 .....	33
<b>参考 .....</b>	<b>36</b>
8.1 認識オプション .....	36
8.2 エラー一覧.....	37

# 1

## はじめに

### 1.1 本書の目的

本書ではおもに SpeechRecSDK のライブラリ仕様、およびサンプルソースの利用方法を記述しています。

### 1.2 対象となる読者

本書は SpeechRecSDK を利用する開発者向けの資料です。

以下のような技術者の方を前提として記述しています。

- ・ 開発対象となるプラットフォーム、開発ツール、ソフトウェア開発の知識を有する方。
- ・ 開発言語に関する知識を有する方。

- ・ 音声認識の基本的な知識を有する方。

## 1.3 用語の定義

用語	説明
ユーザ	SpeechRecSDKを利用して作成した音声認識アプリケーションを利用する利用者
音声認識アプリケーション	音声認識を利用するアプリケーション
SpeechRec	NTT アイティの音声認識システムの総称
SpeechRec SDK	SpeechRec ラインアップのひとつで、お客様独自の音声認識アプリケーションを作成するための開発キット
SpeechRec Server(SRS)	音声認識エンジンを制御するサーバ
SpeechRec Client(SRC)	音声認識アプリケーションが、音声認識を利用するためのインタフェースを提供するライブラリ
音声認識サービス	SRC を通じて音声認識を行うサービスクラス
区間検出制御ライブラリ	区間検出ライブラリを制御するラッパーライブラリ。音声認識サービスと連動し、取得した音声の始端・終端の検出、雑音抑圧を行う
区間検出ライブラリ	区間検出、雑音抑圧の機能を有するライブラリ
エンコード制御ライブラリ	エンコードライブラリを制御するラッパーライブラリ。音声認識サービスと連動し、取得した音声の符号化を行う
エンコードライブラリ	任意のコーデックにて音声データの符号化を行う



	ライブラリ
<b>API キー</b>	docomo Developer support から払い出される API キー
<b>SBM モード</b>	<p>音声区間検出を利用する際に下記の何れかを指定する</p> <p>0: 雑音が大きい環境</p> <p>1: 雑音が比較的小さい環境</p>
<b>中継サーバ</b>	WebSocket プロトコルを用いたプロキシサーバで、音声認識クライアントと音声認識サーバの中継サーバの役割をなす。
<b>認証サーバ</b>	音声認識サーバへの接続時に別途払い出された API キーを用いて認証を行うサーバ

# 2

## SpeechRecSDK の構成

SpeechRecSDK のファイル構成は以下の通りです。

フォルダ/ファイル	概要
SpeechRecSDK/	
Docs/	
SpeechRecSDK for docomo Developer support 製品仕様書.pdf	SpeechRec SDK の製品仕様書
SpeechRecSDK for docomo Developer support 開発ガイド.pdf	アプリ開発時に必要となるライブラリの 配置方法や API の使用方法など
Libs/	
android-websockets.jar	WebSocket ライブラリ
sbm.jar	区間検出制御ライブラリ
speechrec-audio-codec.jar	エンコード制御ライブラリ

<b>speechrec-client.jar</b>	音声認識クライアントライブラリ(SRC)
<b>speechrec-service.jar</b>	音声認識サービスライブラリ
<b>armeabi-v7a/</b>	
<b>libsbm.so</b>	区間検出ライブラリ
<b>libaudiocodec.so</b>	エンコードライブラリ
<b>Assets/</b>	
<b>divide/model_140319</b>	区間検出音声モデル
<b>Sample/</b>	
<b>SpeechRecForAndroid</b>	サンプルプロジェクト
<b>*</b>	

# 3

## 音声認識の流れ

### 3.1 システム構成

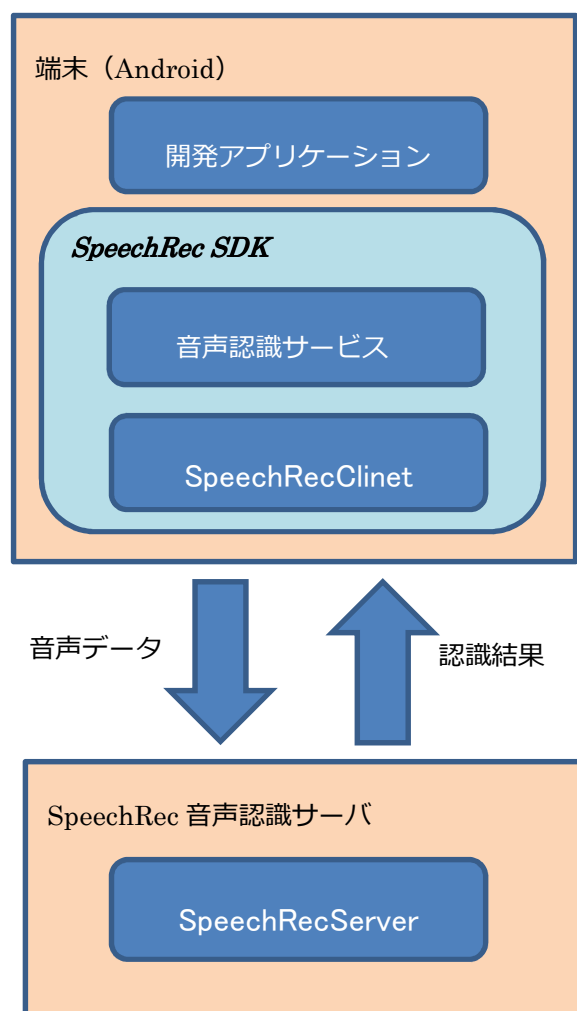
SpeechRec 音声認識システムはユーザが利用する端末と、音声認識処理を行う SpeechRec 音声認識サーバとから構成されます。

SpeechRecSDK として提供される物には、音声認識を利用する為のインタフェースを提供するライブラリ SpeechRecClient(SRC)、及び SRC を通じて音声認識を行う音声認識サービスが含まれています。

端末に向かってユーザが発話をする、その音声はインターネットを経由して音声認識サーバに送信されます。

音声認識サーバでは、送信された音声进行处理し、認識結果を SpeechRecClient に返却します。

SpeechRecSDK を利用して作成したアプリケーションは音声認識サービスから結果を取得し、ユーザに対してその結果などを表示します。



## 3.2 動作の流れ

基本的な動作の流れは以下のようになります。

動作	概要
初期化	音声認識サービスを利用するための初期化処理を行います。

<b>オプションの設定</b>	音声区間を検出するためのモードを設定します。
<b>認識の開始</b>	<p>認識の開始要求を行います。</p> <p>認識の開始要求を受けると、音声認識サービスはマイクデバイスから音声の取得を始めます。</p> <p>また、SpeechRec 音声認識サーバへのセッションを確立し、発話開始を検出すると音声の送信を始めます。</p>
<b>発話の終了/中止</b>	<p>認識の開始後、発話の中止、または停止ができます。</p> <p>発話を終了した場合、認識結果が返却されます。</p> <p>発話を中止した場合には、認識結果を取得せずに復帰します。</p> <p>発話の終了条件は、</p> <ul style="list-style-type: none"> <li>・発話区間の終端を検出した場合</li> <li>・ユーザ（アプリケーション）が明示的に終了メソッドを呼び出した場合</li> </ul> <p>のいずれかです。</p> <p>発話の中止条件は</p> <ul style="list-style-type: none"> <li>・一定時間以上発話区間の始端を検出できない場合</li> <li>・一定時間以上発話区間の終端を検出できない場合</li> <li>・ユーザ（アプリケーション）が明示的に中止メソッドを呼び出した場合</li> </ul> <p>のいずれかです。</p>
<b>認識結果の取得</b>	発話が終了すると認識結果を取得することができます。
<b>終了</b>	音声認識サービスの終期化処理を行います。

# 4

## 開発環境の準備

### 4.1 ビルド環境の準備

SpeechRec SDK 開発では、以下の開発環境を準備して下さい。

#### 4.1.1 JDK (Java Development Kit)

動作環境を参考に、下記のサイトから JDK をダウンロードし、インストールして下さい。

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

#### 4.1.2 ADT Bundle

動作環境を参考に、下記のサイトから ADT (Android Development Tools) Bundle をダウンロードし、インストールして下さい。

<http://developer.android.com/sdk/index.html>

### 4.1.3 Pleiades プラグイン

Eclipse でメニューや表示されるメッセージを日本語化したい場合は、下記のサイトから Pleiades プラグインをダウンロードし、インストールして下さい。

<http://mergedoc.sourceforge.jp/>

## 4.2 ライブラリの配置

ライブラリファイルをアプリケーションに組み込むための方法について説明します。

1. Eclipse で新規プロジェクトを作成し、プロジェクトフォルダ配下にライブラリファイル「jar ファイル」、「so ファイル」を配置して下さい。

libs¥android-websockets.jar

libs¥sbm.jar

libs¥speechrec-audio-codec.jar

libs¥speechrec-client.jar

libs¥speechrec-service.jar

libs¥armeabi-v7a¥libaudiocodec.so

libs¥armeabi-v7a¥libsbm.so

2. プロジェクトのビルドパス設定（ライブラリ）（「プロパティ」 - 「Java のビルドパス」 - 「ライブラリ」タブ）に上記で配置した「jar ファイル」を追加して下さい。
3. Android Manifest の「許可」タブ（Android Manifest Permissions）で、Uses Permission として以下の 2 つを定義して下さい。
  - android.permission.INTERNET
  - android.permission.RECORD\_AUDIO



## 4.3 assets ファイルの配置

assets ファイルをアプリケーションに組み込むための方法について説明します。

1. プロジェクトフォルダ配下に assets ファイル「model\_140319」を配置して下さい。

assets¥divide¥model\_140319

# 5

## API の利用方法

### 5.1 処理概要

SpeechRec SDK で提供する音声認識サービスは、利用者が開発するアプリケーションからサービスへの接続により操作可能です。

#### 5.1.1 初期化

1. DivideFileManager クラスを使用して、区間検出音声モデルを assets フォルダから端末内の任意のディレクトリに展開します。
2. Connect()メソッドを呼び出し、音声認識サービスに接続します。
3. 接続に成功した場合、onServiceConnected() がコールバックされます。このコールバック後、startRecognition()メソッドが呼び出し可能となります。

以下は、サンプルアプリでの初期化の例です。

```
// Divide関連ファイルを端末内に展開
```

```

DivideFileManager divideFileManager =
    new DivideFileManager(this);
if (!divideFileManager.isExtracted()) {
    try {
        divideFileManager.extract();
    } catch (IOException e) {
        .....
    }
}
// 音声認識サービスと接続
helper.connect(this, this);

```

## 5.1.2 オプションの設定

1. 設定値格納用の Bundle を生成します。
2. Bundle に SBM モード、API キーを追加します。
3. Bundle に区間検出音声モデルのファイルパスを追加します。

以下は、サンプルアプリでの SBM モード、API キー設定の例です。<sup>1</sup>

```

public void onClickStart(final View view) {
    int sbm_mode = radioHigh.isChecked() ? 0 : 1;
    Intent intent = new Intent(this, RecognitionActivity.class);
    intent.putExtra(RecognitionActivity.KEY_SBM_MODE, sbm_mode);
    // 別途発行されるAPIキーを設定してください(以下の値はダミーです)
    intent.putExtra(RecognitionActivity.KEY_API_KEY, "123456789");
}

```

<sup>1</sup> サンプルアプリでは、インテント発行時に SBM モード、API キーを Intent の extras に設定しています。

```
startActivityForResult(intent, RECOGNIZE_ACTIVITY_REQUEST_ID);  
}
```

以下は、サンプルアプリでのオプション設定の例です。

```
bundle = new Bundle();  
// BundleにIntentの値（SBMモード、APIキー）を追加  
Intent intent = getIntent();  
bundle.putAll(intent.getExtras());  
// Bundleに区間検出モデルファイルを追加  
bundle.putString(KEY_VAD_MODEL,  
                 divideFileManager.getDivideModelPath());
```

### 5.1.3 認識の開始

1. 設定値を格納した Bundle を引数に startRecognition() メソッドを呼び出し、音声認識処理を開始します。
2. 認識が開始されると、一定間隔（20msec）で音声録音通知 onRecord() がコールバックされます。録音した音声データ（16kHz、16bit リニア PCM 形式）が引数で渡されますので、アプリケーション側では音声データをファイルとして保存したり、レベルメーターの表示に利用可能です。音声データを利用しない場合は、onRecord() メソッドをオーバーライドする必要はありません。

以下は、サンプルアプリでの認識開始の例です。<sup>2</sup>

```
@Override  
public void onServiceConnected() {  
    // 音声認識処理を開始
```

<sup>2</sup> 接続成功時にコールバックされる onServiceConnected() メソッド内で音声認識を開始しています。

```
helper.startRecognition(bundle);  
}
```

## 5.1.4 発話の終了/中止

1. stopRecognition()メソッドを呼び出し、音声認識処理を停止し認識完了通知待ち状態にします。但し、音声区間検出が有効の場合、終端を検知すると自動的に stopRecognition()がコールされますので、アプリケーション側から stopRecognition()を呼ぶ必要はありません。
2. 録音が終了すると、音声録音終了通知 onStopRecording()がコールバックされます。
3. 認識処理を中止したい場合は、close()メソッドを呼び出すことで認識処理は中止可能です。

以下は、サンプルアプリでの発話終了の例です。<sup>3</sup>

```
public void onClickFinishButton(View view) {  
    // 音声認識処理を終了  
    helper.stopRecognition();  
}
```

## 5.1.5 認識結果の取得

1. stopRecognition()メソッドを呼び出すと、認識結果通知 onResult()がコールバックされます。

---

<sup>3</sup> 終端を検知すると自動的に stopRecognition()がコールされるので、ここではマイクアイコンに割り当てた終了ボタンクリック時の呼び出しになります。

2. 認識結果は Nbest クラスに格納されて引数で渡されますので、Nbest クラスのメソッドを使い、認識結果を取得します。

以下は、サンプルアプリでの認識結果取得の例です。

```
@Override
public void onResult(Nbest result) {
    List<Sentence> sentenceList = result.getSentenceList();
    if (resultList == null) {
        resultList = new LinkedList<StringBuilder>();
        for (int i = 0; i < sentenceList.size(); i++) {
            resultList.add(new StringBuilder());
        }
    } else {
        while (resultList.size() > sentenceList.size()) {
            resultList.removeLast();
        }
    }
    for (int i = 0; i < resultList.size(); i++) {
        StringBuilder sb = resultList.get(i);
        Sentence sentence = sentenceList.get(i);
        for (Word word : sentence.getWordList()) {
            String label = word.getLabel();
            // ラベルがnullは無視
            if (label == null) {
                continue;
            }
            // セミコロン以降、空白のみは無視
            label = label.replaceAll(";.*", "").replaceAll("[ ]", "");
            // ラベルが空文字列は無視
            if (label.length() == 0) {
```

```

        continue;
    }
    // 音声認識結果に追加
    if ((separator != null) && (sb.length() > 0)) {
        sb.append(separator);
    }
    sb.append(label);
}
}
}

```

## 5.1.6 終了

1. 音声認識が完了すると、音声認識完了通知 `onFinish()` がコールバックされます。このコールバック後、再び `startRecognition()` メソッドが呼び出し可能となります。
2. 再び認識を開始しない場合は、`close()` メソッドを呼び出し音声認識サービスとの接続を切断します。`close()` メソッドは、`connect()` 成功後であればどの時点でも呼び出し可能です。

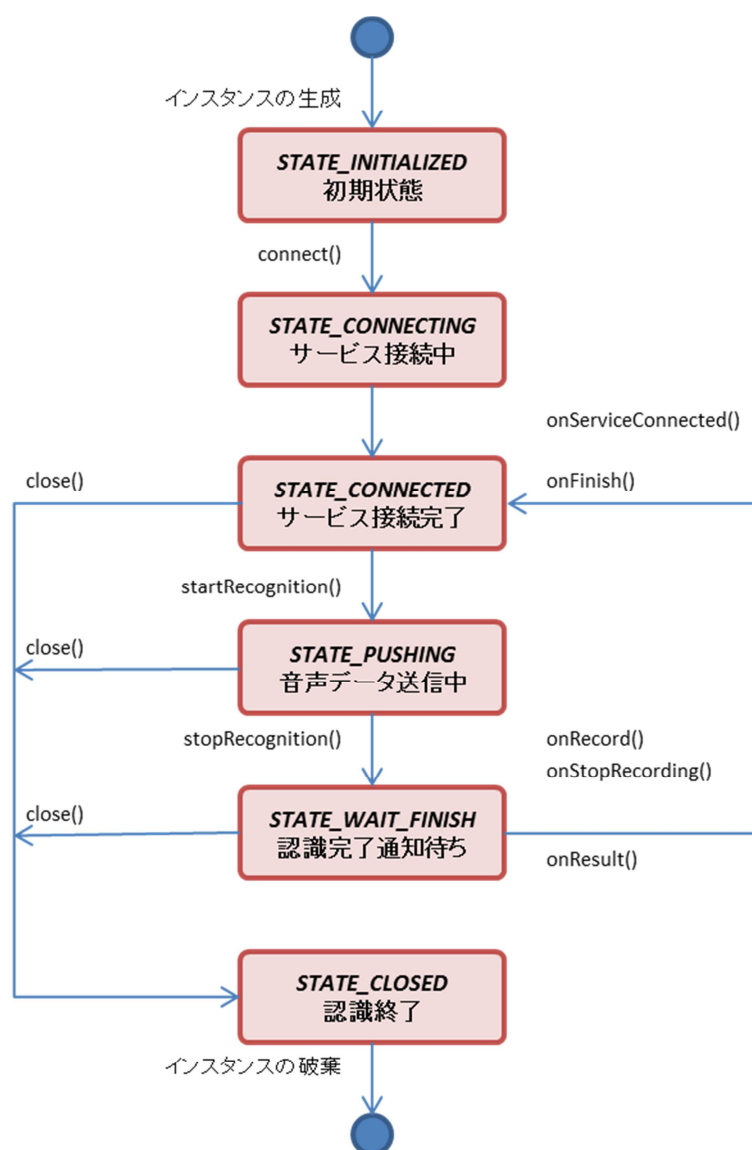
以下は、サンプルアプリでの終了の例です。

```

@Override
protected void onDestroy() {
    // 音声認識サービスとの接続を切断
    helper.close();
    super.onDestroy();
}

```

## 5.2 状態遷移





# 6

## API リファレンス

### 6.1 音声認識サービスヘルパークラス

#### SpeechRecServiceHelper

音声認識サービス（SpeechRecService クラス）の仕様を簡略化するヘルパークラスです。

#### 6.1.1 音声認識サービスとの接続

音声認識サービスに接続します。

```
void connect(Context context,  
             VoiceRecognitionEventListener callback)
```

パラメータ	context	アプリケーションコンテキスト
-------	---------	----------------

	callback	イベントリスナ
例外	NullPointerException	引数に null が指定された場合
	IllegalStateException	既に接続済みか close()メソッド呼び出し後だった場合
解説	接続に成功した場合、  VoiceRecognitionEventListener.onServiceConnected() がコールバックされます	

## 6.1.2 音声認識の開始

音声認識を開始します。

```
void startRecognition(Bundle parameter)
```

パラメータ	parameter	音声認識処理のパラメータを格納した Bundle
例外	IllegalStateException	接続済みでない、既に開始済み、close()メソッド呼び出し後のいずれかの場合
解説	音声区間検出が無効の場合、  SpeechRecServiceHelper.stopRecognition() を呼び手で終了しなければなりません。音声区間検出が有効の場合、自動で認識が完了し VoiceRecognitionEventListener.onFinish(Exception) がコールバックされます。  この場合でも SpeechRecServiceHelper.stopRecognition() を呼び手で終了することができます。	

### 6.1.3 音声認識の停止

音声認識を停止します。

```
void stopRecognition()
```

パラメータ	-
例外	IllegalStateException 音声認識を開始していない場合
解説	音声認識を停止し、音声認識完了通知待ち状態にします。  本メソッドは音声認識の開始後に呼び出すことができます。

### 6.1.4 音声認識サービスとの接続の切断

音声認識サービスとの接続を切断します。

```
void close()
```

パラメータ	-
例外	-
解説	本メソッドはconnect()成功後であればどの時点でも 呼び出し可能です。音声認識を中止したい場合は、stopRecognition()メソッドをコールせず、本メソッドをコールする事で音声認識は中止されます。

## 6.2 音声認識イベントリスナー

### VoiceRecognitionEventListener

音声認識サービスヘルパーから音声認識のイベントを受け取るインターフェースです。

#### 6.2.1 音声認識サービス接続通知

音声認識サービスクラスに接続完了した場合に通知されます。

```
void onServiceConnected()
```

パラメータ	-
例外	-
解説	このコールバック後、startRecognition()メソッドが呼び出し可能となります。

#### 6.2.2 音声録音通知

音声を録音した場合に通知されます。

```
void onRecord(short[] samples)
```

パラメータ	samples	16bit, 16kHz, リニア PCM 形式の音声データ
例外	-	

解説	一定間隔(20msec)でコールバックされます。
----	--------------------------

## 6.2.3 音声録音終了通知

音声録音が終了した場合に通知されます。

```
void onStopRecording()
```

パラメータ	-
例外	-
解説	マイクデバイスからの音声データ取得スレッドが終了した契機で通知されます。

## 6.2.4 認識結果通知

音声認識サーバからの認識結果通知です。

```
void onResult(Nbest result)
```

パラメータ	result	音声認識結果
例外	-	
解説	認識結果通知は音声認識の開始から、音声認識完了通知を受信するまでの間に、複数回通知されます。	

## 6.2.5 音声認識完了通知

音声認識サーバからの音声認識完了通知です。

```
void onFinish(Exception e)
```

パラメータ	e	音声認識でエラーが発生した場合は例外オブジェクトが、成功した場合はnullが渡されます。
例外	-	
解説	このコールバック後、再びstartRecognition()が呼び出し可能となります。	

## 6.3 候補クラス

### Nbest

音声認識結果のうち、候補の内容取得機能を提供します。

音声認識結果クラスは以下のクラス群で構成されます。

- Nbest (候補クラス)
  - Sentence (文候補クラス)
    - ◇ Word (単語クラス)

### 6.3.1 文候補クラスのリスト取得

文候補クラスのインスタンスのリストを取得します。

```
List<Sentence> getSentenceList()
```

パラメータ	-
例外	-
解説	<p>取得可能なリスト数は候補のセンテンス数となります。</p> <p>取得可能なセンテンス数は音声認識サーバの設定に依存します。</p>

## 6.4 文候補クラス

### Sentence

音声認識結果のうち、文候補の内容取得機能を提供します。

#### 6.4.1 単語クラスのリスト取得

単語クラスのインスタンスのリストを取得します。

```
List<Word> getWordList()
```

パラメータ	-
例外	-
解説	取得可能なリスト数は文候補の単語（形態素）数となります。

## 6.5 単語クラス

### Word

音声認識結果のうち、単語の内容取得機能を提供します。

#### 6.5.1 単語表記取得

単語表記を取得します。

```
String getLabel()
```

パラメータ	-
例外	-
解説	単語表記は以下のフォーマットで表現しています。 「表記;読み;発音;品詞情報」

#### 6.5.2 単語の開始位置取得

単語の開始位置（秒）を取得します。

```
double getStartPoint()
```

パラメータ	-
例外	-
解説	取得値は、SpeechRecServiceHelper#startRecognition()呼び出し後に、最初に設定した音声データ先頭位置を基準としていま



---

す。

---

### 6.5.3 単語の終了位置取得

単語の終了位置（秒）を取得します。

```
double getEndPoint()
```

---

パラメータ	-
-------	---

---

例外	-
----	---

---

解説	取得値は、SpeechRecServiceHelper#startRecognition()呼び出し後に、最初に設定した音声データ先頭位置を基準としています。
----	---

---

## 6.6 DIVIDE ファイル管理クラス

### DivideFileManager

区間検出機能で使用するファイルを端末内に展開するクラスです。

#### 6.6.1 コンストラクタ

インスタンスを生成します。

```
DivideFileManager(Context context)
```

<b>パラメータ</b>	context	アプリケーションコンテキスト
<b>例外</b>	-	
<b>解説</b>	インスタンスを生成します。	

## 6.6.2 音声モデルパス取得

端末内に展開された音声モデルファイルのパスを取得します。

```
String getDivideModelPath()
```

<b>パラメータ</b>	-
<b>例外</b>	-
<b>解説</b>	端末内に展開された音声モデルファイルの絶対パスをします。

## 6.6.3 ファイル展開状態取得

端末内にファイルが展開された否かの状態を取得します。

```
boolean isExtracted()
```

<b>パラメータ</b>	-
<b>例外</b>	-
<b>解説</b>	ファイルが展開済みの場合 true、展開されていない場合 false を返します。

## 6.6.4 ファイル展開

区間検出で使用するファイルを端末内に展開します。

```
void extract() throws IOException
```

パラメータ	
例外	<code>IOException</code> ファイルの展開に失敗した場合
解説	assetsフォルダに置かれたDIVIDE関連ファイルを端末内に展開します。

# 7

## サンプルアプリケーション

### 7.1 サンプルアプリ概要

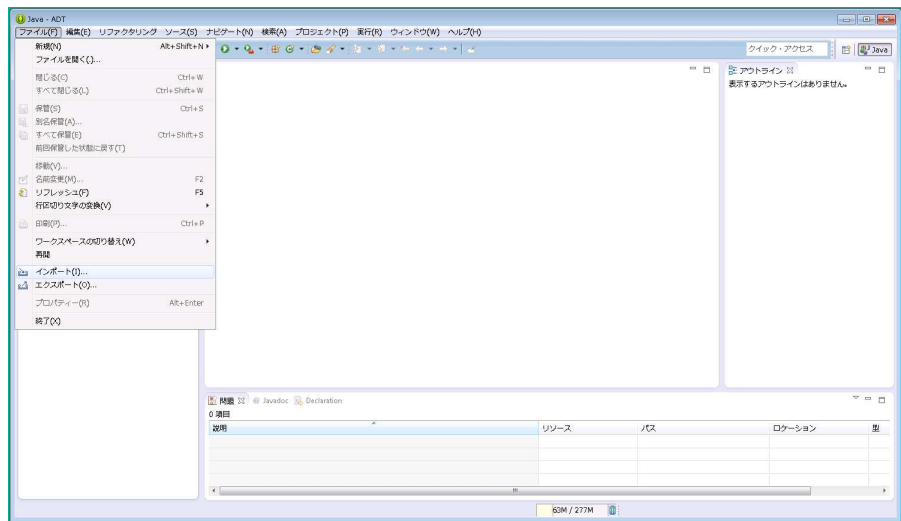
SpeechRec SDK には、音声認識サービスを利用した Android のサンプルアプリが含まれています。

### 7.2 サンプルアプリビルド方法

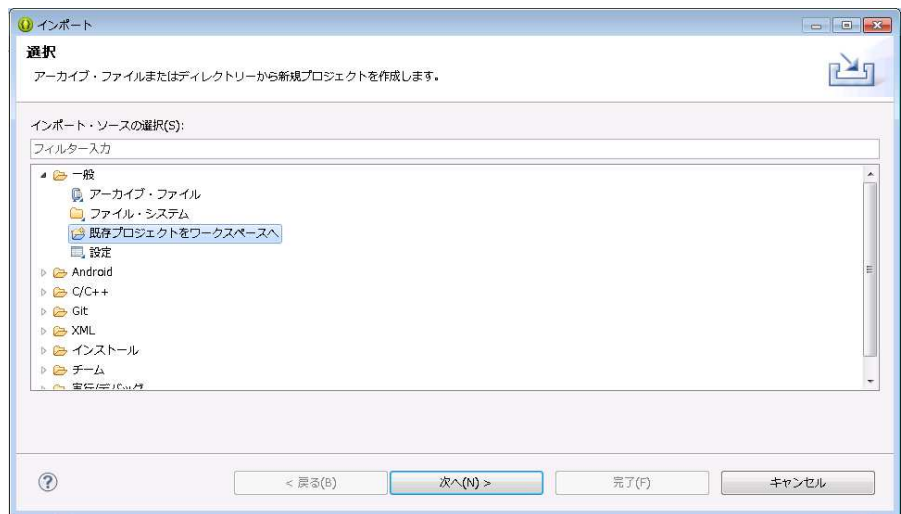
サンプルアプリのビルド方法は次の通りです。

以下、ビルド手順は Eclipse v4.2 での説明となります。

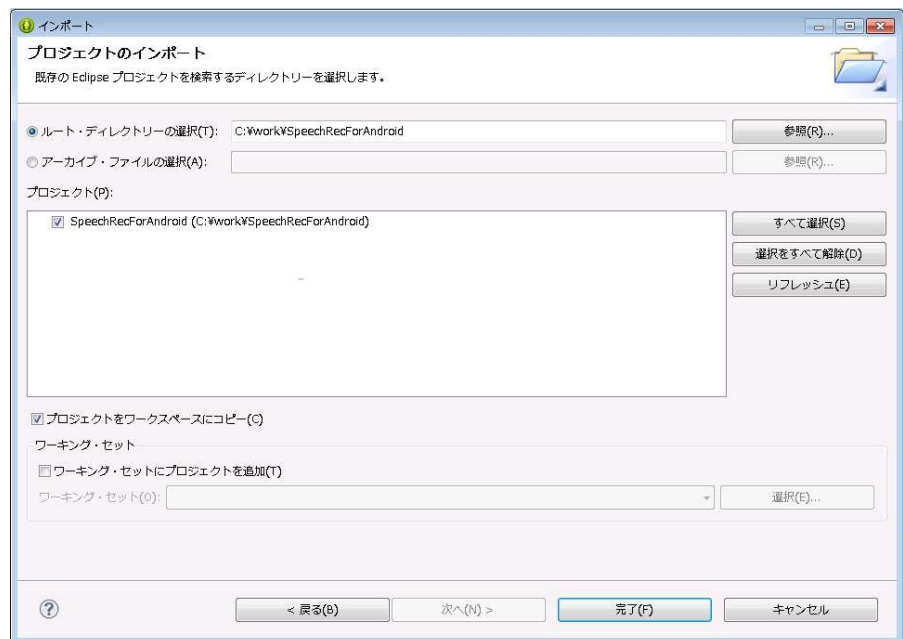
1. Eclipse を起動します。
2. ファイルメニューから「インポート」を選択します。



3. インポート画面の「インポートソースの選択」で「一般」内の「既存プロジェクトをワークスペースへ」を選択し、「次へ」を選択します。



4. インポート画面の「ルートディレクトリの選択」でサンプルアプリのディレクトリを選択し、「完了」を選択します。



5. パッケージビューにインポートしたプロジェクトが追加されます。正常にインポートされない場合には、プロジェクトを削除し、再インポートしてください。

## 7.3 サンプルアプリ操作方法

サンプルアプリの操作方法是次の通りです。

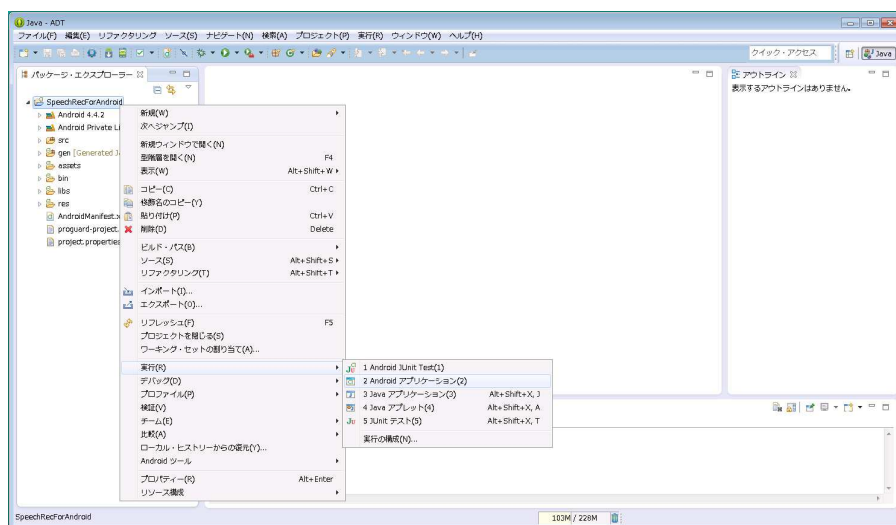
1. Eclipse で Android アプリケーションプロジェクトを読み込みます。

「7.2 サンプルアプリビルド方法」を参照ください。

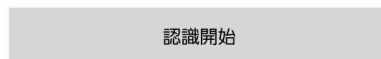
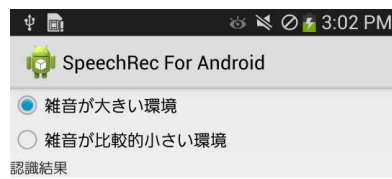
2. サンプルアプリをインストールする端末を USB 接続し、接続が確立されたことを確認してください。

事前に端末用 USB ドライバと ADB USB ドライバをインストールしておいてください。

3. Eclipse メニューから「実行」を選択してください。サンプルアプリがインストールされ、起動します。



4. 周囲の騒音状況に応じて、推奨環境変数セット 1、2 のいずれかを選択してください。
5. 「認識開始」をタップして、音声認識を開始します。



6. 文章や単語をお話してください。

発話を行うと、音量に応じてレベルメーターが表示されます。



7. 認識結果が表示されます。





認識開始

8. 再度、音声認識を行う場合は、手順 4 から始めてください。
9. バックキー押下で画面が終了します。

#### ※ご利用上の注意

- ネットワーク経由で音声認識サーバに接続して認識するためパケット通信を行います。
- 会話をするように自然にお話ください。
- 話し方や周囲の騒音状況によっては、お客様が意図しない認識結果となる場合があります。
- 認識できる言語は日本語のみです。
- 音声認識を中止する場合は、バックキー押下で音声認識をキャンセルします。

# 8

## 参考

### 8.1 認識オプション

認識開始時に指定可能なオプションは以下の通りです。

No	項目	キー	型	必須	説明
1	API キー	api_key	String	○	docomo Developer support から払い出される API キーを設定
2	SBM モード	sbm_mode	int	×	環境によって下記の何れかを設定 0: 雑音が大きい環境 1: 雑音が比較的小さい環境 2: 雑音が高い環境（始端検出なし） 3: 雑音が比較的小さい環境

(始端検出し)

default: 2

## 8.2 エラー一覧

SpeechRec SDK で出力されるエラー一覧を以下に示します。

No	エラーメッセージ	発生契機
1	Divide モデルファイルが指定されてい ません	パラメータを格納した Bundle の内容が不正
2	接続試行間隔に負の値が指定されていま す	パラメータを格納した Bundle の内容が不正
3	接続リトライ回수에 0 未満の値が指定さ れています	パラメータを格納した Bundle の内容が不正
4	接続タイムアウト時間に 0 未満の値が指 定されています	パラメータを格納した Bundle の内容が不正
5	認識結果取得タイムアウト時間に 0 未満 の値が指定されています	パラメータを格納した Bundle の内容が不正
6	音声入力タイムアウト時間に 0 未満の値 が指定されています	パラメータを格納した Bundle の内容が不正
7	区間検出時間に 0 未満の値が指定されて います	パラメータを格納した Bundle の内容が不正
8	SBM モードに 0, 1 以外の値が設定され ています	パラメータを格納した Bundle の内容が不正
9	API キーが指定されていません	パラメータを格納した Bundle の内容が不正
10	マイクデバイスの初期化に失敗しました	端末のマイクデバイスが取得できない

11	既に録音開始しています	録音スレッドが開始している状態で start()メソッドがコールされた
12	接続が確立できません	終端検出までに接続が確立できない
13	音声聞こえません	始端が検出できず、音声入力タイムアウトが発生
14	終端が検出できません	終端が検出できず、区間検出タイムアウトが発生
15	音声データのバッファが最大値を超えました	音声データ蓄積用の内部バッファでオーバーフロー発生
16	SBM モードの値が無効です	SbmFilter のコンストラクタで不正な引数が指定された
17	SBM ライブラリのロードに失敗しました	libsbm.so がロードできない
18	SBM ライブラリが存在しません	libsbm.so が見つからない
19	SBM の初期化に失敗しました	SBM ライブラリの初期処理で例外発生
20	区間検出でエラーが発生しました	SBM ライブラリの区間検出、雑音除去処理で例外発生
21	既に区間検出を開始しています	区間検出が開始している状態で open()メソッドがコールされた
22	モデルファイルが存在しません	Divide 音声モデルが見つからない
23	モデルファイルの読み込みに失敗しました	Divide 音声モデルが読み込めない
24	区間検出に失敗しました	DetectSbm()、ReadSbm()の戻り値が異常
25	エンコーダの初期化に失敗しました	エンコードライブラリの初期処理で例外発生
26	エンコード処理でエラーが発生しました	エンコードライブラリの圧縮処理で例外発生

27	divide/model_140319	Divide が使用するファイルを端末内に展開できない
28	通知の形式が不正なため、文の開始位置の通知に失敗しました	電文不正
29	通知の形式が不正なため、文の終了位置の通知に失敗しました	電文不正
30	通知の形式が不正なため、文の暫定結果位置の通知に失敗しました	電文不正
31	通知の電文にボディが含まれていません	電文不正
32	認識結果に Score タグが含まれていません	電文不正
33	認識結果の取得でタイムアウトが発生しました	認識結果待ち状態でタイムアウト発生
34	接続でタイムアウトが発生しました	音声認識サーバとの接続中にタイムアウト発生
35	接続処理中に例外が発生しました	音声認識サーバとの接続中に例外発生
36	WebSocket の接続でタイムアウトが発生しました	中継サーバとの接続中にタイムアウト発生
37	WebSocket の認証に失敗しました	中継サーバの認証ができない
38	コネクションが切断されました	中継サーバへデータ送信中にコネクションが切断された
39	開始処理を行える状態ではありません	状態不正（データ送受信処理を開始できる状態でない）
40	送信処理を行える状態ではありません	状態不正（リクエスト、レスポンスを送信できる状態でない）
41	body の終端が “—” で終了しています	電文不正（バウンダリがパース出来ない）

	ん	
42	リクエストラインまたはステータスラインのフォーマットが不正です	電文不正（ステータスライン、リクエストラインの正規表現パターンに一致しない）
43	ヘッダ名と値の区切り文字が存在しません	電文不正（ヘッダーがパース出来ない）
44	ヘッダ名が空白です	電文不正（ヘッダーがパース出来ない）
45	content-type に multipart/form-data が指定されていますが、バウンダリが存在しない、または不正です	電文不正（content-length が存在しない、マルチパートに関わらずバウンダリが存在しない）
46	content-length の値に数値以外の文字が存在します	電文不正（content-length が存在しない、content-length の値に数値以外の文字が含まれる）
47	マルチパート電文のヘッダー部とボディ部の区切りが存在しません	電文不正（ヘッダーがパース出来ない）
48	マルチパート電文の content-length と実際の body のサイズに相違があります	電文不正（ボディ部の長さが異なる）
49	リクエストを送信できる状態ではありません	状態不正（リクエストを送信できる状態でない）
50	レスポンスリスナが登録されていない状態でレスポンス電文を受信しました	状態不正（レスポンスリスナ未登録）
51	レスポンス電文の応答前にリクエストを受信しました	状態不正（リクエストを受信できる状態でない）
52	リクエストリスナが登録されていない状態でリクエスト電文を受信しました	状態不正（リクエストリスナ未登録）
53	レスポンスを送信できる状態ではありません	状態不正（レスポンスを送信できる状態でない）

54	レスポンス電文の送信処理で例外が発生しました	レスポンス送信処理中に例外発生
55	ランタイムエラーが発生しました。詳細=XXX	ローカルファイル（認識モデル）の読み込みで例外発生
56	引数に不正な値が指定されました。	メソッドの引数に不正な値が指定された
57	コネクト処理に失敗しました。原因=XXX	音声認識サーバとの接続に失敗
58	通信異常が発生しました。原因=XXX	コネクションが切断された
59	タイムアウトが発生しました。	通信処理でタイムアウト発生
60	プロトコルエラーが発生しました。	通知電文の XML のタグ名に該当する子ノードが存在しない
61	エラーレスポンスを受信しました。ステータスコード=XXX	ステータスコードが 200 OK でない
62	不正なパラメータが指定されました。	setParameter()にて不正なパラメータが指定された
63	指定された処理は未サポートです。	未サポートの API がコールされた
64	範囲外の領域へアクセスしました。	範囲外の領域へのアクセスが発生
65	通知処理に失敗しました。	通知電文のノード取得時に例外発生
66	エンコードモジュールでエラーが発生しました。詳細=XXX	エンコードモジュールで例外発生
67	音声分析モジュールでエラーが発生しました。詳細=XXX	音声分析モジュールで例外発生
68	音声認識または音声録音の完了処理に失敗しました。詳細=XXX	完了待ち合わせ中に例外が発生、または完了待ち合わせタイムアウト発生
69	認証処理に失敗しました。原因=XXX	API キーでの認証ができない

## SpeechRecSDK

### 開発ガイド

---

---

発行	エヌ・ティ・ティ アイティ株式会社
	〒231-0032
	神奈川県横浜市中区不老町二丁目 9 番地 1
	<a href="http://www.ntt-it.co.jp/">http://www.ntt-it.co.jp/</a>

---