

絵文字利用アプリケーション開発ガイド

— 絵文字パッケージ搭載機種におけるキャリア絵文字及びデコメ絵文字の
入力・表示のためのアプリケーション開発者ガイド —

Ver. 1.2.1

オムロン ソフトウェア株式会社
2012/12/3

目次

1. 本ドキュメントについて	1
2. 略語・記号の定義	1
3. 概要	1
3.1. 絵文字パッケージとは	1
3.2. 本ドキュメントの読者の前提	2
3.3. 本ドキュメントが前提とする環境	2
3.4. 関連ドキュメント	2
4. 絵文字パッケージの機能	3
4.1. ソフトウェア構成	3
4.2. 絵文字パッケージで提供するモジュール	4
4.2.1. 絵文字描画モジュール	4
5. 絵文字の入力と表示	5
5.1. TextViewでの絵文字の入力・表示	5
5.1.1. 絵文字入力可能なエディタの生成	5
5.1.2. EmojiAssistに組み込み、絵文字表示を可能にする	5
5.1.3. 文字列からデコメ絵文字を取得	6
5.1.4. アプリケーションからキャリア絵文字を含む文字列を出力する際の注意事項	6
5.2. WebViewへの絵文字の表示	7
5.2.1. キャリア絵文字の表示	7
5.2.2. デコメ絵文字の表示	8
5.3. 絵文字パッケージをサポートしないAndroid端末での動作について	8
6. デコメ絵文字の登録、削除	9
6.1. 登録先ディレクトリ	9
6.2. デコメ絵文字の定義	9
7. 絵文字利用アプリケーションの拡張機能	9
7.1. デコメ絵文字拡張情報の利用	9
8. 注意事項	11

8.1. スリープ時の注意事項	11
8.1.1. 対応策 (アクティビティアプリケーション向け)	11
8.1.2. 対応策 (サービスアプリケーション向け (IME を含む))	11
8.2. ハードウェアアクセラレーションに関する制限事項	12
9. API仕様	13
9.1. TextView/EditText (絵文字入力の有効化)	13
9.2. EmojiAssistクラス	13
9.2.1. インスタンス取得	13
9.2.2. TextView登録	14
9.2.3. TextView削除	15
9.2.4. TextView全削除	16
9.2.5. アニメーション開始	17
9.2.6. アニメーション停止	18
9.2.7. 絵文字データ取得	19
9.2.8. デコメピクチャ拡張表示設定	20
9.3. DecoEmojiSpanクラス	21
9.3.1. URI取得	21
9.3.2. 幅取得	22
9.3.3. 高さ取得	23
9.3.4. デコメピクチャ判定	24
9.4. EmojiDrawableクラス	25
9.4.1. 絵文字判定	25
9.4.2. 絵文字判定(コード)	26
10. デコメ絵文字の読み付加情報フォーマット	27

1. 本ドキュメントについて

本ドキュメントは、Android PF 上で絵文字の入力・表示を可能にするソリューション「絵文字パッケージ」を利用する機能をアプリケーションに組み込むためのガイドラインです。

本ガイドラインに沿って作成されたアプリケーションでは、絵文字パッケージに対応した Android 端末において、キャリア絵文字・デコモ絵文字の表示や、絵文字パッケージに対応した IME でのキャリア絵文字・デコモ絵文字の入力を行うことが可能となります。

2. 略語・記号の定義

IME	Input Method Editor
絵文字	キャリア絵文字とデコモ絵文字の総称。
キャリア絵文字	携帯電話事業者ごとに定められた、文字コードを持つ絵文字。
デコモ絵文字	文章中にキャリア絵文字と同様に用いられる画像。 携帯電話事業者により様々な呼称があるが、本書では代表して「デコモ絵文字」と呼ぶ。
デコモピクチャ	文章中に挿入される自由なサイズの画像。
デコモコンテンツ	デコモ絵文字、デコモピクチャなどの画像ファイルの総称。 ※「デコモ」、「デコモ絵文字」、「デコモピクチャ」は、株式会社 NTT ドコモの登録商標です。

3. 概要

3.1. 絵文字パッケージとは

Android 絵文字パッケージは、Android アプリケーションで絵文字(キャリア絵文字、デコモ絵文字)・デコモピクチャの入力および表示を行えるようにするためのソフトウェアです。Android 標準のフレームワークではキャリア絵文字の入力、表示はできませんが、デコモ絵文字の入力、表示をすることはできません。しかし、アプリケーションに絵文字パッケージを組み込むことによって、標準的なアプリケーションフレームワーク(IMF、TextView、WebView)を使い、キャリア絵文字だけでなく、デコモ絵文字についても入力および表示を行うことが可能となります。

また、IME を起動して文字入力する際に変換候補として表示するデコモ絵文字や、デコモ絵文字選択パレットに表示するデコモ絵文字をユーザが追加することも可能となります。

そのためには、アプリケーション開発者はライブラリをアプリケーションに組み込む必要があります。また、絵文字の入力を行うためには、Android 端末の IME が絵文字パッケージに対応している必要があります。

以下に、絵文字パッケージをアプリケーションに組み込むことで可能となる機能を説明します。

大項目	中項目	小項目	機能概要
絵文字の入力と表示	TextView	キャリア絵文字	Android 標準フレームワークの TextView を用いて、通常文字と合わせてキャリア絵文字を入力・表示する。
		デコメ絵文字	Android 標準フレームワークの TextView を用いて、通常文字と合わせてデコメ絵文字を入力・表示する。
	WebView	キャリア絵文字	Android 標準フレームワークの WebView を用いて、キャリア絵文字を表示する。
		デコメ絵文字	Android 標準フレームワークの WebView を用いて、デコメ絵文字を表示する。
デコメ絵文字の登録、削除	登録	-	ユーザが IME を起動して文字入力する際に変換候補として表示されるデコメ絵文字や、デコメ絵文字選択パレットに表示されるデコメ絵文字を登録する。
	削除	-	ユーザが IME を起動して文字入力する際に変換候補として表示されるデコメ絵文字や、デコメ絵文字選択パレットに表示されるデコメ絵文字を削除する。

表 1 絵文字パッケージ機能一覧

3.2. 本ドキュメントの読者の前提

本ドキュメントは以下を満たす方が読むことを前提に記載しています。

- 一般的な Android アプリの開発方法を熟知している方。
本ドキュメントでは、Android アプリケーション開発の詳細には触れません。絵文字パッケージを Android アプリケーションに組み込む方法にのみ言及します。
- キャリア絵文字、デコメ絵文字の仕様を熟知している方。
本ドキュメントでは、キャリア絵文字、デコメ絵文字についての詳細には触れません。キャリア絵文字、デコメ絵文字については、各携帯電話事業者の Web サイトを参照してください。

3.3. 本ドキュメントが前提とする環境

本ドキュメントは以下の環境で動作するアプリを前提として記載しています。

- OS: Android OS V4.0(ICS)
- Android 端末: 絵文字パッケージを搭載している Android 端末
詳細については各携帯電話事業者の Web サイトを参照してください。

3.4. 関連ドキュメント

キャリア絵文字やデコメ絵文字が入力可能な IME を開発する方あるいは絵文字を選択する絵文字一覧(パレット)を開発する方は、「絵文字パッケージ対応 IME 開発ガイド」を参照してください。

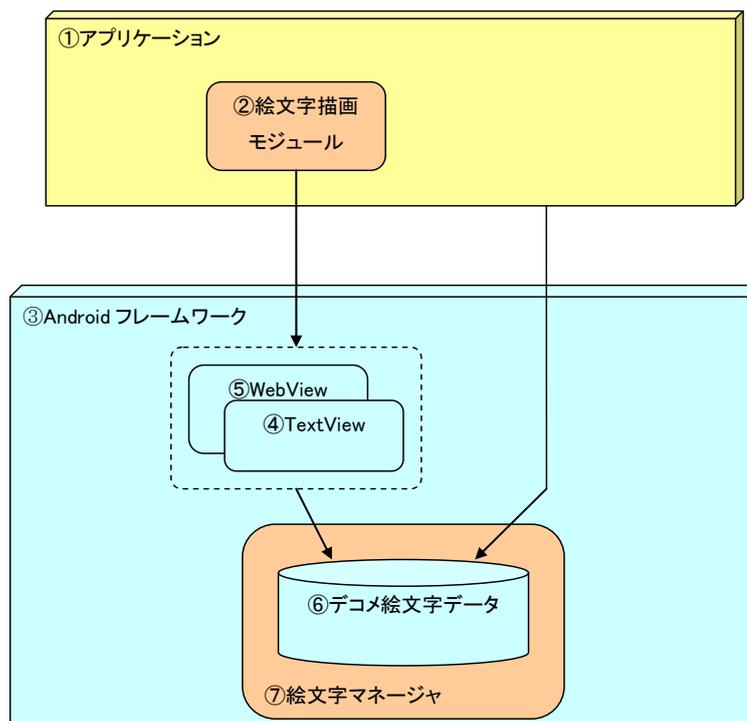
また、デコメ絵文字に読み情報を付加する方法は「デコメ絵文字コンテンツ作成ガイド」を参照してください。

なお、キャリア絵文字、デコメ絵文字については、各携帯電話事業者の Web サイトを参照してください。

4. 絵文字パッケージの機能

4.1. ソフトウェア構成

アプリケーションに絵文字パッケージを組み込んだ場合のソフトウェア構成を以下に記載します。



凡例:

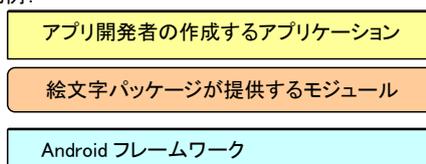


図 1 ソフトウェア構成の概略

	機能ブロック名	モジュール名	説明
①	アプリケーション	-	アプリケーション開発者が開発する Android アプリケーションです。 アプリケーション開発者は、アプリケーションに絵文字パッケージの提供するモジュール「②絵文字描画モジュール」を組み込みます。
②	絵文字描画モジュール	emojis1_type2.jar	「②絵文字描画モジュール」は絵文字パッケージがアプリ開発者に提供するモジュールで、アプリケーションから TextView 、 WebView で絵文字（キャリア絵文字、デコメ絵文字）・デコメピクチャの入力・表示を行うためのものです。
③	Android フレームワーク	-	Android 端末に搭載されている Android のフレームワークを指します。Android フレームワークには、絵文字の入力・表示に使用する TextView や WebView 、デコメ絵文字データを格納するためのデータベースなどを含みます。
④	TextView	-	Android フレームワークに含まれる TextView です。絵文字描画モジュールを用いることにより、絵文字の入力、表示が可能です。
⑤	WebView	-	Android フレームワークに含まれる WebView です。絵文字描画モジュールを用いることにより、絵文字の表示が可能です。
⑥	デコメ絵文字データ	-	Android フレームワークに含まれるデータベースに格納されたデコメ絵文字データを指します。
⑦	絵文字マネージャ	DecoEmojiManager.apk	絵文字パッケージに対応した Android 端末には、絵文字マネージャが組み込まれています。絵文字マネージャは、デコメ絵文字データの追加、削除を検知し、変換候補やデコメ絵文字一覧に表示するデコメ絵文字を追加、削除します。

4.2. 絵文字パッケージで提供するモジュール

絵文字パッケージでは、アプリケーション開発者に対し以下のモジュールを提供します。

4.2.1. 絵文字描画モジュール

絵文字描画モジュール (emojis1_type2.jar) は、Android アプリケーションでキャリア絵文字・デコメ絵文字を表示・入力するために必要なモジュールです。

キャリア絵文字・デコメ絵文字を表示・入力するアプリケーションは、絵文字描画モジュールをアプリケーションに含めた上で、アプリケーションに処理を追加する必要があります。

絵文字描画モジュールには、下記のクラスを含んでいます。

EmojiAssist クラス (jp.co.omronsoft.android.emoji パッケージ)

EmojiDrawable クラス (jp.co.omronsoft.android.text パッケージ)

DecoEmojiSpan クラス (jp.co.omronsoft.android.text.style パッケージ)

各クラスのAPIの詳細は「9.API仕様」を参照してください。

5. 絵文字の入力と表示

本章では、アプリケーションから絵文字(キャリア絵文字、デコメ絵文字)・デコメピクチャの入力及び表示を行うために、アプリケーションで必要な対応について記載します。

TextView と WebView では対応方法が異なるため、それぞれについて説明します。

5.1. TextViewでの絵文字の入力・表示

絵文字パッケージを組み込んだアプリケーションは、TextView を用いて、キャリア絵文字、デコメ絵文字を含む文字列を入力及び表示することができます。以下にその方法を記述します。

5.1.1. 絵文字入力可能なエディタの生成

TextView の subclasses である EditText で絵文字(キャリア絵文字、デコメ絵文字)・デコメピクチャを入力する場合は、絵文字入力許可フラグを設定します。

具体的には、"allowEmoji"(キャリア絵文字)、“allowDecoEmoji”(デコメ絵文字)の Bundle を true にすることで、設定した EditText での絵文字入力が可能になります。

- allowEmoji フラグ
EditText において、キャリア絵文字の入力を可能にする属性
- allowDecoEmoji フラグ
EditText において、デコメ絵文字の入力を可能にする属性

"allowEmoji"(キャリア絵文字)、“allowDecoEmoji”(デコメ絵文字)の Bundle を true にすることで、設定した EditText での絵文字の入力が可能になります。

サンプルコード:

```
EditText text = new EditText(this);
Bundle b = text.getInputExtras(true);
b.putBoolean("allowEmoji", true);
b.putBoolean("allowDecoEmoji", true);
```

5.1.2. EmojiAssistに組み込み、絵文字表示を可能にする

絵文字を表示する TextView (EditText) の生成時、そのインスタンスを EmojiAssist に組み込みます。

EmojiAssist は絵文字パッケージの一部として提供されるプラグイン機能であり、次の機能を持ちます。

- (1) TextView に絵文字・デコメ絵文字画像を自動的に表示させます。
- (2) 設定した TextView を定期的に更新(invalidate)し、デコメ絵文字をアニメ動作させます。
- (3) 設定した TextView の文字列内容が(IMEからの入力などで)更新されたとき、更新内容にデコメ絵文字が含まれるかどうかをチェックし、含まれる場合は TextView で処理可能なオブジェクト(DecoEmojiSpan : android.text.style.ReplacementSpan の継承クラス)に変換します。DecoEmojiSpan は絵文字パッケージとして提供されるクラスで、デコメ絵文字の表示を行います。

サンプルコード:

```
// TextView のインスタンスを取得する
TextView emojiView1 = findViewById(R.id.emoji_view1);
TextView emojiView2 = findViewById(R.id.emoji_view2);

// EmojiAssist に TextView を割り当てる
EmojiAssist ea = EmojiAssist.getInstance();
ea.addView(emojiView1);
ea.addView(emojiView2);

// EmojiAssist のアニメ動作を開始させる
ea.startAnimation();
```

5.1.3. 文字列からデコメ絵文字を取得

EditText の文字列を取得して処理を行う場合、デコメ絵文字(DecoEmojiSpan)が含まれるかどうかをチェックし、デコメ絵文字が含まれる文章に対応することが必要な場合があります。デコメ絵文字の実体は画像ファイル (gif、jpeg ファイル) ですので、DecoEmojiSpan からデコメ絵文字の URI 情報を取り出します。

サンプルコード:

```
android.text.Spanned sp = (Spanned)edittext.getText();
DecoEmojiSpan[] decoEmojiSpans = sp.getSpans(0, sp.length(), DecoEmojiSpan.class);
String uri = decoEmojiSpans[0].getURI();
```

アプリケーションは、入力/表示において通常文字とキャリア絵文字・デコメ絵文字を区別する必要はありません。しかし、アプリケーションがデコメ絵文字の混在した文章を保存、出力(メールへの添付など)を行う際には、デコメ絵文字についてはファイルとして扱う必要があります。そのため、絵文字パッケージではデコメ絵文字をファイルとして扱うための URI を取得する機能を提供します。

5.1.4. アプリケーションからキャリア絵文字を含む文字列を出力する際の注意事項

キャリア絵文字は各携帯電話事業者がそれぞれ独自に文字コードを割り当てており、携帯電話事業者間の互換性はありません。そのため、アプリケーションからキャリア絵文字を含む文字列を出力する場合、受け取り側のアプリケーションの動作環境によっては、必ずしもキャリア絵文字が正しく表示されるとは限らない点に注意してください。

絵文字パッケージでは、キャリア絵文字の文字コードは Google PUA を用いてフォントを割り当てています。具体的には、以下の URL の表に従い割り当てています。

<http://www.unicode.org/~scherer/emoji4unicode/snapshot/full.html>

そのため、キャリア絵文字を正しく表示するには、アプリケーションにおいてキャリア絵文字の携帯電話事業者文字コード(SJIS、JIS、Unicodeの携帯電話事業者PUA)をGoogle PUAに割り当てられている絵文字の文字コードに適切に変換する必要があります。また、キャリア絵文字を端末外に出力する場合には、アプリケーションにおいて出力先で適切に表示される文字コード(SJIS、JIS、Unicodeの携帯電話事業者PUA)に変換する必要があります。

5.2. WebViewへの絵文字の表示

WebView では表示のみが可能であり、たとえば HTML メール の Preview 画面として用いられます。

WebView にキャリア絵文字を表示するためには、絵文字パッケージの提供するデコ絵マッピング機能を使用します。また WebView にデコ絵文字を表示するためには、HTML 表示機能を利用してデコ絵文字の gif、jpeg ファイルを表示します。

5.2.1. キャリア絵文字の表示

WebView にキャリア絵文字を表示するための機能として、デコ絵マッピング機能を提供します。デコ絵マッピングとは、EmojiFactory に依存しない WebView 絵文字表示機能です。

Android 標準フレームワークで提供される EmojiFactory は、TextView と WebView に絵文字を表示する機能です。しかし、EmojiFactory が実装されていない Android 端末が存在するため、絵文字パッケージでは EmojiFactory に依存せずに WebView で絵文字を表示可能とする仕組みを提供します。

具体的には以下の方法で絵文字を表示してください。

- (1) 表示する文字列にキャリア絵文字が含まれているか判断してください。
キャリア絵文字が含まれているか判断するためには、EmojiDrawable クラスの isEmoji() メソッドを使用します。
- (2) キャリア絵文字が含まれる場合、キャリア絵文字を絵文字フォントデータ (gif/png 画像のバイナリデータ) に変換してください。
絵文字フォントデータに変換するには、EmojiAssist クラスの getEmojiImage() メソッドを使用します。
- (3) HTML 表示機能を利用して、取得した絵文字フォントデータを表示してください。

サンプルコード：

```
StringBuffer htmlBuf = new StringBuffer(html); // html は HTML ソースを所持する String

for (int i = 0; i < htmlBuf.length(); i++) {
    // HTML から絵文字コードを検出する
    int theCode = htmlBuf.codePointAt(i);
    if (Character.isHighSurrogate(htmlBuf.charAt(i))
        && EmojiDrawable.isEmoji(theCode)) {
        byte[] imgData = ea.getEmojiImage(theCode, 24); /* ea は EmojiAssist のインスタンスとする */

        // imgData を GIF ファイル名 imgFile に書き込む(詳細略)

        // HTML を生成する
        String mappingHTML = "";

        // 当該キャリア絵文字の文字コード部分を、img タグに置換する
        htmlBuf.replace(i, i+2, mappingHTML);

        // サロゲートペアを mappingHTML に置換したため、文字数分インクリメントする
        i = i + 2 - 1 + mappingHTML.length();
    } // else do nothing
}
```

5.2.2. デコメ絵文字の表示

WebView にデコメ絵文字を表示するには、HTML 表示機能を利用して、Android 端末に保存されているデコメ絵文字画像ファイル(gif、jpeg)を表示してください。

絵文字パッケージ V1.2 以上の適用端末では、デコメ絵文字だけでなく、デコメピクチャも取得できるようになります。デコメ絵文字とデコメピクチャでは、表示ルールに以下の違いがあります。

- デコメ絵文字: 通常の文字と同じサイズ(高さ)に拡張をして表示する。
- デコメピクチャ: 画像ファイルの元のサイズで表示する。

このような表示を可能にするため、TextView から画像ファイルを取り出す際に、デコメコンテンツの絵文字タイプ(デコメ絵文字かデコメピクチャかの区別)と画像サイズを取得する必要があります。

サンプルコード:

```
android.text.Spanned sp = (Spanned)edittext.getText();
DecoEmojiSpan[] decoEmojiSpans = sp.getSpans(0, sp.length(), DecoEmojiSpan.class);
String uri = decoEmojiSpans[0].getURI(); // URI
int width = decoEmojiSpans[0].getWidth(); // 幅
int height = decoEmojiSpan[0].getHeight(); // 高さ
boolean isPicture= decoEmojiSpan[0].isTypePicture(); // 絵文字タイプ
```

5.3. 絵文字パッケージをサポートしないAndroid端末での動作について

絵文字パッケージをサポートしない Android 端末において、本章で述べた TextView での絵文字の入力・表示や WebView への絵文字の表示を行おうとした場合、単に絵文字が表示されない動作となります。

従って、本ライブラリを組み込んだアプリケーションで Exception の処理などの特別な対応は必要ありません。

6. デコメ絵文字の登録、削除

絵文字パッケージをサポートする Android 端末では、変換候補として表示するデコメ絵文字や、デコメ絵文字選択パレットに表示するデコメ絵文字をアプリケーションから追加・削除することができます。

6.1. 登録先ディレクトリ

デコメ絵文字を追加する場合、アプリケーションから所定のディレクトリにデコメ絵文字ファイルを格納してください。所定のディレクトリにファイルが格納されたことは絵文字パッケージが自動的に検出し、デコメ絵文字を変換候補とパレットに追加します。所定のディレクトリ(「登録先ディレクトリ」と呼びます)については各携帯電話事業者の Web サイトを参照してください。

デコメ絵文字を削除する場合、アプリケーションから登録先ディレクトリに置かれているデコメ絵文字ファイルを削除してください。ファイルが削除されたことは絵文字パッケージが自動的に検出し、デコメ絵文字を変換候補とパレットから削除します。

6.2. デコメ絵文字の定義

絵文字パッケージでは、以下の両方の条件を満たす場合にデコメ絵文字用の画像ファイルとみなします。

- ・ ファイルフォーマット: jpeg、gif
- ・ サイズ: 20x20(pixel)

サイズ: 20x20(pixel)以外の場合はデコメピクチャとして扱います。

7. 絵文字利用アプリケーションの拡張機能

絵文字パッケージの拡張機能を利用すると、単に、絵文字の入出力と表示にとどまらず、絵文字を利用した様々な応用機能を作ることができます。

応用機能の開発のためには、絵文字描画モジュール (emojisl_type2.jar) だけではなく、絵文字マネージャ (DecoEmojiManager) とのインタフェースを利用することが必要になる場合があります。絵文字マネージャとのインタフェースについては、「絵文字パッケージ対応 IME 開発ガイド」の「API 仕様」を参照いただく必要があります。

7.1. デコメ絵文字拡張情報の利用

デコメ絵文字の読み付加情報には、コンテンツプロバイダ (以下CP) が独自のサービスを拡張するためのカスタムタグを設けることができます。アプリケーションは、デコメ絵文字に付加されたカスタムタグとその内容 (拡張情報) を読み取り、それを独自の応用に利用することができます。読み付加情報については、「10 デコメ絵文字の読み付加情報フォーマット」を参照してください。

以下に、選択されると音楽を鳴らすデコメ絵文字の例を示します。

- (1) CPは、提供するデコメ絵文字の読み付加情報にカスタムタグ”melody”を定義し、拡張情報「/mnt/sdcard/aaa.ogg」(音楽ファイルのURI) を埋め込みます。

■ サンプルコード

```

<emoji>
<label><name>えがお</name><note>笑顔</note></label>
<category>感情</category>
<tags name="melody">/mnt/sdcard/aaa.ogg</tags>
</emoji>

```

(2) コンテンツの提供と同時に、CP は、拡張情報を扱うアプリケーションを提供します。

アプリケーションは、入力された文字列の中の拡張情報("melody")を持つデコメ絵文字を選択されると、デコメ絵文字の拡張情報を取得し、その内容に従い、音楽ファイルを開いて、メロディを流します。

■ サンプルコード

```

// デコメ絵文字の ID(URI)を得る
android.text.Spanned sp = (Spanned)edittext.getText();
DecoEmojiSpan[] decoEmojiSpans = sp.getSpans(0, sp.length(), DecoEmojiSpan.class);
String uri = decoEmojiSpans[0].getURI();
// デコメ絵文字 URI とカスタムタグ名から、拡張情報の内容 (音楽ファイルの URI)を得る
List<String> melodylist = new ArrayList<String>();
int ret = aidl_getTagInfo(uri, "melody", melodylist);
// 音楽ファイルを使って、メロディを流す
mp = new MediaPlayer(); // Media Player
mp.setDataSource(melodylist[0]);
mp.prepare();
mp.start();

```

【注】aidl_getTagInfo() は、絵文字マネージャとのインタフェースAPIです。その詳細については「絵文字パッケージ対応 IME 開発ガイド」の「API 仕様」を参照ください。

このように、CP は拡張情報付きのデコメ絵文字と利用アプリケーションをともに提供することによって、独自のサービスを実現することができます。

8. 注意事項

8.1. スリープ時の注意事項

絵文字パッケージでは EmojiAssist (jar) において、絵文字表示中の場合、一定時間ごとに描画処理を行っています。そのため、描画処理を明示的に停止させなければ、スリープ時に消費電力が大きくなります。

スリープ時に行うべき処理を以下に記載します。

8.1.1. 対応策 (アクティビティアプリケーション向け)

絵文字パッケージを使用するアクティビティアプリケーションにおいては、スリープモード遷移時に `onPause()` 関数がコールされます。この関数内において、EmojiAssist に対し、以下のように描画処理を停止させる必要があります。

```
@Override
protected void onPause() {
    --- 中略 ---
    mEmojiAssist.stopAnimation();
    --- 中略 ---
}
```

なお、描画処理を開始／再開させる場合には、`EmojiAssist#startAnimation()` をコールする必要があります。

8.1.2. 対応策 (サービスアプリケーション向け (IME を含む))

絵文字パッケージを使用するサービスアプリケーション (IME を含む) の場合、スリープモード遷移時に `onPause()` 関数はコールされないため、`BroadcastReceiver` にてスクリーン OFF を検知し、EmojiAssist に対し、以下のように描画処理を停止させる必要があります。

- `BroadcastReceiver` において、「ACTION_SCREEN_ON」を検知し、`startAnimation()` をコールする
- `BroadcastReceiver` において、「ACTION_SCREEN_OFF」を検知し、`stopAnimation()` をコールする

以下に、`BroadcastReceiver` を使ったサンプルコードを記載します。

```
private BroadcastReceiver mReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (mIsEmojiAssistWorking) { // EmojiAssist が動作している場合
            if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
                mEmojiAssist.startAnimation();
            } else if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
                mEmojiAssist.stopAnimation();
            }
        }
    }
};
```

8.2. ハードウェアアクセラレーションに関する制限事項

絵文字パッケージ v1.2 では、下記の [発生条件] を全て満たすアプリにおいて、端末にてハードウェアアクセラレーションを ON に設定（設定→開発者向けオプション→GPU レンダリングを使用 のチェックを ON に設定）すると、EmojiAssist がキャリア絵文字／デコメ絵文字を正しく描画できません。

[発生条件]

AndroidManifest.xml の設定において、以下の両方の条件を満たす場合、本現象が発生します。

1. アプリ側の「android:minSdkVersion」、または「android:targetSdkVersion」が「14」以上の場合。
2. アプリ側に「android:hardwareAccelerated="false"」の記述がない場合。

[原因]

ハードウェアアクセラレーション ON 設定時に、Android Framework の API のうち、非対応の API / クラスが存在すると報告されており、絵文字パッケージはその中の Movie クラスを使用しているため、正しく描画を行うことができません。

[調査環境]

Android 4.0.4

[対策]

アプリ側で「android:hardwareAccelerated="false"」の設定を追加する必要があります。

9. API仕様

絵文字パッケージのライブラリがアプリケーションに対し公開する API を記載します。

9.1. TextView/EditText（絵文字入力の有効化）

アプリで使用する TextView(EditText)で、キャリア絵文字、デコメ絵文字を入力したい場合は、下記のように絵文字有効フラグ (allowEmoji) 及びデコメ絵文字有効フラグ (allowDecoEmoji) を Bundle する必要がある。それぞれ個別にも指定可能である。

```
EditText text = new EditText(this);
Bundle b = text.getInputExtras(true);
b.putBoolean("allowEmoji", true);
b.putBoolean("allowDecoEmoji", true);
```

9.2. EmojiAssistクラス

パッケージ名: jp.co.omronsoft.android.emoji

9.2.1. インスタンス取得

API 名		getInstance			
説明		EmojiAssistのインスタンスを取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	EmojiAssist	EmojiAssist のインスタンス			
前提					
なし					
概略説明					
EmojiAssistのインスタンスを取得する。					

9.2.2. TextView登録

API名		addView			
説明		TextViewを登録する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	view	TextView	I	Must	登録するTextView
返値					
No	型	説明			
前提					
getInstance()で取得したEmojiAssistで実行すること。					
概略説明					
TextViewをEmojiAssistに登録する。					
補足					
同一プロセス内で絵文字を描画されるTextViewは全てEmojiAssistに登録すること。 アニメーション動作中に実行してもよい。					

9.2.3. TextView削除

API名		removeView			
説明		登録したTextViewを削除する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	view	TextView	I	Must	削除するTextView
返値					
No	型	説明			
前提					
TextView が登録済みであること					
概略説明					
登録したTextViewをEmojiAssistから削除する。					
補足					
アニメーション動作中に実行してもよい。					
登録されていないviewが指定された場合は、無視される。					
終了処理などで登録したTextViewの更新が不要になる場合は、removeViewなどで登録を解除すること。					

9.2.4. TextView全削除

API名		clearView			
説明		登録したTextViewを全て削除する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
前提 なし					
概略説明 登録したTextViewをEmojiAssistから全て削除する。					
補足 アニメーション動作中に並行実行してもよい。 終了処理などで登録したTextViewの更新が不要になる場合は、clearViewなどで登録を解除すること。					

9.2.5. アニメーション開始

API 名		startAnimation			
説明		絵文字のアニメーションを開始する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
前提 なし					
概略説明 絵文字のアニメーションを開始する。addViewで登録したTextViewに対して定期的にinvalidate()するタイマーを開始させる。TextViewのアニメーションが不要になるときはstopAnimationにより停止させる必要がある。					

9.2.6. アニメーション停止

API 名		stopAnimation			
説明		絵文字のアニメーションを停止する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
前提					
なし					
概略説明					
絵文字のアニメーションを停止する。					
スリープモード遷移時など、アニメーションが不要になるときは、本 API により停止させること。					

9.2.7. 絵文字データ取得

API 名		getEmojiImage			
説明		キャリア絵文字の画像データ(Binary)を取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	code	int	I	Must	対象文字のAndroidPUA
2	textsize	float	I	Must	フォントサイズ (文字の高さ) [pixel]
返値					
No	型	説明			
1	byte[]	絵文字データ(Binary)(失敗時 : NULL)			
前提 なし					
概略説明 指定された文字コードのキャリア絵文字について、指定サイズにもっとも近いサイズのGIF / PNG 画像データ(Binary)を取得する。 例：キャリア絵文字として、下記の5種類が用意されている場合；					
<ul style="list-style-type: none"> ・ 12pixel: 14 pixel以下の時に表示する画像 ・ 16pixel: 15～18 pixelの時に表示する画像 ・ 20pixel: 19～22 pixelの時に表示する画像 ・ 24pixel: 23～26 pixelの時に表示する画像 ・ 28pixel: 27 pixel以上の時に表示する画像 					
<p>サイズ17を指定すると</p> <ul style="list-style-type: none"> ・ 15～18 pixelの時に表示する画像 <p>が取得される。</p> <p>なお、標準の絵文字パッケージでは48x48の画像のみ用意されている。</p>					

サンプルコード:

```
StringBuffer htmlBuf = new StringBuffer(html); // html はHTML ソースを所持する String

for (int i = 0; i < htmlBuf.length(); i++) {
    // HTML から絵文字コードを検出する
    int theCode = htmlBuf.codePointAt(i);
    if (Character.isHighSurrogate(htmlBuf.charAt(i))
        && EmojiDrawable.isEmoji(theCode)) {
        byte[] imgData = ea.getEmojiImage(theCode, 24); /* ea はEmojiAssist のインスタンスとする */

        // imgData を GIF ファイル名 imgFile に書き込む(詳細略)

        // HTML を生成する
        String mappingHTML = "<img src=¥" + imgFile + "¥>";

        // 当該キャリア絵文字の文字コード部分を、img タグに置換する
        htmlBuf.replace(i, i+2, mappingHTML);

        // サロゲートペアを mappingHTML に置換したため、文字数分インクリメントする
        i = i + 2 - 1 + mappingHTML.length();
    } // else do nothing
}
```

9.2.8. デコメピクチャ拡縮表示設定

API 名		setPictureScale			
説明		デコメピクチャに対し、デコメ絵文字と同様の拡縮表示を行う設定を行う。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	scale	boolean	I	Must	デコメピクチャ縮小表示可否の指定 true : デコメピクチャを拡縮して表示 false : デコメピクチャを等倍で表示
返値					
No	型	説明			
前提					
addView()で TextView を登録する前に実行すること。					
概略説明					
アプリ側において、表示フィールドのサイズなどの制限によりデコメピクチャの等倍表示ができない際に、デコメピクチャの拡縮表示設定を行うことができる。					
拡縮は、デコメ絵文字と同様に、表示フィールドの文字サイズ(高さ)に合わせて行われる。					
本 API を利用していない場合には、デコメピクチャは等倍で表示される。					

9.3. DecoEmojiSpanクラス

デコメ絵文字情報を保持し、描画を受け持つクラスである。

パッケージ名:jp.co.omronsoft.android.text.style

9.3.1. URI取得

API名		getURI			
説明		デコメ絵文字の画像ファイルを指すURIを返却する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	String	URI			
前提 なし					
概略説明 デコメ絵文字の画像ファイルを指すURIを返却する。					

サンプルコード:

```
CharSequence charSeq = text.getText();
SpannableStringBuilder ssb = SpannableStringBuilder.valueOf(charSeq);
DecoEmojiSpan[] decoEmojiSpans = ssb.getSpans(0, ssb.length(), DecoEmojiSpan.class);
String uri = decoEmojiSpan[0].getURI();
```

9.3.2. 幅取得

API 名		getWidth			
説明		デコメ絵文字の画像ファイルの幅を取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	int	幅			
前提 なし					
概略説明 デコメ絵文字の画像ファイルの幅を取得する。					

9.3.3. 高さ取得

API名		getHeight			
説明		デコメ絵文字の画像ファイルの高さを取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	int	高さ			
前提 なし					
概略説明 デコメ絵文字の画像ファイルの高さを取得する。					

9.3.4. デコメピクチャ判定

API 名		isTypePicture			
説明		絵文字種別がデコメピクチャか否かを判定する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	boolean	true : デコメピクチャ false : デコメピクチャ以外			
前提 なし					
概略説明 絵文字種別がデコメピクチャか否かを判定する。					

サンプルコード:

```
CharSequence charSeq = text.getText();
SpannableStringBuilder ssb = SpannableStringBuilder.valueOf(charSeq);
DecoEmojiSpan[] decoEmojiSpans = ssb.getSpans(0, ssb.length(), DecoEmojiSpan.class);
boolean ret = decoEmojiSpans[0].isTypePicture();
```

9.4. EmojiDrawableクラス

パッケージ名:jp.co.omronsoft.android.text

9.4.1. 絵文字判定

API名	isEmoji				
説明	引数の文字列に絵文字が含まれているか否かを判定する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	text	CharSequence	I	Must	判定対象文字列
返値					
No	型	説明			
1	static boolean	true : 絵文字が含まれている			
前提 なし					
概略説明 引数の文字列に絵文字が含まれているか判定し、結果を返却する。					

9.4.2. 絵文字判定(コード)

API名	isEmoji				
説明	引数の文字が絵文字か否かを判定する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	code	int	I	Must	判定対象文字
返値					
No	型	説明			
1	static boolean	true : 絵文字である			
前提 なし					
概略説明 引数の文字が絵文字かどうか判定し、結果を返却する。 引数にはString.codePointAt()等で取得したコードを渡す。					

10. デコモ絵文字の読み付加情報フォーマット

デコモコンテンツに付加する「読み付加情報」のデータフォーマットを以下に記載します。タグは将来拡張する可能性があります。絵文字パッケージでは、デコモ絵文字として用いられる GIF および JPEG ファイルにこれを格納することによって、デコモ絵文字がプリインストールまたはダウンロードされた Android 端末で、デコモ絵文字を IME の予測・変換で用いられるように自動的に登録します。読み付加情報は、画像ファイルの本来の利用を妨げない領域に書きこまれます。

・ GIF ファイルフォーマット

Header	シグネチャ/バージョン/画像サイズ/カラーテーブルなど
Block 1	Application Extension
	Graphic Control Extension - Image Block (0xf92c)
	: (複数の Image Block で Animated GIF のデータを格納)
	block terminator (0x00)
:	
Block n	Extension Block - Comment Extension (0x21fe)
	block size
	[付加情報] を格納 [最大 255byte]
	: 「付加情報」は複数定義可能、ブロックサイズを超える場合、ブロックを追加。
	block terminator (0x00)
Trailer	(0x3b)

・ JPEG ファイルフォーマット

SOI SEGMENT	シグネチャ/バージョン/画像サイズ/カラーテーブルなど
APPO SEGMENT	marker (0xff 0xe0) JFIF application segment
	length(x)
	data x type
:	
COM SEGMENT	marker (0xff 0xe0)
	Length
	[付加情報] を格納 [最大 10240byte]
EOI SEGMENT	(0xff 0xd9)

◆付加情報定義フォーマット

```
<emoji>
<label><name>読み</name> [<part>品詞</part>][<note>表記</note>]</label>
[<category>カテゴリ</category>]
[<type>タイプ</type>]
[<tags name="カスタム">拡張情報</tags>]
</emoji>
```

◆付加情報文字列の上限

上限は 10,240bytes とする。

◆付加情報文字列の分割

付加情報が GIF の Block のサイズ (255byte) に収まらない場合、複数の Block を併用し記述することとする。先頭 Block は必ず <emoji> タグで始まるものとする。

◆文字コード

文字コードは UTF-8 とする。

◆タグ格納例

```
<emoji><label><name>えがお</name><note>笑顔</note></label>
<label><name>わらう</name><part>動詞</part></label>
<category>感情</category>
<type>emoji</type>
<tags name="melody">/mnt/sdcard/aaa.ogg</tags></emoji>
```

タグ名	項目	出現性 (個数)	内容
emoji	絵文字定義	必須 (1)	画像ファイルを絵文字として扱うための付加定義情報を記述する。 一つの画像ファイルに対し、emoji は 1 つだけ記述可能とする。 emoji には、複数の label を含めることができる。
label	ラベル	省略可 (0-10)	読み情報のセットをくくるためのタグ。絵文字は読み情報を複数持つことができる。 (最大 10 個。) label 内で name は省略できない。
name	読み	必須 (1)	読み文字列を格納する。(24 文字まで) 読みは原則ひらがなのみで記述する。 辞書の読みとして漢字等を使用したい場合は、それらの文字を含めてもよい。(フォーマットとしては特に制限は設けない。)ただし、予測・変換などを行った場合の動作は IME に依存する。通常の IME であれば、読みはひらがなのみが有効である。
part	品詞	省略可 (0-1)	品詞名を格納する。 日本語向け品詞一覧は別途定義する。 品詞を指定しない場合は part を省略可とする。 (品詞を指定することにより、IME がより適切な予測・変換を行うことを可能にする。)
note	表記	省略可 (0-1)	表記文字列を格納する。(24 文字まで) 表記を指定しない場合は note を省略可とする。 (IME が本タグで記述されている文字列を解析し、最適な品詞情報を取得し、自然なつながりを実現することを可能にする。)
category	カテゴリ	省略可 (0-n)	カテゴリを格納する。 1 つの emoji 内に複数のカテゴリを記述可能とする。 カテゴリを指定しない場合は category を省略可とする。 (Android 端末内の該当するカテゴリに分類することを可能にする。)
type	タイプ	省略可 (0-1)	画像ファイルが「デコメ絵文字」、「デコメピクチャ」のいずれであるかを格納する。 「デコメ絵文字」である場合は、「emoji」を指定し、「デコメピクチャ」である場合は「picture」を指定する。省略時は「emoji」を指定したものとみなされる。 なお、本項目は現在 Android 端末では未使用。
tags	カスタムタグ	省略可 (0-n)	カスタムタグを指定する。タグ名はname属性により指定する。 (例) <tags name="melody">/mnt/sdcard/aaa.ogg</tags> カスタムタグは自由に定義することができ、コンテンツプロバイダは、固有の拡張情報を記述することができる。 (例では、このコンテンツは「melody」というカスタムタグを持ち、内容として音声ファイルのURIを持つ。アプリを用意すると、たとえばコンテンツクリックによって、音声を鳴らすなどの拡張機能を持たせることができる。) ファイル内で、カスタムタグは任意の個数を指定することができるが、カスタムタグ名は最大 10 種類とする。また、同一タグ名の拡張情報は最大 10 個とする。

◆読み付加情報フォーマット DTD(参考)

```
<!-- Decoration Emoji Character Data Extension ( Version 1.2 ) -->
<!DOCTYPE emoji [
  <!ELEMENT emoji (label*, category*, type?, tags*) >
    <!ELEMENT label ((name, part?, note?) | (name, note?, part?)) >
      <!ELEMENT name (#PCDATA) >
      <!ELEMENT part (#PCDATA) >
      <!ELEMENT note (#PCDATA) >
    <!ELEMENT category (#PCDATA) >
    <!ELEMENT type (#PCDATA) >
    <!ELEMENT tags (#PCDATA) >
      <!ATTLIST tags name CDATA #REQUIRED >
  ] >
```

◆日本語向け品詞一覧

品詞*	説明
名詞	普通名詞
名詞-固有名詞	固有名詞
名詞-代名詞	いわゆる代名詞, 不定語. 例: 「それ」「ここ」「あいつ」「あなた」「みんな」「わたし」
名詞-副詞可能	曜日, 月など時間を表す副詞的な用法を持つ名詞. 量や割合などを表し副詞的に使うことのできる名詞. 例: 「金曜」「一月」「午後」「少量」…
名詞-サ変接続	後ろに「する」「できる」「なさる」「くださる」などが後接することができる名詞. 例: 「インプット」「愛着」「悪化」「悪戦苦闘」「一安心」「下取り」…
名詞-形容動詞語幹	形容動詞語幹で, 「な」の前に現れるもの. 例: 「健康」「安易」「駄目」「だめ」…
名詞-ナイ形容詞語幹	助動詞の「ない」の直前に現れて形容詞的な働きをする語 例: 「申し訳」「仕方」「とんでも」「違い」…
名詞-数	漢数字, 算用数字, および, 「何(回)」「数(%)」「幾(夜)」. 例: 「0」「1」「2」「何」「数」「幾」…
名詞-接尾	一般に「接尾語」というよりも範囲が広く, 複合名詞の後ろ要素として用いられることが普通なもの. 例: 「君」「様」「町」「市」「県」「個」「つ」「本」「冊」「パーセント」「cm」「kg」「カ月」「時間」「時半」…
名詞-接続詞的	単語と単語を接続する接続詞的な働きをするもの. 例: 「(日本)対(アメリカ)」「対(アメリカ)」「(3)対(5)」「(女優)兼(主婦)」
接頭詞	名詞(形容動詞語幹を含む)に前接する接頭語, 数に接続するもの. 動詞の命令形あるいは[動詞連用形]+「なる/なさる/くださる」に前接する接頭語. 例: 「お(水)」「某(氏)」「同(社)」「故(～氏)」「約」「およそ」「お(読みなさい)」「お(座り)」
動詞	動詞
形容詞	形容詞
副詞	必ず後ろで切れるもの, 連体修飾が不可能な副詞. 例: 「あいかわらず」「多分」など
連体詞	名詞を修飾する形しかもたないもの. 例: 「この」「その」「あの」「どの」「いわゆる」「なんらかの」「何らかの」「いろんな」「こういう」「微々たる」「堂々たる」「単なる」「いかなる」「我が」「同じ」「亡き」…
接続詞	独立に現れる接続詞. 例: 「が」「けれども」「そして」「じゃあ」「それどころか」…
感動詞	感動詞. あいさつなど. 例: 「おはよう」「おはようございます」「こんにちは」「こんばんは」「ありがとう」「どうもありがとう」「ありがとうございます」「いただきます」「ごちそうさま」「さよなら」「さようなら」「はい」「いいえ」「ごめん」「ごめんなさい」…
記号	一般的な記号. 例: 「○」「◎」「@」「\$」「〒」「→」「+」など

*品詞はオープンな品詞体系「IPADIC」を参考に抽出した。

改定履歴		
V1.0.0	2011/9/30	新規作成
V1.1.0	2012/3/23	V1.1(ICS)対応 API 仕様の追加・修正と、サンプルコードの修正 スリープモード遷移時にアニメーション停止の必要なことを記述
V1.2.0	2012/9/14	V1.2 対応
V1.2.1	2012/12/3	デコメ絵文字画像の高詳細化に関する記載を見直し