

## 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. 1.1.0

オムロン ソフトウェア株式会社

2012/03/23

**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	8
7. スリープ時の注意事項【Level A/B/C】	9
7.1. 対応策 (アクティビティアプリケーション向け)	9
7.2. 対応策 ((サービスアプリケーション向け (IME を含む))	9
8. API仕様	10
8.1. TextView/EditText (絵文字入力の有効化)	10

---

<b>8.2. EmojiAssist</b> クラス	<b>11</b>
8.2.1. インスタンス取得	11
8.2.2. TextView登録	12
8.2.3. TextView削除	13
8.2.4. TextView全削除	14
8.2.5. アニメーション開始	15
8.2.6. アニメーション停止	16
8.2.7. デコメ絵文字Span設定	17
8.2.8. 絵文字データ取得	18
<b>8.3. DecoEmojiSpan</b> クラス	<b>19</b>
8.3.1. URI取得	19
<b>8.4. EmojiDrawable</b> クラス	<b>20</b>
8.4.1. 絵文字判定	20
8.4.1. 絵文字判定(コード)	21

## 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。 ※「デコモ絵文字」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)の入力及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はなく、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、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 V2.3(GB) および 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。

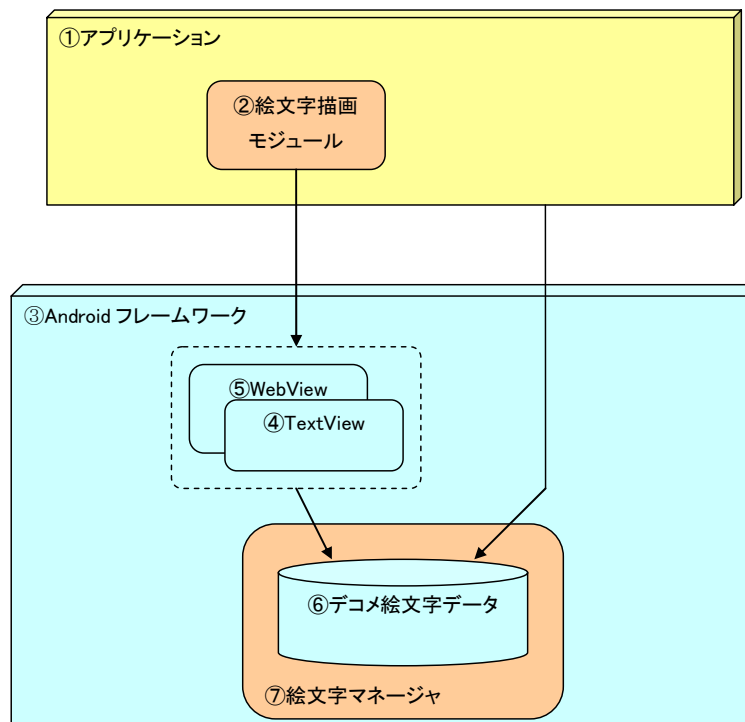
また、デコメ絵文字に読み情報を付加する方法は「デコメ絵文字コンテンツ作成ガイド」を参照してください。

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:

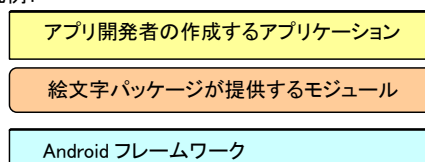


図 1 ソフトウェア構成の概略

	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 <b>Android</b> 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 <b>TextView</b> 、 <b>WebView</b> で絵文字（キャリア絵文字、デコメ絵文字）の入力・表示を行うためのものです。
③	<b>Android</b> フレームワーク	-	<b>Android</b> 端末に搭載されている <b>Android</b> のフレームワークを指します。 <b>Android</b> フレームワークには、絵文字の入力・表示に使用する <b>TextView</b> や <b>WebView</b> 、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	<b>TextView</b>	-	<b>Android</b> フレームワークに含まれる <b>TextView</b> 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	<b>WebView</b>	-	<b>Android</b> フレームワークに含まれる <b>WebView</b> 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	-	<b>Android</b> 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	-	絵文字パッケージに対応した <b>Android</b> 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8.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)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)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 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) 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。
- (2) 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出力、保存 (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 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。

### 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、絵文字が表示されない動作となります。

従って、アプリケーションには 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。

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。

なお、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)

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。

## 7. スリープ時の注意事項【Level A/B/C】

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。

### 7.1. 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。

### 7.2. 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. 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。

### 8.1. 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);
```

## 8.2. EmojiAssist クラス

パッケージ名:jp.co.omronsoft.android.emoji

### 8.2.1. インスタンス取得

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

## 8.2.2. TextView 登録

<b>API 名</b>		addView			
<b>説明</b>		TextViewを登録する。			
<b>引数</b>					
No	引数名	型	I/O	Must/Opt.	説明
1	view	TextView	I	Must	登録するTextView
<b>返値</b>					
No	型	説明			
<b>前提</b>					
getInstance()で取得したEmojiAssistで実行すること。					
<b>概略説明</b>					
TextViewをEmojiAssistに登録する。					
<b>補足</b>					
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。					

## 8.2.3. TextView 削除

<b>API 名</b>	removeView				
<b>説明</b>	登録したTextViewを削除する。				
<b>引数</b>					
<b>No</b>	<b>引数名</b>	<b>型</b>	<b>I/O</b>	<b>Must/Opt.</b>	<b>説明</b>
1	view	TextView	I	Must	削除するTextView
<b>返値</b>					
<b>No</b>	<b>型</b>	<b>説明</b>			
<b>前提</b>					
TextView が登録済みであること					
<b>概略説明</b>					
登録したTextViewをEmojiAssistから削除する。					
<b>補足</b>					
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。					



## 8.2.4. TextView 全削除

API 名	clearView				
説明	登録したTextViewを全て削除する。				
<b>引数</b>					
No	引数名	型	I/O	Must/Opt.	説明
<b>返値</b>					
No	型	説明			
<b>前提</b> なし					
<b>概略説明</b> 登録したTextViewをEmojiAssistから全て削除する。					
<b>補足</b> 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。					

### 8.2.5. アニメーション開始

<b>API 名</b>		startAnimation			
<b>説明</b>		絵文字のアニメーションを開始する。			
<b>引数</b>					
No	引数名	型	I/O	Must/Opt.	説明
<b>返値</b>					
No	型	説明			
<b>前提</b> なし					
<b>概略説明</b> 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。					

### 8.2.6. アニメーション停止

<b>API名</b>	stopAnimation				
<b>説明</b>	絵文字のアニメーションを停止する。				
<b>引数</b>					
<b>No</b>	<b>引数名</b>	<b>型</b>	<b>I/O</b>	<b>Must/Opt.</b>	<b>説明</b>
<b>返値</b>					
<b>No</b>	<b>型</b>	<b>説明</b>			
<b>前提</b> なし					
<b>概略説明</b> 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。					

## 8.2.7. デコメ絵文字 Span 設定

<b>API 名</b>	getDecoEmojiText				
<b>説明</b>	デコメ絵文字属性 (Annotation) を設定した文字を取得する。				
<b>引数</b>					
<b>No</b>	<b>引数名</b>	<b>型</b>	<b>I/O</b>	<b>Must/Opt.</b>	<b>説明</b>
1	uri	String	I	Must	デコメ絵文字の画像ファイル格納位置を指すURI
<b>返値</b>					
<b>No</b>	<b>型</b>	<b>説明</b>			
1	CharSequence	デコメ絵文字 Span を設定した文字			
<b>前提</b> なし					
<b>概略説明</b> IMEからアプリケーションに出力するデコメ絵文字属性を含むSpan (android.text.Annotation) を設定した文字を取得する。					

## 8.2.8. 絵文字データ取得

<b>API 名</b>		getEmojiImage			
<b>説明</b>		キャリア絵文字の画像データ(Binary)を取得する。			
<b>引数</b>					
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]
<b>返値</b>					
No	型	説明			
1	byte[]	絵文字データ(Binary)(失敗時 : NULL)			
<b>前提</b> なし					
<b>概略説明</b> 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"> <li>・ 12pixel: 14 pixel以下の時に表示する画像</li> <li>・ 16pixel: 15～18 pixelの時に表示する画像</li> <li>・ 20pixel: 19～22 pixelの時に表示する画像</li> <li>・ 24pixel: 23～26 pixelの時に表示する画像</li> <li>・ 28pixel: 27 pixel以上の時に表示する画像</li> </ul>					
<p>サイズ17を指定すると</p> <ul style="list-style-type: none"> <li>・ 15～18 pixelの時に表示する画像</li> </ul> <p>が取得される。</p>					

### 8.3. 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

#### 8.3.1. URI 取得

<b>API 名</b>		getURI			
<b>説明</b>		デコメ絵文字の画像ファイルを指すURIを返却する。			
<b>引数</b>					
No	引数名	型	I/O	Must/Opt.	説明
<b>返値</b>					
No	型	説明			
1	String	URI			
<b>前提</b> なし					
<b>概略説明</b> 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 = decoEmojiSpans[0].getURI();
```

## 8.4. EmojiDrawable クラス

パッケージ名:jp.co.omronsoft.android.text

### 8.4.1. 絵文字判定

<b>API 名</b>		isEmoji			
<b>説明</b>		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。			
<b>引数</b>					
No	引数名	型	I/O	Must/Opt.	説明
1	text	CharSequence	I	Must	判定対象文字列
<b>返値</b>					
No	型	説明			
1	static boolean	true : 絵文字が含まれている			
<b>前提</b> なし					
<b>概略説明</b> 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。					

## 8.4.1. 絵文字判定(コード)

<b>API 名</b>	isEmoji				
<b>説明</b>	引数の文字が絵文字か否かを判定する。				
<b>引数</b>					
<b>No</b>	<b>引数名</b>	<b>型</b>	<b>I/O</b>	<b>Must/Opt.</b>	<b>説明</b>
1	code	int	I	Must	判定対象文字
<b>返値</b>					
<b>No</b>	<b>型</b>	<b>説明</b>			
1	static boolean	true : 絵文字である			
<b>前提</b> なし					
<b>概略説明</b> 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。					



改定履歴		
V1.0.0	2011/9/30	新規作成
V1.1.0	2012/3/23	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