

絵文字パッケージ対応 IME 開発ガイド

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	5
2. 参照資料	5
2.1. 参照ドキュメント	5
3. 略語・記号の定義	5
4. 機能概要	6
5. 構成	6
6. 絵文字パッケージ	7
6.1. 絵文字パッケージでできること	7
6.2. 絵文字パッケージの機能レベル	7
6.3. 絵文字パッケージのI/F	8
7. <i>Application Programming Interface</i>	9
7.1. 入力/表示シーケンス	9
8. デコメ絵文字 <i>Data Interface</i>	13
8.1. デコメ絵文字の定義	13
8.2. デコメ絵文字の登録と削除	13
8.2.1. 登録先ディレクトリ	13
8.2.2. IMEとの関係	13
9. <i>IME Interface</i>	14
9.1. キャリア絵文字入力I/F (A機能)	14
9.2. デコメ絵文字一覧入力I/F (B機能)	15
9.3. デコメ絵文字予測/変換I/F (C機能)	16
10. 【実装手順】IMEからの絵文字パッケージ利用手順	18
10.1. IMEのキャリア絵文字一覧入力機能追加	18
10.2. IMEのキャリア絵文字予測/変換入力対応	18
10.3. IMEのデコメ絵文字一覧起動・入力の追加	18
10.3.1. デコメ絵文字一覧の起動	18
10.3.2. 入力されたデコメ絵文字の伝送	19

10.4. IMEのデコメ絵文字予測/変換入力対応	19
10.5. 候補画面の表示	19
10.5.1. TextViewをEmojiAssistに組み込み、絵文字表示を可能にする	19
10.5.2. 候補画面にデコメ絵文字を表示する	20
10.6. 入力フィールドの絵文字属性判定	20
10.6.1. キャリア絵文字入力許可フラグ (allowEmoji)	20
10.6.2. デコメ絵文字入力許可フラグ (allowDecoEmoji)	21
10.6.3. 一覧表示の順序制御	21
11. 【実装手順】デコメ絵文字読み情報更新手順	23
11.1. デコメ絵文字読み情報更新手順 (ContentProviderインタフェース)	23
11.2. デコメ絵文字読み情報更新手順 (aidlインタフェース)	24
11.2.1. デコメ絵文字辞書情報更新の基本手順	24
11.2.2. シーケンス図	25
12. 【実装手順】デコメ絵文字の一覧表示の更新	28
12.1. 一覧表示アプリケーションの表示更新	28
13. 【実装手順】IMEに関する付加情報	29
13.1. Android Framework全画面モード時の既存不具合対応	29
13.2. スリープ時の注意事項	29
13.2.1. 対応策 (アクティビティアプリケーション向け)	29
13.2.2. 対応策 (サービスアプリケーション向け (IME を含む))	30
14. 【API】インタフェース仕様	31
14.1. TextView/EditText (絵文字入力の有効化)	33
14.2. EmojiAssistクラス	33
14.2.1. インスタンス取得	33
14.2.2. TextView登録	34
14.2.3. TextView削除	35
14.2.4. TextView全削除	36
14.2.5. アニメーション開始	37
14.2.6. アニメーション停止	38
14.2.7. デコメ絵文字Span設定	39
14.2.8. デコメ絵文字Span設定(拡張版)	40
14.2.9. 絵文字データ取得	41
14.2.10. ExtractEditText差し替え	42
14.2.11. 入力文字列の絵文字有無判定	43
14.2.12. デコメ絵文字pop識別子表示	44
14.2.13. デコメ絵文字pop識別子表示可否設定取得	45
14.2.14. デコメピクチャ拡張表示設定	46

14.3. DecoEmojiSpanクラス	47
14.3.1. URI取得	47
14.3.2. 幅取得	48
14.3.3. 高さ取得	49
14.3.4. デコメ絵文字pop識別子取得	50
14.3.5. デコメピクチャ判定	51
14.4. EmojiDrawableクラス	52
14.4.1. 絵文字判定	52
14.4.2. 絵文字判定(コード)	53
14.5. 絵文字一覧アプリDecoEmojiList	54
14.5.1. Intent起動	54
14.5.2. 返値取得	55
14.6. 絵文字マネージャDecoEmojiManager	56
14.6.1. 絵文字マネージャのサービス(aidl)インタフェースのAPI	57
14.6.2. 利用クラス	68
14.6.3. 定数	69
14.6.4. DecoEmojiManager発行インテント	70
14.7. 絵文字マネージャDecoEmojiManager (Content Provider)インタフェース	72
14.7.1. 絵文字パッケージで提供する Content Provider 機能	72
14.7.2. 前提条件	72
14.7.3. 取得モード一覧および取得データ一覧	72
14.7.4. 使用方法	72
14.7.5. 取得データ詳細	73
14.7.6. 取得モード詳細	74
14.7.7. 利用クラス (DecoEmojiContractクラス)	83
14.7.8. 定数	85
14.7.9. 例外	86
15. Data Format	87
15.1. デコメ絵文字フォーマット	87
15.1.1. GIF ファイルフォーマット	87
15.1.2. デコメ絵文字popファイルの判定方法	87
15.1.3. JPEG ファイルフォーマット	87
15.1.4. 付加情報定義フォーマット	88
15.1.5. 付加情報の詳細	89
15.1.6. 付加情報フォーマットDTD(参考)	89
15.1.7. 付加情報文字列の上限	90
15.1.8. 付加情報文字列の文字コード	90
15.1.9. 付加情報文字列の分割	90
15.1.10. 日本語向け品詞一覧	91

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」と連携する機能を IME に組み込むためのガイド(手引書)です。

本ガイドに沿って作成された IME では、キャリア絵文字、デコメ絵文字の一覧選択入力および予測/変換入力と表示を行うことが可能となります。

2. 参照資料

2.1. 参照ドキュメント

3. 略語・記号の定義

IMF	Input Method Framework
IME	Input Method Editor
絵文字	キャリア絵文字とデコメ絵文字の総称。
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。
デコメ絵文字	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「デコメ絵文字」と呼ぶ。
デコメピクチャ	文章中に挿入される自由なサイズの画像。
デコメコンテンツ	デコメ絵文字、デコメピクチャなどの画像ファイルの総称。 ※「デコメ」「デコメ絵文字」「デコメピクチャ」は、株式会社 NTT ドコモの登録商標です。

4. 機能概要

絵文字パッケージは、Android プラットフォームに、携帯電話で普及している絵文字(キャリア絵文字およびデコメ絵文字)の入力と表示の機能を加えるソリューションです。標準的な Android プラットフォームにこのパッケージをインストールすることによって、アプリケーションに共通の絵文字機能を付け加えることができます。アプリケーションは、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、絵文字の入出力を行うことができます。文字入力アプリ(IME)は、絵文字辞書機能を備えることによって、絵文字パッケージと連携して絵文字を入力できるようになります。

5. 構成

絵文字パッケージに IME を加えた絵文字関連のモジュール構成は図 1 絵文字関連モジュール構成(IME 関連のみ抽出) のようになります。

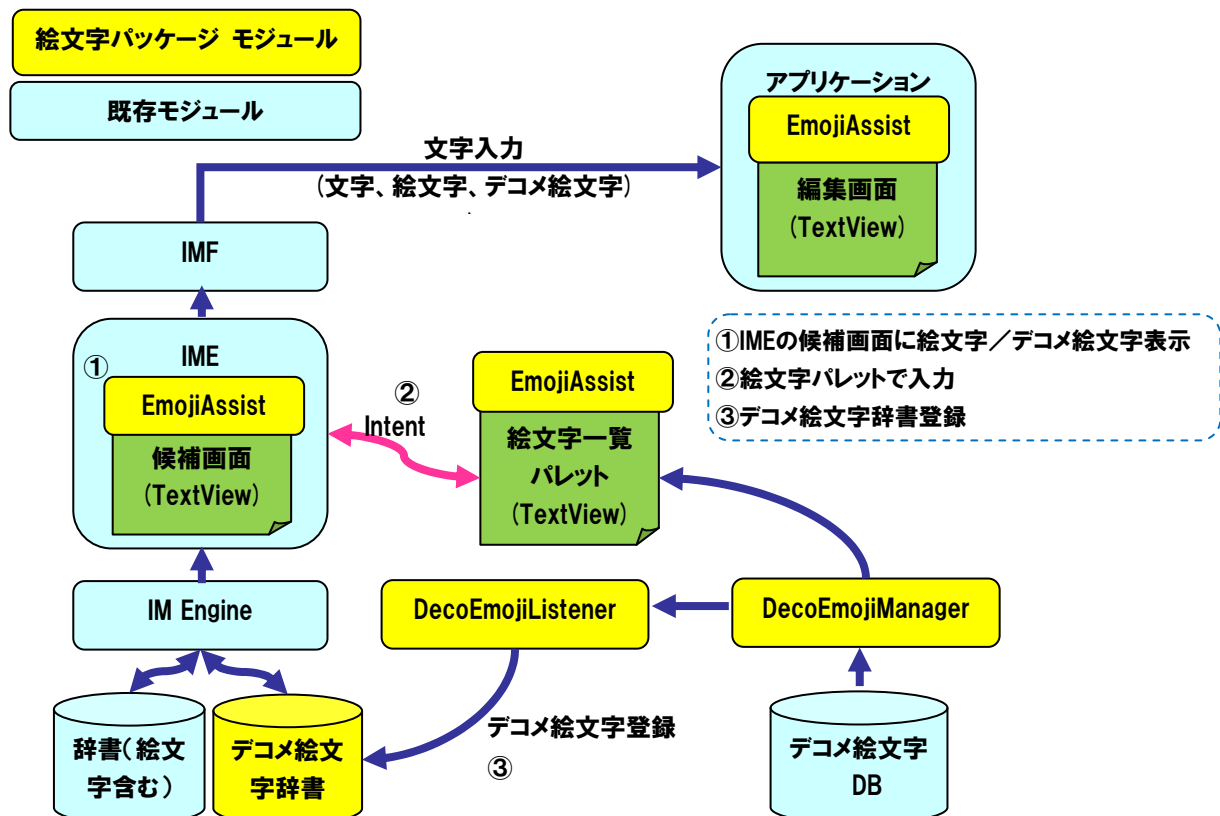


図 1 絵文字関連モジュール構成 (IME 関連のみ抽出)

6. 絵文字パッケージ

Android 絵文字パッケージは、Android プラットフォームに絵文字（キャリア絵文字、デコメ絵文字）・デコメピクチャの入力および表示機能を加えるソリューションです。

6.1. 絵文字パッケージでできること

- ・ 標準的な Android 端末は、絵文字パッケージをインストールすることによって、アプリケーションに共通の絵文字機能を付け加えることができます。
 - (1) Android 端末の FW に絵文字機能を組み込んでいるか否かに関わらず、絵文字パッケージをインストールすることでアプリ間で共通の絵文字を利用することができます。
 - (2) Android 端末にあらかじめインストールしたデコメ絵文字またはダウンロードしたデコメ絵文字を一覧表示し、アプリから入力することを可能にします。
 - (3) デコメ絵文字に読みを付けることができ、IME の予測/変換機能を用いて入力することも可能になります。
- ・ アプリケーションは、絵文字パッケージを利用する機能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、絵文字の入出力を行うことができます。
- ・ ユーザは、通常文字、キャリア絵文字、デコメ絵文字を特別に意識せずにシームレスに利用して、豊かな表現を行うことができます。
- ・ 絵文字パッケージは IM サービス（IME）と連携して、絵文字の入力を可能にします。絵文字パッケージが備えるすべての機能を Android 端末で可能にするには、IME は絵文字の一覧選択機能と、絵文字辞書機能を備えることが必要です。

6.2. 絵文字パッケージの機能レベル

絵文字パッケージには次の 3 つの機能レベルがあります。

機能レベル	IM	アプリ
【レベル A】 キャリア絵文字対応	キャリア絵文字の <u>一覧および予測/変換入力</u>	キャリア絵文字の表示
【レベル B】 デコメ絵文字対応	デコメ絵文字の <u>一覧からの入力</u>	デコメ絵文字の表示
【レベル C】 デコメ絵文字予測/変換入力	デコメ絵文字の <u>予測/変換入力</u>	デコメ絵文字の表示と追加

6.3. 絵文字パッケージのI/F

絵文字パッケージは、アプリケーション、絵文字データを格納する DB 、および IME との間の 3 つのインタフェースを持っています。機能レベルごとに必要となる I/F を下表に示します。

表 1 絵文字パッケージ I/F 一覧

絵文字パッケージ 機能レベル	1.アプリケーション I/F	2. DB I/F	3. IME I/F
(A)キャリア絵文字 入力/表示	・絵文字有効フラグ (allowEmoji)	・絵文字フォント データライブラリ	・絵文字辞書
(B)デコメ絵文字 一覧入力/表示	・デコメ絵文字有効フ ラグ (allowDecoEmoji) ・デコメ絵文字 ID(URI)	・カテゴリ分類 (ディレクトリ)	・一覧表示 ・デコメ絵文字 更新情報
(C) デコメ絵文字 変換入力/表示	(B に同じ)	・デコメ絵文字読み 付加情報	・デコメ絵文字辞書 更新情報

IMEは、絵文字パッケージに対して絵文字を利用するアプリケーションという側面と、絵文字パッケージと連携してアプリケーションの絵文字入力を可能にするサービスという側面を持っています。

7. Application Programming Interface

絵文字を扱うアプリケーションは、**TextView** を用いて、絵文字を含む文字列を入力および表示します。**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。

絵文字は文字コードの問題などから、使えるアプリケーションを限る必要があります。このため、キャリア絵文字を扱えるかどうか、デコメ絵文字を扱えるかどうかのフラグ (**TextView** の属性) を **API** として以下のように定義しています。

(1) **allowEmoji** フラグ

TextView において、キャリア絵文字の入力、表示を可能にする属性 (**Hidden API**)

(2) **allowDecoEmoji** フラグ

TextView において、デコメ絵文字の入力と表示を可能にする属性 (本パッケージで定義した **API**)

(3) **デコメ絵文字 ID**

アプリケーションは、入力/表示において通常文字と絵文字を区別する必要はありません。

しかし、メールへの添付など、デコメ絵文字コンテンツファイルを意識して扱う場合があります。

デコメ絵文字 **ID** はこれをファイルとして扱うためのシステム内の **ID(URI)** です。

7.1. 入力/表示シーケンス

絵文字/デコメ絵文字の入力と表示を行うアプリは以下の流れで実行します。

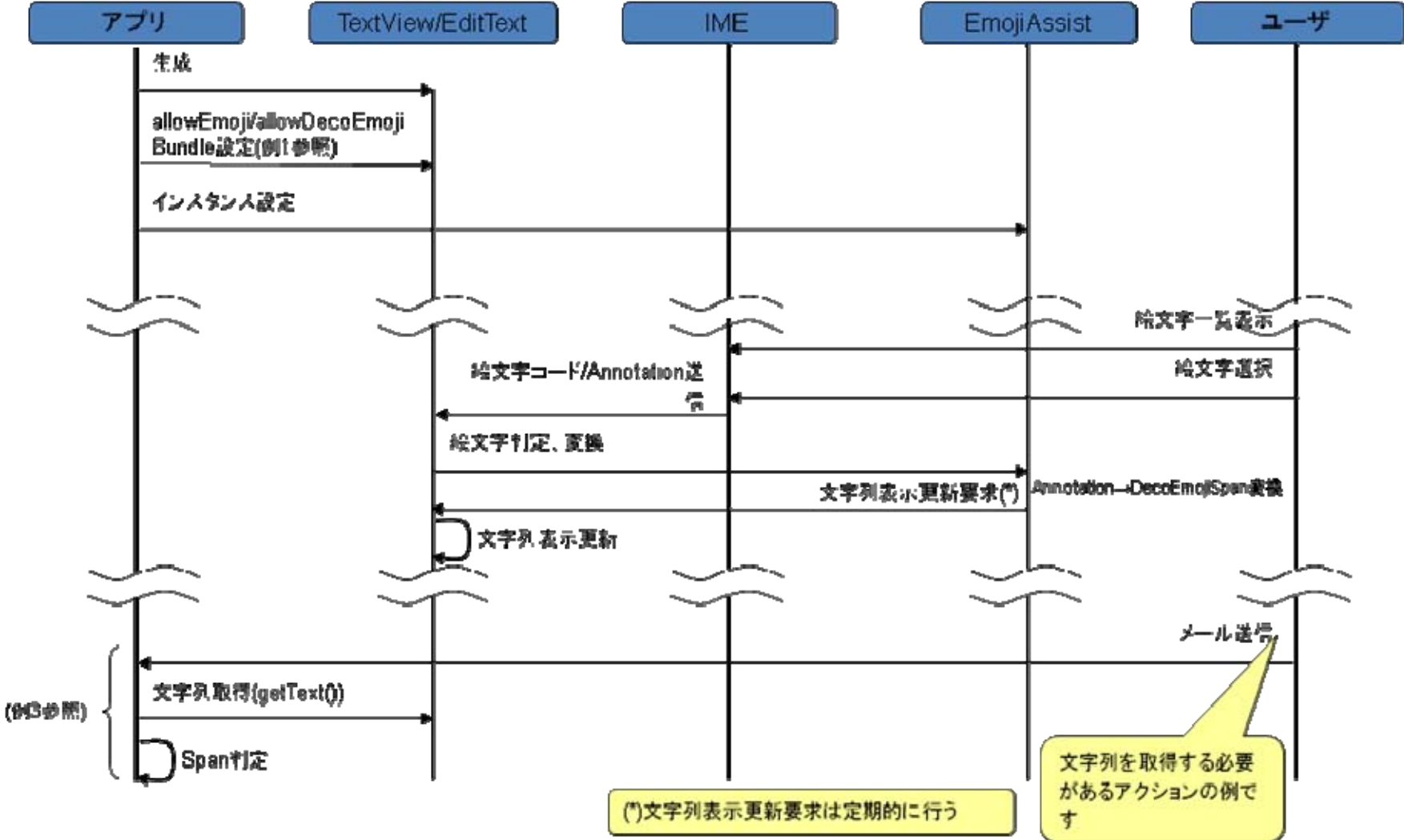


図 2 アプリ実行時のシーケンス

① TextView に絵文字フラグをバンドル

TextView に、絵文字の入力と表示を可能にするフラグを立てます。これは、TextView のインスタンスに Bundle パラメータとして、追加することによって行ないます。

“allowEmoji”

キャリア絵文字の入力と表示を可能にします。

“allowDecoEmoji”

デコメ絵文字の入力と表示を可能にします。

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);
```

② インスタンス生成

絵文字表示を行う TextView のインスタンスを、絵文字制御ライブラリ EmojiAssist に引き渡します。アニメーション絵文字を含む画像表示を自動的に行うため、アニメ動作を開始します。

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();
```

③-1 文字入力

(1) TextView を通じて文字列の入力を行ないます。(getText)

デコメ絵文字は通常文字、キャリア絵文字と同様、文字列の中の 1 文字として入力されます。

文字コードは、別に定めます(*1)。通常表示に用いられない文字が利用されます。このコードは、定義によって変化する可能性があるため、このコードのみによって、デコメ絵文字を識別(デコメ絵文字として認識)してはなりません。

*1:現在の定義は Unicode 0x3013、“=”

(2) 表示について

TextView では EmojiAssist によって、入力が行われるとともに表示は行われているので、アプリケーションが表示のために行うことはありません。

③-2 デコメ絵文字の URI の読み込み

デコメ絵文字の保存や外部出力(たとえば、HTML への変換)のためには、ファイルとしての識別子が

必要となります。

(1) スパン属性の識別

デコメ絵文字は `DecoEmojiSpan` というスパン属性をもちますのでこれで識別します。

(2) URI の読み込み

デコメ絵文字のバンドル情報から、URI を読み込みます。

(3) ファイルの読み込み

ファイルの実体は `ContentProvider` または直接ファイル入出力を用いて読み込みます。

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();
```

8. デコメ絵文字Data Interface

絵文字パッケージをサポートする Android 端末では、デコメ絵文字選択パレットに表示するデコメ絵文字や、変換候補として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。

8.1. デコメ絵文字の定義

絵文字パッケージでは、基本的に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。

- (1) ファイルフォーマット: jpeg、gif
- (2) 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。

8.2. デコメ絵文字の登録と削除

8.2.1. 登録先ディレクトリ

デコメ絵文字は、システムで規定する所定のディレクトリにおかれることによって、絵文字パッケージからデコメ絵文字として認識されます。このデコメ絵文字を登録するディレクトリのことを「登録先ディレクトリ」と呼びます。

デコメ絵文字を追加する場合、アプリケーションから登録先ディレクトリにデコメ絵文字ファイルを格納します。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、IME と連携してデコメ絵文字を変換候補とパレットに追加します。

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ます。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、IME に通知してデコメ絵文字を変換候補とパレットから削除します。

8.2.2. IMEとの関係

絵文字パッケージ対応の IME は、絵文字マネージャ(DecoEmojiManager)から次の二つの通知を受けて、自らが保持するデコメ絵文字の一覧データとデコメ絵文字辞書を更新します。

(1) デコメ絵文字更新通知

- ・ 絵文字マネージャはデコメ絵文字が登録先ディレクトリに追加、削除されたことを Broadcast で通知します。
- ・ IME(その他サービスアプリ)が、デコメ絵文字の一覧データを自アプリ内に保持する場合、そのデータを更新することが必要です。

(2) デコメ絵文字辞書更新通知

- ・ 絵文字マネージャはデコメ絵文字の読み情報が追加、削除されたことを Broadcast で通知します。
- ・ IME は、自らのデコメ絵文字辞書を更新することが必要です。

9. IME Interface

絵文字パッケージは IME と連携して、絵文字の入力を可能にします。IME は、絵文字の一覧選択機能と、絵文字辞書機能を備えることが必要です。

- ・ 一覧選択機能
 - キャリア絵文字一覧選択機能
キャリア絵文字の DB から一覧 Preview を表示し、選択する機能です。携帯電話事業者ごとに定義されます。
 - デコメ絵文字一覧選択機能
デコメ絵文字の DB から一覧 Preview を表示し、選択する機能です。
- ・ 絵文字辞書機能
 - キャリア絵文字辞書
キャリア絵文字をその読みから予測/変換するための辞書です。携帯電話事業者ごとに定義します。
 - デコメ絵文字辞書
デコメ絵文字をその読みから予測/変換するための辞書です。所定の領域に格納されたコンテンツから、自動的に生成されます。(本パッケージで I/F を定義)

9.1. キャリア絵文字入力I/F（A機能）

A-1. 一覧表示（パレット）入力

- (1) 一覧表示アプリを用意します。
 - IME からインテントで呼び出します。
 - 選ばれたキャリア絵文字のコードを返却します。
 - 選ばれた履歴を一定量保持し、先頭に表示します。
- (2) 表示定義ファイル
 - 表示順序定義
一覧に表示するカテゴリと、表示順序は、一覧表示アプリのリソースとして、XML ファイルに定義します。

A-2. 予測/変換入力

- (1) キャリア絵文字を含む予測/変換辞書を用意します。
キャリア絵文字は文字コードを持ちますので、辞書を用意すれば入力が可能です。
(IME 毎にフォーマットは異なります。)
- (2) IME 内の候補画面に絵文字を出力させるために、EmojiAssist を利用します。

【インタフェース】

- ・ 絵文字表示ライブラリ (EmojiAssist)
- ・ 絵文字一覧アプリ (DecoEmojiList)

9.2. デコメ絵文字一覧入力/I/F（B機能）

(1) 一覧表示（パレット）入力

- ・一覧表示アプリを用意します。
 - IME からインテントで呼び出します。
 - 選ばれたデコメ絵文字の URI を返却します。
 - 選ばれた履歴を一定量保持し、先頭に表示します。
- ・IME は、パッケージが提供するライブラリにより、URI をデコメ絵文字属性付き文字に変換して、アプリケーションに返します。
- ・カテゴリ名定義に従い、デコメ絵文字は分類して表示されます。

(2) 表示定義ファイル

- ・カテゴリ名定義
 - カテゴリ区分はサブディレクトリによって区別します。このディレクトリ名と表示する区分タイトルの対応を絵文字パッケージの設定として定義します。

【インタフェース】

- ・絵文字表示ライブラリ（EmojiAssist、getDecoEmojiText）
- ・絵文字一覧アプリ（DecoEmojiList）



図 3 絵文字一覧の UI

9.3. デコメ絵文字予測/変換I/F（C機能）

(1) 予測/変換入力

- ・デコメ絵文字用予測/変換辞書を用意します。
 - デコメ絵文字は文字コードを持ちませんので、特別な辞書を用意する必要があります。（IME 毎にフォーマットは異なります。）
 - デコメ絵文字辞書には、次の項目を含むことを想定します。
 - ・デコメ絵文字 ID（URI）
 - ・読み
 - ・品詞（オプション）
- ・IME 内の候補画面に絵文字を出力させるために、EmojiAssist を利用します。

(2) デコメ絵文字辞書の更新

- ・IME に対し、デコメ絵文字が追加されるとパッケージは DecoEmojiListener を通じて次の情報を提供します。
 - デコメ絵文字 ID（URI）
 - 読み
 - 品詞
- ・IME はこの情報を受けて、デコメ絵文字辞書に情報を追加します。
- ・デコメ絵文字が削除されたとき、ID の削除を通知します。
 - IME はデコメ絵文字辞書からデコメ絵文字を削除します。
- ・デコメ絵文字が変更されたとき、削除と追加を通知します。
 - IME は、デコメ絵文字辞書の削除と追加を行いません。
- ・IME は Android 端末の稼働中に切り換えられたり、新たにインストールされたりすることがあるため、絵文字パッケージのデコメ絵文字データベースとデコメ絵文字辞書が不一致になることがあります。このため、IME と絵文字パッケージとは適切なタイミングで同期を取る必要があります。

【インタフェース】

- ・絵文字表示ライブラリ（EmojiAssist、getDecoEmojiText）
- ・絵文字マネージャ（DecoEmojiManager）

[説明]

A,B 機能において、絵文字一覧を、絵文字一覧アプリを使わず、IME 内に組み込むことも可能です。絵文字一覧アプリは、Reference Implementation として、ソースコードも開示します。

表 2 IME の絵文字対応とインタフェース

IMEに実装を期待する機能			IME固有機能	絵文字パッケージの用意するインタフェース			
				(1) 絵文字表示Lib (EmojiAssist)	(2) 絵文字一覧アプリ (DecoEmojiList)	(3) デコメ絵文字文字変換 (getDecoEmojiText)	(4) デコメ絵文字辞書更新通知 (DecoEmojiManager I/F)
凡例	●:絵文字パッケージインタフェースを利用する ◎:独自に機能を用意する		IME側で機能を実装	候補画面 (TextView)に絵文字を表示する	絵文字一覧を表示し、文字を選択入力する	デコメ絵文字属性付き文字を生成する	端末に登録されたデコメ絵文字を検出し、デコメ絵文字読み情報の追加、削除をIMEに通知する
A-1 絵文字一覧入力	絵文字一覧を表示し、絵文字を選択入力する	1.絵文字一覧を呼ぶ【注1】			●		
A-2 絵文字予測/変換入力	通常文字と同様に絵文字(文字コードあり)の予測/変換を行う	1.絵文字を含む辞書を用意する	◎				
		2.候補画面(TextView)に絵文字を表示する		●			
B デコメ絵文字一覧入力	絵文字一覧を表示し、デコメ絵文字を選択入力する	1.絵文字一覧を呼ぶ【注1】			●		
		2.デコメ絵文字属性付き文字コードを生成し、文字列に含めてアプリに出力する。				●	
C デコメ絵文字予測/変換入力	通常文字と同様にデコメ絵文字の予測/変換を行う	1.IMEの固有の機能を用いて、デコメ絵文字予測/変換辞書(入れ物)を用意する	◎				
		2.デコメ絵文字属性付き文字コードを生成し、文字列に含めてアプリに出力する。				●	●
		3.候補画面(TextView)にデコメ絵文字を表示する		●			
		4.デコメ絵文字更新通知を受けて、デコメ絵文字予測/変換辞書(内容)を更新する					●

【注1】IME固有の絵文字一覧を実装することも可能。

10. 【実装手順】IMEからの絵文字パッケージ利用手順

キャリア絵文字、デコメ絵文字が入力可能なIME開発において、一般的なアプリで必要となる「7 Application Programming Interface」に加えて、IMEとして対応が必要となる内容を記載しています。

10.1. IMEのキャリア絵文字一覧入力機能追加

キャリア絵文字一覧は、IMEに機能追加して実現するか、絵文字一覧入力アプリ(DecoEmojiList)をIME側から起動する動線を追加します。

絵文字一覧アプリはIntentにより起動されるアプリとして実装されています。(「10.3.1」項参照)

10.2. IMEのキャリア絵文字予測/変換入力対応

IMEの予測/変換入力機能に、キャリア絵文字も候補に含めるためには、IMEにて下記の対応を行う必要があります。

- (a) キャリア絵文字辞書を追加する。
- (b) 候補表示において、絵文字パッケージのキャリア絵文字表示のための対応を行う。(「10.5」項参照)

10.3. IMEのデコメ絵文字一覧起動・入力の追加

10.3.1. デコメ絵文字一覧の起動

デコメ絵文字一覧は、IMEに機能追加して実現するか、絵文字一覧入力アプリ(DecoEmojiList)をIME側から起動する動線を追加する必要があります¹。絵文字一覧アプリはIntentにより起動されるアプリとして実装されています。

サンプルコード:

```
Intent intent = new Intent();
// クラス名にデコメ絵文字一覧アプリを指定
intent.setClassName("jp.co.omronsoft.android.decoemojilist_<career>",
    "jp.co.omronsoft.android.decoemojilist_<career>.DecoEmojiList");(*1)
// 呼び出す Activity 以外の Activity をクリアして起動させる設定を付与
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
// デコメ絵文字一覧アプリ上で表示する絵文字タグ切替値の設定
intent.putExtra("switch_emoji_flg", 1); // 0:both、1:deco、2:carrier
// 縦画面時のデコメ絵文字一覧アプリを表示する高さの領域を設定
intent.putExtra("current_ime_height_port", heightP);
// 横画面時のデコメ絵文字一覧アプリを表示する高さの領域を設定
intent.putExtra("current_ime_height_land", heightL);
// デコメ絵文字取得方法を設定
```

¹ 同様の機能を、IMEではなくメーラ等のアプリで実装することも考えられます。その場合、絵文字一覧入力アプリからのレスポンスを当該アプリで受け付け、受け取ったキャリア絵文字/デコメ絵文字を編集画面に反映するなどの処理を追加する必要があります。

```
intent.putExtra("current_input_type", 0); // 0:複数候補選択後にアプリに返す、1:候補を一つ選択した時点でアプリに返す
// デコメ絵文字一覧アプリ起動
startActivityForResult(intent, 0); // request code = 0
```

10.3.2. 入力されたデコメ絵文字の伝送

絵文字一覧入力アプリは、入力されたデコメ絵文字の列を、その起動元(通常 IME)に対し次の形式の Bundle 情報として返します。

型	内容
List	android.net.Uri の List

IME でデコメ絵文字を扱うには、IME で Bundle 内の List から URI(デコメ絵文字を示す)を取り出し、デコメ絵文字属性付き文字コードに変換して EditText(アプリケーションの入力フィールド)に伝達するように修正する必要があります。

サンプルコード:

```
EmojiAssist ea = EmojiAssist.getInstance();
// URI からデコメ絵文字属性付き文字を生成する
CharSequence decoemoji = ea.getDecoEmojiText(uriString, ctx);
// uriString はデコメ絵文字の uri、ctx はアプリケーションのコンテキスト
```

10.4. IMEのデコメ絵文字予測/変換入力対応

IME の予測/変換入力機能に、デコメ絵文字も候補に含めるためには、下記の対応を行う必要があります。

- 予測/変換に用いるデコメ絵文字辞書を追加する。
- 候補表示において、絵文字パッケージのデコメ絵文字表示のための対応を行う。
- デコメ絵文字の追加/削除の通知を DecoEmojiManager から受け取る機能を追加する。

絵文字パッケージでは、IME でデコメ絵文字の予測/変換を可能にするためのインタフェースを用意しています。

- 絵文字表示ライブラリ(EmojiAssist)
 - 候補画面(TextView)表示、デコメ絵文字生成インタフェース。次項参照。
- 絵文字マネージャ (DecoEmojiManager)
 - デコメ絵文字読み情報(辞書更新)通知インタフェース。
 - 「11 【実装手順】デコメ絵文字読み情報更新手順」を参照。

10.5. 候補画面の表示

10.5.1. TextViewをEmojiAssistに組み込み、絵文字表示を可能にする

絵文字を表示する候補画面(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) 設定された `TextView` にキャリア絵文字、デコメ絵文字があったとき、これを画像に変えて表示します。
- (2) 設定された `TextView` を定期的に更新(`invalidate`)し、アニメーション絵文字をアニメ動作させます。
- (3) 設定された `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 を TextView に割り当てる
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();
```

10.5.2. 候補画面にデコメ絵文字を表示する

候補画面にデコメ絵文字を表示するためには、デコメ絵文字をデコメ絵文字属性付き文字に変換して、`TextView` に書き込む必要があります。

サンプルコード:

```
EmojiAssist ea = EmojiAssist.getInstance();
// URI からデコメ絵文字属性付き文字を生成する
CharSequence decoemoji = ea.getDecoEmojiText(uriString, ctx);
// uriString はデコ絵文字の uri、ctx はアプリケーションのコンテキスト
```

10.6. 入力フィールドの絵文字属性判定

「7 Application Programming Interface、」にあるように、アプリの入力フィールド(`EditText`)には `Bundle` パラメータとして下記の絵文字入力属性が設定されています。

- (1) キャリア絵文字入力許可フラグ `key:"allowEmoji"`
- (2) デコメ絵文字入力許可フラグ `key:"allowDecoEmoji"`

IME は、入力フィールドの属性に従って、対応する入力候補を表示しなくてはなりません。

10.6.1. キャリア絵文字入力許可フラグ (allowEmoji)

IME はキャリア絵文字入力許可フラグ(`allowEmoji`)がオンのとき、キャリア絵文字の入力を可能にします。`allowEmoji` フラグが `Off (false)` の時、または設定されていない時は、キャリア絵文字の入力候補を表示しません。

キャリア絵文字の入力候補は、次のように表示します。

(1) 絵文字一覧(パレット)

カテゴリと、表示順序を設定した絵文字コードの定義表を元に、一覧表を表示します。

キャリア絵文字コードの設定は、事業者ごとに異なります。

一覧表示アプリ(DecoEmojiList)を利用する場合は、Intent でこれを起動し、ユーザの入力結果を受け取ります。

(2) 予測変換入力

キャリア絵文字コードを含む、辞書を用意し、予測・変換を行います。

辞書の実装方法は、IME によって異なりますが、allowEmoji が Off または指定されない場合は、予測変換に利用されないようにしなければなりません。

10.6.2. デコメ絵文字入力許可フラグ (allowDecoEmoji)

IMEはデコメ絵文字入力許可フラグ(allowDecoEmoji)がOnのとき、デコメ絵文字の入力を可能にします。allowDecoEmoji フラグが Off (false) の時、または設定されていない時は、デコメ絵文字の入力候補を表示しません。

デコメ絵文字の入力候補は、次のように表示します。

(1) 絵文字一覧(パレット)

カテゴリと、表示順序に従ったデコメ絵文字 ID(URI) の一覧データを絵文字マネージャ(DecoEmojiManager)から入力し、一覧表を表示します。

カテゴリと一覧データの入力方法は、ContentProvider インタフェースを用いるか、aidl インタフェースを用いるかによって異なります。

一覧表示アプリ(DecoEmojiList)を利用する場合は、Intent でこれを起動し、ユーザの入力結果を受け取ります。

(2) 予測変換入力

デコメ絵文字辞書を用意し、予測・変換を行います。

方法は、IME によって異なりますが、allowDecoEmoji が Off または指定されない場合は、予測変換に利用されないようにしなければなりません。

10.6.3. 一覧表示の順序制御

一覧表示では、履歴表示が求められることがあります。IME をはじめ、いくつもの一覧表示アプリが存在するとき、それぞれのアプリ内で履歴が管理されるため、ユーザにとっての操作履歴としては不統一となり、UX 上の不都合となります。

これを避けるため、絵文字マネージャにデコメ絵文字の選択操作を通知し、履歴を集めることにより、統一された履歴管理を実現します。以下の API を利用します。

(1) 利用履歴更新通知 `aidl_changeHistory(uri)`

選択、採用したデコメ絵文字の ID(URI)を絵文字マネージャに通知します。絵文字マネージャは利用履歴を共通の DB で管理します。

(2) カテゴリ内利用順更新

(1)の通知を行うことで、カテゴリ内の表示順序も、利用履歴の最新順に更新することができます。この表示順序は絵文字プロバイダにソートを指示することによって取得可能になります。なお、カテゴリ内利用順序には、ダウンロードされ、新しく追加されたデコメ絵文字も新しいものからこの順序に加えられ、ユーザ

に気づきの機会を与えることができますようになります。(DL 絵文字の順序制御は絵文字マネージャ内で自動的に行われます。)

(3) 利用履歴のリセット `aidl_resetHistory(target)`

利用履歴の記録をリセットすることができます。カテゴリ内の表示順序もリセットすることが可能です。

(4) 利用履歴リスト取得 `aidl_getHistoryList(max, emojiType, uriList)`

デコメ絵文字の利用履歴の最新のものから順に指定個数を取り出すことができます。

詳細は「14.6.1 絵文字マネージャのサービス(aidl)インタフェースのAPI」を参照ください。

一覧表示アプリケーションは、表示パレットの再表示など、適切なタイミングで、絵文字プロバイダから一覧情報を取り直し、共通の表示順序を反映することが必要です。

絵文字パッケージの提供する絵文字一覧アプリ(DecoEmojiList)は、その中で順序制御の処理を行っていますので、呼び出し側のアプリ(IME)はこの処理を行う必要はありません。

11. 【実装手順】デコメ絵文字読み情報更新手順

IME は、絵文字マネージャ (DecoEmojiManager) からの「デコメ絵文字辞書更新通知」を受けて、絵文字辞書を更新 (追加、削除) します。

IME と絵文字マネージャのインタフェースには、aidl インタフェースと、(絵文字パッケージ V1.2 から追加された) ContentProvider インタフェースとがあります。ContentProvider インタフェースは、アプリや IME からの要求に対する柔軟性に優れているため、今後のバージョンでは、このインタフェースを中心に機能を強化していきます。

従来の IME 実装方法との互換性のため、aidl インタフェースによる実装手順の説明を「11.2 デコメ絵文字読み情報更新手順 (aidl インタフェース)」に記載します。

11.1. デコメ絵文字読み情報更新手順 (ContentProvider インタフェース)

【対応バージョン: 絵文字パッケージ V1.2 以上】

絵文字マネージャ ContentProvider 機能 (以下、絵文字プロバイダと呼びます) を用いて、絵文字マネージャの持つデコメ絵文字 DB にアクセスすることができます。絵文字プロバイダへのアクセスは、参照 (Query) だけが可能です。絵文字プロバイダを通じて DB にデータを追加したり、変更することはできません。

デコメ情報読み情報更新は以下のように行います。

- (1) デコメ絵文字の更新 (追加・削除) があったとき、
 - (ア) 絵文字マネージャから「デコメ絵文字情報更新通知」が Broadcast されます。
 - (イ) IME の予測・変換機能としては何もする必要はありません。一覧表示アプリ、または IME 内で行っている一覧表示機能では、これによって一覧表示データの更新が必要になります。これについては次章「12 【実装手順】デコメ絵文字の一覧表示の更新」を参照ください。
- (2) 更新されたデコメ絵文字に読み情報があったとき、
 - (ア) 絵文字マネージャから「デコメ絵文字辞書更新通知」が Broadcast されます。
 - (イ) IME は、読み情報の更新 (削除・追加) 情報を元に、デコメ絵文字辞書を更新します。辞書の更新は、IME の予測・変換表示とは非同期に行われ、辞書の更新が終わったときから、予測・変換に用いられます。
- (3) システムの初期化、デコメ絵文字格納場所の大幅な変更 (SD カードの差し替え) 等によって、絵文字パッケージの DB も初期化が必要な場合があります。絵文字パッケージの初期化が行われた場合、
 - (ア) 絵文字マネージャから、(1) と (2) の更新通知に乗せて、「初期化」が通知されます。
 - (イ) (1) の通知に関しては、一覧表の保有データをリセットし、絵文字プロバイダからすべての対象絵文字データを取り直すことが必要です。
 - (ウ) (2) の通知に関しては、IME がデコメ絵文字辞書をリセットし、絵文字プロバイダから、すべてのデコメ絵文字読み情報を取り直すことが必要です。
- (4) IME アプリの初期時など、IME から初期化が必要となった場合、
 - (ア) IME は、絵文字プロバイダからすべての絵文字読み情報を取り込み、デコメ絵文字辞書を更新します。
 - (イ) IME の一覧表示機能、一覧表示アプリは、絵文字プロバイダから、対象となる絵文字データを取り直して、表示を行います。

絵文字プロバイダのインタフェースについては、「14.7 絵文字マネージャ DecoEmojiManager (Content Provider) インタフェース」を参照ください。

11.2. デコメ絵文字読み情報更新手順（aidlインタフェース）

11.2.1. デコメ絵文字辞書情報更新の基本手順

辞書情報更新の関係モジュールは以下の通りです。

(1) 絵文字マネージャ（DecoEmojiManager）

デコメ絵文字の登録、変更、削除を検知すると、IME に対して、更新情報を載せて Intent で通知を行います。絵文字マネージャではあらかじめ IME が特定できないため、ブロードキャストを行います。

(2) IME / DecoEmojiListener

絵文字マネージャからの Intent を受け取って、デコメ絵文字の辞書を更新します。Intent は、辞書更新情報をブロック化して分割して送信されるため、辞書の更新後、残りの更新がないか絵文字マネージャに問い合わせ、次の更新情報の通知を待ちます。絵文字マネージャへの問い合わせ/要求は、絵文字マネージャサービスへの Bind（aidl）インタフェースによるプロセス間通信で行います。

（実装例）

DecoEmojiListener は IME 側で Intent のレシーバの役割を担います。

1. DecoEmojiListener（Listener と省略）は絵文字マネージャからの Intent のレシーバ機能を持つ。IME が起動しているか否かに関わらず、絵文字マネージャからの更新通知（Intent）を受けられることが望ましい。絵文字マネージャの絵文字検出と通知は IME とは非同期に行われるため、IME が起動していないときの通知を逃すことで辞書に不一致が生じるためである。（IME 起動時に、起動していない間の通知を問い合わせる実装も可能。）
2. 一方で、Listener がデコメ絵文字辞書を更新するとき、IME とは、デコメ絵文字辞書を共有することになるため、両方で競合管理を行うことが必要なことがある。
3. IME と Listener を、同一プロセス（別スレッド）とするか別プロセスとするかはインプリメントに依存する。

表 3 モジュール間のメッセージ

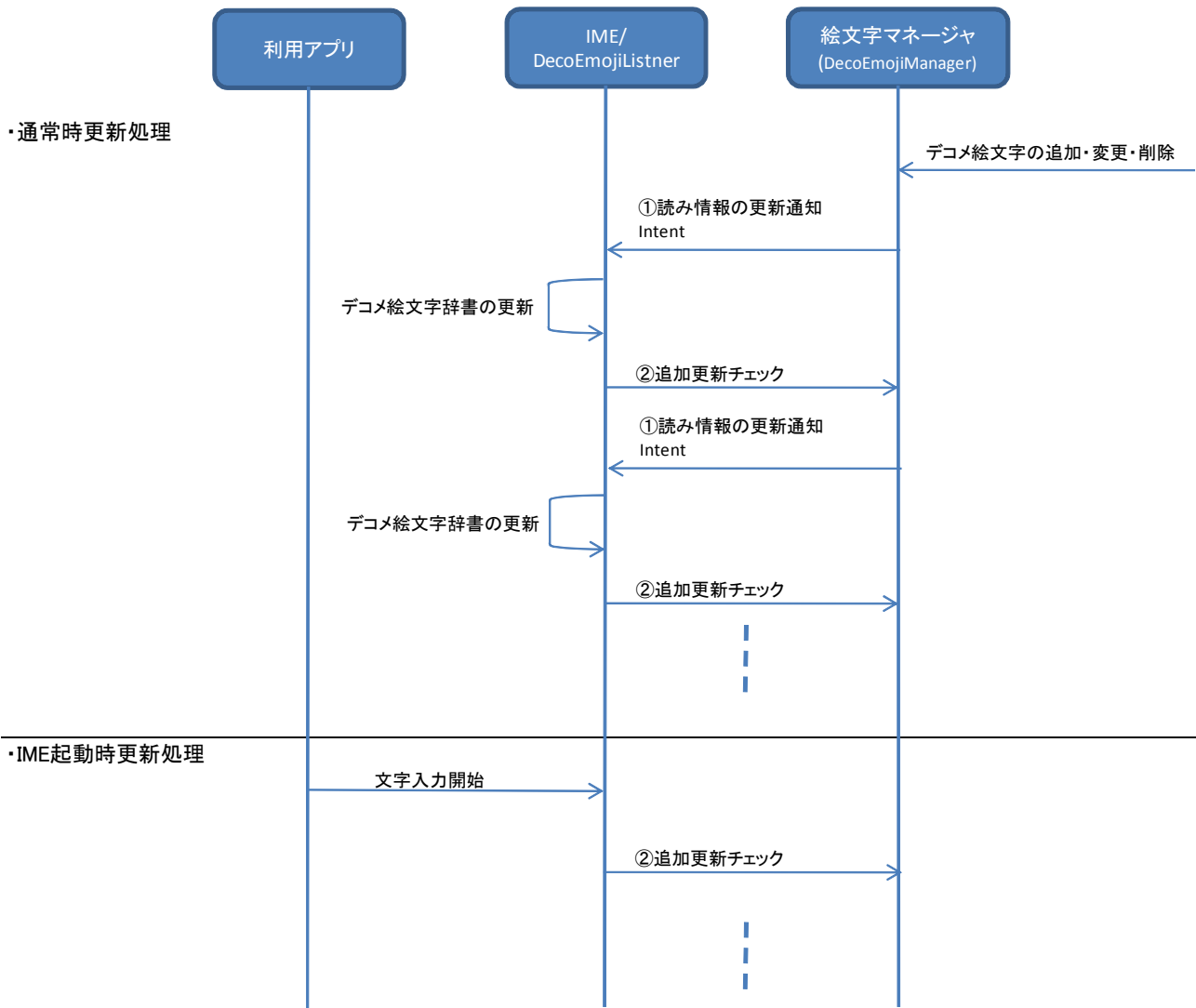
メッセージ	通知	経路	IME の処理	備考
①デコメ絵文字辞書更新通知	M→I	Intent	デコメ絵文字辞書の更新（追加と削除） IME 不特定時のブロードキャストと ②③に応える IME 向けユニキャストがある が処理は同じ	最大 100 件の通知
②追加更新チェック	I→M	Bind	更新した最終 ID を通知し、追加の更新通知を要求する	
③デコメ絵文字辞書初期化要求	I→M	Bind	全読み情報の再送を求める	初期処理
④デコメ絵文字辞書初期化通知	M→I	Intent	デコメ絵文字辞書のリセットを行う	初期処理

M: 絵文字マネージャ、 I: IME

11.2.2. シーケンス図

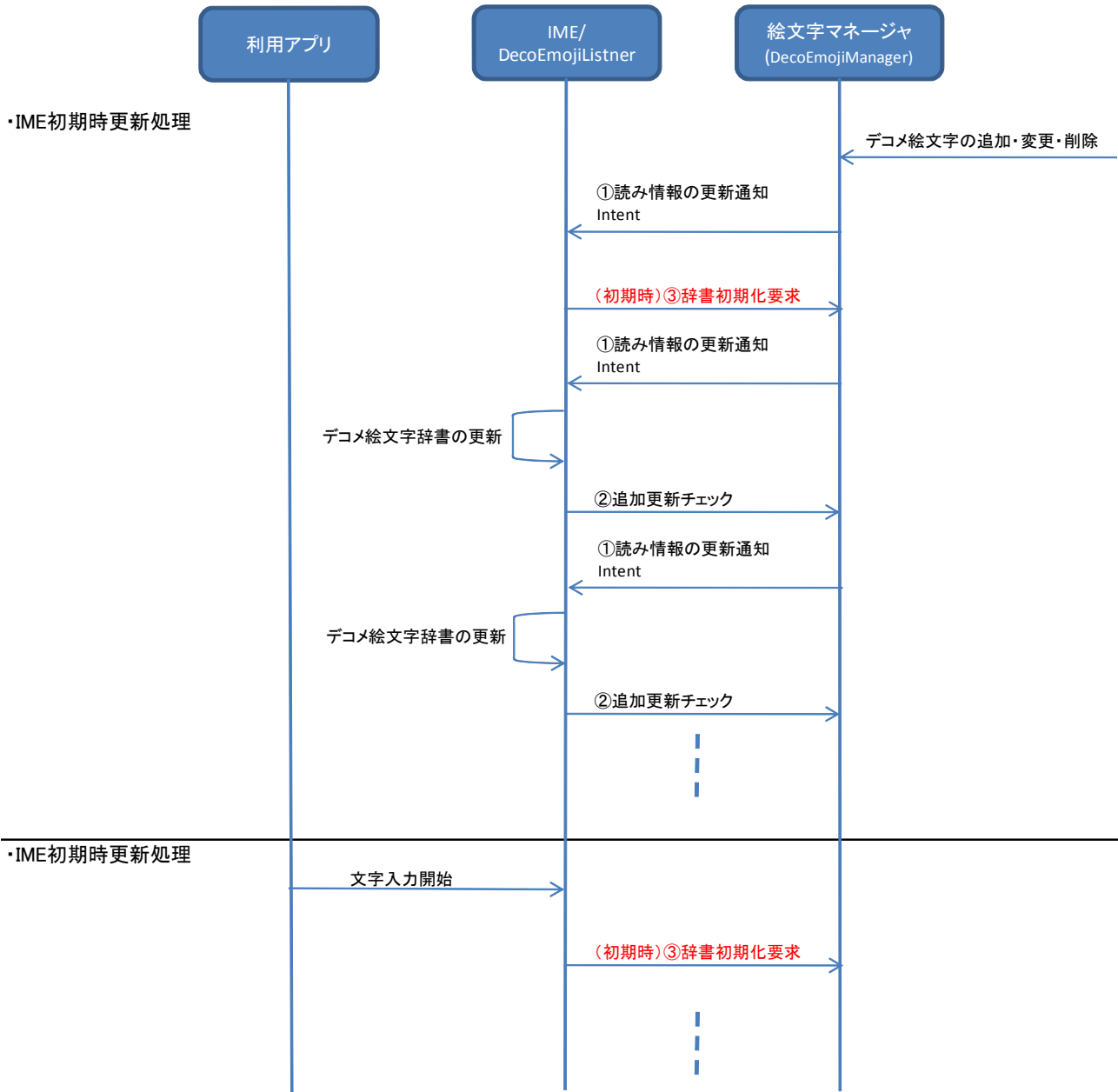
想定している実装シーケンスを示します。

(1) 通常更新処理



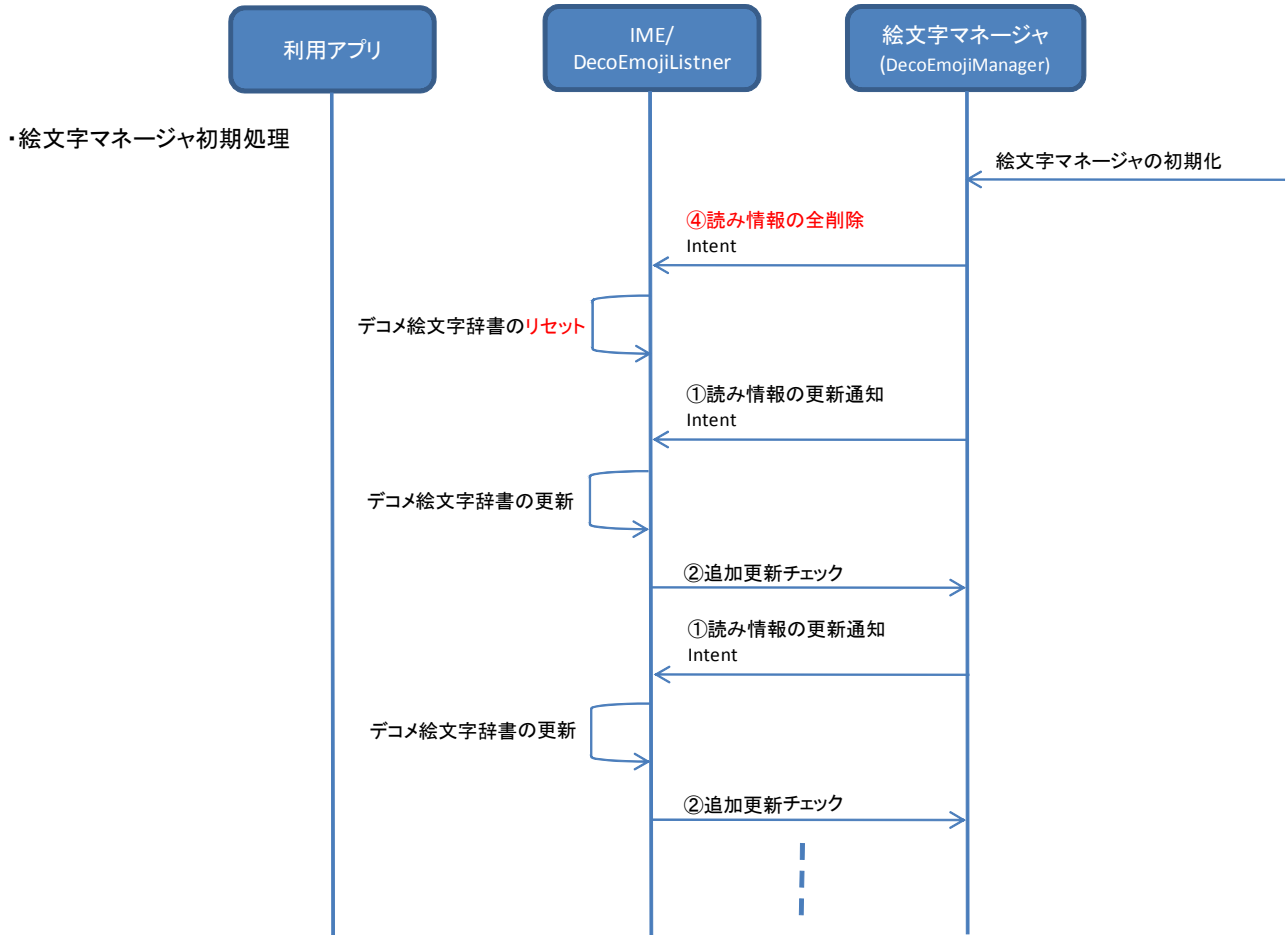
(2) IME 初期処理

IME が初めてデコメ絵文字辞書を作成するとき、絵文字マネージャに全更新データを求めます。



(3) 絵文字マネージャ初期処理

絵文字マネージャ自身が初期化されたとき、IME に、デコメ絵文字辞書の再作成を求めます。



12. 【実装手順】デコメ絵文字の一覧表示の更新

【対応バージョン:絵文字パッケージ V1.1 以降】

絵文字マネージャがデコメ絵文字の更新を検出した時に、この通知を **Broadcast** する機能が提供されました。

このことにより、デコメ絵文字の一覧表を表示したり、デコメ絵文字の付加情報を利用する IME などのサービスアプリケーションが、デコメ絵文字の更新の通知を受けて、一覧表などの情報を更新することが可能となりました。

12.1. 一覧表示アプリケーションの表示更新

絵文字の一覧表示など、自アプリ内にデコメ絵文字の情報を取り込んで処理をしているアプリの場合、動的に変化しているデコメ絵文字の更新情報を取り込む手段がなく、表示内容を最新の状態に保つことが難しいという課題がありました。

絵文字マネージャ(DecoEmojiManager)は、登録先ディレクトリ内のデコメ絵文字の更新(追加/変更/削除)を検出すると、Broadcastでこれを通知します。(14.6.4.1「デコメ絵文字情報更新通知」)

また、Bundle データとして、更新を検出したデコメ絵文字の URI、更新区分(追加、削除)、(更新データをDBに登録した)Timestampを通知します。(デコメ絵文字の「変更」は、「削除」と「追加」として通知されます。)

一覧表示アプリケーションは、この Broadcast を受信してデコメ絵文字情報を取り込み、自らの表示内容を更新することができます。

デコメ絵文字の一覧表の更新方法にはいくつかの方法があります。

- (1) 絵文字一覧アプリ(DecoEmojiList)を用いている場合、絵文字一覧アプリを再起動し、一覧情報を取り直す。
- (2) Bind(aidl)インタフェースを用いて、絵文字マネージャから一覧情報を取り直す。
詳細は「14.6.1 絵文字マネージャのサービス(aidl)インタフェースのAPI」を参照ください。
- (3) 更新通知の内容から、当該デコメ絵文字の情報を更新する。必要な場合はデコメ絵文字の ID(URI)を元に、絵文字プロバイダにアクセスし、情報を補足します。

なお、更新通知に乗せて、絵文字マネージャ(DB)の初期化が通知されることがあります。この場合は、すべてのデータをリセットし、必要な一覧情報を、絵文字マネージャから取り直すことを行う必要があります。

13. 【実装手順】IMEに関する付加情報

13.1. Android Framework全画面モード時の既存不具合対応

Android Framework には IME の全画面モード時の既存不具合が存在します。(V2.3 現在)

- ・ 全画面モードに適用される `ExtractEditText` 内で文字の編集集中に、`Span` 属性が削除されることがある。

その不具合対応として、IME 側に `ExtractEditText` の差し替え処理およびそこで必要となるリソースの追加を実装する必要があります。

絵文字パッケージでは、`EmojiAssist` 内に `ExtractEditText` の代替え処理を用意し、`iWnn IME` の V2 をベースとした追加リソースおよび `ExtractEditText` の差し替え処理の修正差分を用意しています。カスタム対応された `iWnnIME` に適用する場合は、その差分をカスタム版に反映する必要があります。

他の IME にも、参考情報としてこれを提供することは可能です。

追加リソース:

- ・`res¥drawable¥emoji_extract_edit_text.xml`
- ・`res¥drawable-hdpi¥emoji_keyboard_textfield_selected.9.png`
- ・`res¥drawable-hdpi¥emoji_textfield_disabled.9.png`
- ・`res¥layout¥emoji_extract_view.xml`

サンプルコード:

```
@Override public View onCreateExtractTextView() {
    View v = super.onCreateExtractTextView();
    extractEditText =
    (ExtractEditText)EmojiAssist.replaceExtractEditText((ViewGroup)v,
    R.layout.emoji_extract_view);
    return v;
}
```

13.2. 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。

13.2.1. 対応策 (アクティビティアプリケーション向け)

`EmojiAssist` を使用するアクティビティアプリケーション(一覧表示アプリなど)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。

startAnimation()、stopAnimation() の使用例につきましては、絵文字パッケージ内のリファレンスソースコード「DecoEmojiList/src/DecoEmojiList.zip」をご参照下さい。

13.2.2. 対応策 (サービスアプリケーション向け (IME を含む))

EmojiAssist 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();
            }
        }
    }
};

```

14. 【API】インタフェース仕様

●:使用可能 ▲:使用可能(非推奨) ×:使用不可能

クラス名	API名称	API概要	対応バージョン			
			v1.0	v1.1	v1.2	以降(予定)
EmojiAssist	getInstance()	EmojiAssistのインスタンスを取得する。	●	●	●	●
	addView()	TextViewを登録する。	●	●	●	●
	removeView()	登録したTextViewを削除する。	●	●	●	●
	clearView()	登録したTextViewを全て削除する。	●	●	●	●
	startAnimation()	絵文字のアニメーションを開始する。	●	●	●	●
	stopAnimation()	絵文字のアニメーションを停止する。	●	●	●	●
	getDecoEmojiText(String uri)	デコメ絵文字属性 (DecoEmojiSpan、もしくはAnnotation)を設定した文字を取得する。	●	●	●	●
	getDecoEmojiText(String uri, Context context)	デコメ絵文字属性 (DecoEmojiSpan、もしくはAnnotation)を設定した文字を取得する。	×	×	●	●
	getEmojiImage()	キャリア絵文字の画像データ(Binary)を取得する。	●	●	●	●
	replaceExtractEditText()	ExtractEditTextを差し替える。	●	●	●	●
	checkTextData()	文字のチェックを行う。	×	×	●	●
	setDisplayPop()	デコメ絵文字pop識別子の表示可否を設定する。	×	×	●	●
	getDisplayPop()	デコメ絵文字pop識別子の表示可否を取得する。	×	×	●	●
	setPictureScale()	デコメピクチャに対し、デコメ絵文字と同様の拡張表示を行う設定を行う。	×	×	●	●
DecoEmojiSpan	getURI()	デコメ絵文字の画像ファイルを指すURIを返却する。	●	●	●	●
	getWidth()	デコメ絵文字の画像ファイルの幅を取得する。	×	×	●	●
	getHeight()	デコメ絵文字の画像ファイルの高さを取得する。	×	×	●	●
	getPopFlag()	デコメ絵文字の画像ファイルのデコメ絵文字pop識別フラグを取得する。	×	×	●	●
	isTypePicture()	絵文字種別がデコメピクチャか否かを判定する。	×	×	●	●
EmojiDrawable	isEmoji(int code)	引数の文字が絵文字か否かを判定する。	●	●	●	●
	isEmoji(CharSequence text)	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。	●	●	●	●

●:使用可能 ▲:使用可能(非推奨) ×:使用不可能

クラス名	API名称	API概要	対応バージョン			
			v1.0	v1.1	v1.2	以降(予定)
IDecoEmojiManager	aidl_getDecoUri()	デコメ絵文字のURIを取得する。	●	●	▲	×
	aidl_getCategoryList()	デコメ絵文字のカテゴリ情報一覧を返却する。	●	●	▲	×
	aidl_getCategoryList_ex()	指定された対象絵文字タイプ (“emojiType”) のデコメ絵文字のカテゴリ情報一覧を返却する。	×	×	▲	×
	aidl_getDecoUriList()	引数のカテゴリIDに紐づくデコメ絵文字のURI一覧を返却する。	●	●	▲	×
	aidl_getDecoUriList_ex()	引数の対象絵文字タイプ (“emojiType”)、カテゴリIDに紐づくデコメ絵文字のURI一覧を返却する。	×	×	▲	×
	aidl_checkDecoEmoji()	引数のデコメ絵文字IDより後に追加更新されたデコメ絵文字を抽出し、追加分のデコメ絵文字辞書更新情報をブロードキャスト送信する。	●	●	▲	×
	aidl_getTagInfo()	デコメ絵文字の画像データ (GIF、JPEG) 内に格納されている拡張情報データを取得する。	●	●	▲	●
	aidl_changeHistory()	デコメ絵文字/デコメピクチャの利用履歴を更新する。	×	×	●	●
	aidl_resetHistoryCnt()	デコメ絵文字情報テーブル. 利用履歴表示順をリセットする。	×	×	●	●
	aidl_getHistoryUriList()	デコメ絵文字/デコメピクチャの利用履歴のリストを取得する。	×	×	▲	×
DecoEmojiContract	makeStringEmojiKind()	対象絵文字タイプ (“emojiType”) の値を絵文字種別の値に変換する。 (ContentProvider 利用時などに使用する)	×	×	▲	●
	convertEmojiType()	絵文字種別の値を対象絵文字タイプ (“emojiType”) の値に変換する。(入力フィールドの対象絵文字タイプ (“emojiType”) と、取得結果の絵文字種別を比較する際などに使用する)	×	×	▲	●

以下、各コンポーネントについて、インタフェース仕様を規定する。

なお、文中のパッケージ名については事業者ごとに異なる場合があるため、該当部分を”<carrier>”として表記している。

例: "jp.co.omronsoft.android.decoemoji manager_<carrier>"

14.1. 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) およ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);
```

14.2. EmojiAssistクラス

パッケージ名 : jp.co.omronsoft.android.emoji

14.2.1. 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。					

14.2.2. 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うる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。					

14.2.3. 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。					

14.2.4. 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。					

14.2.5. 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。					

14.2.6. 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。					

14.2.7. デコメ絵文字Span設定

API 名		getDecoEmojiText			
説明		デコメ絵文字属性（Annotation）を設定した文字を取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
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
返値					
No	型	説明			
1	CharSequence	デコメ絵文字 Span を設定した文字			
前提 なし					
概略説明					
IMEからアプリケーションに出力するデコメ絵文字属性（android.text.Annotation）を設定した文字を取得する。					
Annotationには以下の情報が設定されている。					
・コンテンツのURI					

14.2.8. デコメ絵文字Span設定(拡張版)

API 名	getDecoEmojiText				
説明	デコメ絵文字属性（Annotation）を設定した文字を取得する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
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
2	context	Context	I	Must	コンテキスト
返値					
No	型	説明			
1	CharSequence	デコメ絵文字 Span を設定した文字			
前提 なし					
概略説明					
IMEからアプリケーションに出力するデコメ絵文字属性（android.text.Annotation）を設定した文字を取得する。					
Annotationには以下の情報が”\t”区切りで設定されている。					
・コンテンツのURI ・画像ファイルの幅 ・画像ファイルの高さ ・絵文字種別 ・デコメ絵文字pop識別子					

サンプルコード：

```
CharSequence Candidate = EmojiAssist.getInstance().getDecoEmojiText (uri, context);
SpannableStringBuilder text = (SpannableStringBuilder)candidate;
for (Annotation s : text.getSpans(0, 1, Annotation.class)) {
    String[] split = s.getValue().split(EmojiAssist.SPLIT_KEY, 0);
    String uri = split[0];
    int width = Integer.parseInt(split[1]);
    int height = Integer.parseInt(split[2]);
    int emojiKind = Integer.parseInt(split[3]);
    int pop = Integer.parseInt(split[4]);
}
```


14.2.9.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；					
・ 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					
サイズ17を指定すると					
・ 15～18 pixelの時に表示する画像					
が取得される。					
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。					

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
}

```

14.2.10. ExtractEditText差し替え

API 名	replaceExtractEditText				
説明	ExtractEditTextを差し替える。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	view	ViewGroup	I	Must	ExtractEditTextを持つView
2	id	int	I	Must	差し替えるExtractEditTextのリソースID
返値					
No	型	説明			
1	ExtractEditText	差し替えた ExtractEditText 継承クラスのインスタンス(失敗時：NULL)			
前提 なし					
概略説明 全画面表示で使用する ExtractEditText クラスを、任意の継承クラスに置換する(IME 向け)					
補足 フレームワークの全画面表示用部品ExtractEditTextは絵文字表示において不具合が発生するため、絵文字パッケージではその不具合を回避するjp.co.omronsoft.android.inputmethodservice.EmojiExtractEditTextを用意している。IMEで全画面への切り替えを検出した際、EmojiExtractEditTextに差し替える必要がある。フレームワークのバージョンアップによりExtractEditTextが改善されれば、本対応は不要になる。					

サンプルコード:

```
public class MyIME extends InputMethodService {
    private ExtractEditText mExtractEditText;
    <中略>
    @Override public View onCreateExtractTextView() {
        View v = super.onCreateExtractTextView();
        EmojiAssist assist = EmojiAssist.getInstance();
        mExtractEditText = assist.replaceExtractEditText((ViewGroup)v,
        R.layout.emoji_extract_view);
        return v;
    }
    public ExtractEditText getExtractEditText() {
        return mExtractEditText;
    }
    <以下略>
}
```

サンプルコード (emoji_extract_view.xml) :

```
<jp.co.omronsoft.android.inputmethodservice.EmojiExtractEditText
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:scrollbars="vertical"
    android:gravity="top"
    android:minLines="1"
    android:inputType="text"
    android:background="@drawable/emoji_extract_edit_text"
    >
</jp.co.omronsoft.android.inputmethodservice.EmojiExtractEditText>
```

14.2.11. 入力文字列の絵文字有無判定

API 名

checkTextData

説明

入力文字列にキャリア絵文字 / デコメ絵文字 / デコメピクチャが含まれているかどうかの判定を行う。

引数

No	引数名	型	I/O	Must/Opt.	説明
1	text	Spannable	I	Must	Spannable型の文字列

返値

No	型	説明
1	int	EmojiAssist.TYPE_TEXT_NORMAL(0) ： キャリア絵文字 / デコメ絵文字 / デコメピクチャが含まれていない。 EmojiAssist.TYPE_TEXT_EMOJI(1) ： キャリア絵文字が含まれている。 EmojiAssist.TYPE_TEXT_DECOEMOJI(2) ： 「デコメ絵文字またはデコメピクチャ」が含まれている。 EmojiAssist.TYPE_TEXT_BOTH(3) ： キャリア絵文字と「デコメ絵文字またはデコメピクチャ」がともに含まれている。

前提

なし

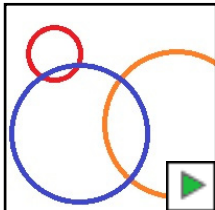
概略説明

入力文字列にキャリア絵文字 / デコメ絵文字 / デコメピクチャが含まれているかどうかを判定する。

補足

文字列の判定を行うことでViewの登録/削除、アニメーションの開始 / 停止を適切に判断することができ、CPU使用率を下げる事が可能になる。

14.2.12. デコメ絵文字pop識別子表示

API 名		setDisplayPop			
説明		デコメ絵文字pop識別子の表示可否を設定する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	Display	Boolean	I	Must	デコメ絵文字pop識別子の表示可否を指定。 true：デコメ絵文字pop識別子を表示する。 false：デコメ絵文字pop識別子を表示しない。
返値					
No	型	説明			
前提					
addView()で TextView に EmojiAssist を登録する前に実行すること。					
概略説明					
デコメ絵文字pop識別子の表示制御に利用する。					
本APIが利用されていない場合はデコメ絵文字pop識別子を表示しない。					
デコメ絵文字pop識別子の画像サイズは一定とし、画像の右下に表示する。					
表示イメージ					
① デコメ絵文字popのサイズが20px×20px の場合					
					
② デコメ絵文字popのサイズが20px×100px の場合					
					
③ デコメ絵文字popのサイズが40px×40px の場合					
					

14.2.13. デコメ絵文字pop識別子表示可否設定取得

API 名		getDisplayPop			
説明		デコメ絵文字pop識別子の表示可否設定を取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	boolean	デコメ絵文字pop識別子の表示可否設定。 true：デコメ絵文字pop識別子を表示する。 false：デコメ絵文字 pop 識別子を表示しない。			
前提					
なし					
概略説明					
デコメ絵文字pop識別子の表示可否設定を取得する。					

14.2.14. デコメピクチャ拡張表示設定

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張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張して表示 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張表示設定を行うことができる。					
拡張は、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。					

14.3. 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

14.3.1. 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();
```

14.3.2. 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。					

14.3.3. 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。					

14.3.4. デコメ絵文字pop識別子取得

API 名		getPopFlag			
説明		デコメ絵文字の画像ファイルのデコメ絵文字pop識別フラグを取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
返値					
No	型	説明			
1	int	デコメ絵文字pop識別フラグ			
前提 なし					
概略説明 デコメ絵文字の画像ファイルのデコメ絵文字pop識別フラグを取得する。 デコメ絵文字pop識別フラグの値と定義は以下の通り。 0： デコメ絵文字pop識別フラグなし 1： デコメ絵文字pop識別フラグあり					

14.3.5. 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();
```

14.4. EmojiDrawableクラス

パッケージ名 :jp.co.omronsoft.android.text

14.4.1. 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。					

14.4.2. 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.codePointAt0等で取得したコードを渡す。					

14.5. 絵文字一覧アプリ DecoEmojiList

絵文字一覧アプリは、IME や一般アプリから Intent によって起動され、選択された絵文字を結果として返却する。

絵文字一覧アプリを使用するアプリは、DecoEmojiList のソースコードのうち下記のファイルをアプリのビルド環境にコピーして利用する必要がある。

DecoEmojiListResultInfo.java

14.5.1. Intent起動

IME もしくはアプリから絵文字一覧アプリを起動させるには、下記の通り Intent を発行する。

サンプルコード:

```
Intent intent = new Intent();
// クラス名にデコメ絵文字一覧アプリを指定
intent.setClassName("jp.co.omronsoft.android.decoemojilist_<career>",
    "jp.co.omronsoft.android.decoemojilist_<career>.DecoEmojiList");(*1)
// 呼び出す Activity 以外の Activity をクリアして起動させる設定を付与
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
// デコメ絵文字一覧アプリ上で表示する絵文字タグ切替値の設定
intent.putExtra("switch_emoji_flg", 1); // 0:both、1:deco、2:carrier
// アプリ上で表示する対象絵文字タイプ("emojiType")の設定 (入力フィールド editText から取得)
Bundle b = editText.getInputExtras(true);
int et = b.getInt("emojiType");
intent.putExtra("emojiType", et);
// 縦画面時のデコメ絵文字一覧アプリを表示する高さの領域を設定
intent.putExtra("current_ime_height_port", heightP);
// 横画面時のデコメ絵文字一覧アプリを表示する高さの領域を設定
intent.putExtra("current_ime_height_land", heightL);
// デコメ絵文字取得方法を設定
intent.putExtra("current_input_type", 0); // 0:複数候補選択後にアプリに返す、1:候補を一つ選択した時点でアプリに返す
// デコメ絵文字一覧アプリ起動
startActivityForResult(intent, 0); // request code = 0
```

(*1) パッケージ名の詳細は「emojipkg_type2_ReleaseNotes.txt」に記載。

14.5.2. 返値取得

起動した絵文字一覧アプリにおいて選択された絵文字情報を取得するには、受け取った Intent の Result から List を取り出す。

サンプルコード:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == 0 && resultCode == RESULT_OK) { // req. code = 0,
        ArrayList<DecoEmojiListResultInfo> result = data
            .getParcelableArrayListExtra("uri_list");

        DecoEmojiListResultInfo resultInfo;
        for (int i = 0; i < result.size(); i++) {
            resultInfo = result.get(i);
            if (resultInfo.getTag() == 1) { // 1:deco, 2:carrier
                // デコメ絵文字
                CharSequence charSequence = resultInfo.getUri();
                :
            } else {
                // キャリア絵文字(Stringの各文字がキャリア絵文字の文字コード)
                CharSequence charSequence = resultInfo.getCode();
                :
            }
        }
    }
    finish();
}
```

14.6. 絵文字マネージャDecoEmojiManager

絵文字マネージャ DecoEmojiManager は、デコメ絵文字の画像ファイルと ID を管理するクラスである。

(1) デコメ絵文字の登録

システムで共通に利用するデコメ絵文字は、所定のディレクトリ(登録先ディレクトリ)に格納される必要がある。登録先ディレクトリに登録されたデコメ絵文字は、絵文字パッケージに管理され、アプリケーションから絵文字として利用(入力、表示)することができる。端末の別アプリ(メーラ、ブラウザやコンテンツ管理アプリ)が新規にデコメ絵文字を入手して端末(SD カードなど)に保存したり削除したりする場合、その更新を登録先ディレクトリに加えることによって、絵文字パッケージに通知することができる。

(2) アプリケーションへの通知(Intent)

デコメ絵文字の追加、または削除が検出された場合、DecoEmojiManager は Broadcast によりデコメ絵文字情報を利用アプリに通知する。また、デコメ絵文字予測/変換入力機能を含む場合は、さらに Broadcast によりデコメ絵文字辞書情報を通知する。

(3) アプリケーションインタフェース

デコメ絵文字の一覧を表示したり、デコメ絵文字をアプリケーションに入力するサービスを提供するアプリ(IME)や、デコメ絵文字の付加情報を利用するアプリは、DecoEmojiManager サービス(aidl)インタフェース、または DecoEmojiManager の提供する ContentProvider インタフェースを通じてそれらの情報にアクセスすることができる。

DecoEmojiManager は、サービスとして動作する。サービスに接続する場合は下記のように bind する必要がある。ContentProvider インタフェースについては次章を参照のこと。

なお、DecoEmojiManager のインタフェースおよび関連する定数値を利用するアプリは、ライブラリ DecoEmojiInterface.jar をリンクする必要がある。

サンプルコード:

```
// サービス接続
Intent intent = new Intent(IDecoEmojiManager.class.getName());
bindService(intent, mServiceConn, Context.BIND_AUTO_CREATE);

// bindService のコールバック関数
private ServiceConnection mServiceConn = new ServiceConnection() {
    // サービス接続時のコールバック関数
    public void onServiceConnected(ComponentName name, IBinder binder) {
        mDecoEmojiInterface = IDecoEmojiManager.Stub.asInterface(binder);
    }
    // サービス切断時のコールバック関数
    public void onServiceDisconnected(ComponentName name) {
        mDecoEmojiInterface = null;
    }
};

// サービス切断
unbindService(mServiceConn);
```


14.6.1. 絵文字マネージャのサービス(aidl)インタフェースのAPI

DecoEmojiManager のサービスとして提供されている API について説明する。

パッケージ名 : jp.co.omronsoft.android.decoemojimananger_<career>.interfacedata

14.6.1.1. URI取得

API 名		aidl_getDecoUri			
説明		デコメ絵文字のURIを取得する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	decoemoji_id	int	I	Must	デコメ絵文字ID … 取得対象デコメ絵文字のデコメ絵文字ID デコメ絵文字ID : 0～9999
2	uriInfo	DecoEmojiUriInfo	I/O	Must	URI … 対象デコメ絵文字のURIを格納したDecoEmojiUriInfoのオブジェクト
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK : 正常終了 IDecoEmojiConstant.NG_INITIALIZING : 初期化処理中 IDecoEmojiConstant.NG_ID : 「decoemoji_id」の値が範囲外 IDecoEmojiConstant.NG_URI_INFO : URI の取得中に問題が発生（DB エラーを含む）			
前提					
DecoEmojiManager により初期データが構築されていること					
概略説明					
指定されたデコメ絵文字IDに紐づくデコメ絵文字のURIを返却する。					

14.6.1.2. カテゴリー一覧取得

API 名	aidl_getCategoryList				
説明	デコメ絵文字のカテゴリ情報一覧を返却する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	maxDisplaycnt	int	I	Must	1画面に表示するデコメ絵文字の数 (0 < maxDisplaycnt <= 72)
2	categoryList	List<DecoEmojiCategoryInfo>	O	-	取得したカテゴリ情報のList
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK：正常終了 IDecoEmojiConstant.NG_MAX_DISPLAY_CNT：第 1 引数「maxDisplaycnt」の値が範囲外 IDecoEmojiConstant.NG_CATEGORY_INFO：カテゴリ情報の取得中に問題が発生した (DB エラーを含む)			
前提					
DecoEmojiManager により初期データが構築されていること					
概略説明					
デコメ絵文字のカテゴリ情報一覧を返却する。 本APIにて返却する対象絵文字種別は「①デコメ絵文字(20*20)」のみ。					

14.6.1.3. カテゴリー一覧取得（デコメ絵文字用拡張）

API 名		aidl_getCategoryList_ex			
説明		デコメ絵文字のカテゴリ情報一覧を返却する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	maxDisplaycnt	int	I	Must	1画面に表示するデコメ絵文字の数 (0 < maxDisplaycnt <= 72)
2	categoryList	List<DecoEmojiCategoryInfo>	O	-	取得したカテゴリ情報のList
3	emojiType	int	I	Must	表示対象の対象絵文字タイプ (“emojiType”) を指定する (0x0000 <emojiType <= 0x000F)
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK : 正常終了 IDecoEmojiConstant.NG_CATEGORY_INFO : カテゴリ情報の取得中に問題が発生した (DB エラーを含む)			
前提					
DecoEmojiManager により初期データが構築されていること。					
概略説明					
デコメ絵文字のカテゴリ情報一覧を返却する。					
第3引数(emojiType)にて指定された絵文字種別の画像を含むカテゴリを返却する。					
補足					
現在、emojiTypeにて指定できる対象絵文字タイプ (“emojiType”) は以下のいずれか。					
値	絵文字種別				
0x000F	デコメ絵文字およびデコメピクチャ				
0x000E	デコメピクチャ				
0x0001	デコメ絵文字				
対象絵文字タイプ (“emojiType”) に上記以外の値を指定された場合の動作は不定。					

14.6.1.4. URI一覧取得

API 名		aidl_getDecoUriList			
説明		引数のカテゴリIDに紐づくデコメ絵文字のURI一覧を返却する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	categoryId	int	I	Must	取得対象カテゴリのID（1 ≤ カテゴリID ≤ intの最大値）
2	pageIndex	int	I	Must	取得対象のページ番号（0オリジン）
3	uriList	List<String>	O	-	指定されたページに表示するデコメ絵文字のURIのList
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK：正常終了 IDecoEmojiConstant.NG_ID：第1引数「categoryId」の値が範囲外 IDecoEmojiConstant.NG_PAGE_CNT：第2引数「pageIndex」で指定されたページが存在しない IDecoEmojiConstant.NG_URI_INFO：URIの取得中に問題が発生した（DBエラーを含む） IDecoEmojiConstant.NG_DISP_CNT：表示数が未設定			
前提					
DecoEmojiManagerにより初期データが構築されていること 利用前に、aidl_getCategoryList0を呼び出し、あらかじめ表示数を設定しておくこと（未設定の場合、エラーとなる）。					
概略説明					
引数のカテゴリIDに紐づくデコメ絵文字のURI一覧を返却する。 本APIにて返却する絵文字種別は「①デコメ絵文字(20*20)」のみ。					

サンプルコード:

```

IDecoEmojiManager mIDecoEmojiManagerInterface = null;
ServiceConnection serviceConn = new ServiceConnection() {
    public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
        mIDecoEmojiManagerInterface = IDecoEmojiManager.Stub.asInterface(iBinder);
        :
    }
}
:
Intent intentBindSearvice = new Intent(IDecoEmojiManager.class.getName());
bindService(intentBindSearvice, serviceConn, Context.BIND_AUTO_CREATE);
:
List<DecoEmojiCategoryInfo> categoryInfoList = new
ArrayList<DecoEmojiCategoryInfo>();
List<String> decoEmojiUriList = new ArrayList<String>();
mIDecoEmojiManagerInterface.aidl_getCategoryList(60, categoryInfoList); // 60 個まで
mIDecoEmojiManagerInterface.aidl_getDecoUriList(categoryInfoList[0].getCategoryId(),
0, decoEmojiUriList);

```

14.6.1.5. URI一覧取得（デコメ絵文字用拡張）

API 名		aidl_getDecoUriList_ex			
説明		引数のカテゴリIDに紐づくデコメ絵文字のURI一覧を返却する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	categoryId	int	I	Must	取得対象カテゴリのID (1 <= カテゴリID <= intの最大値)
2	pageIndex	int	I	Must	取得対象のページ番号 (0オリジン)
3	uriList	List<String>	O	-	指定されたページに表示するデコメ絵文字のURIのList
4	emojiType	int	I	Must	表示対象の対象絵文字タイプ (“emojiType”) を指定する (0x0000 <emojiType <= 0x000F)
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK : 正常終了 IDecoEmojiConstant.NG_ID : 第 1 引数「categoryId」の値が範囲外 IDecoEmojiConstant.NG_PAGE_CNT : 第 2 引数「pageIndex」で指定されたページが存在しない IDecoEmojiConstant.NG_URI_INFO : URI の取得中に問題が発生した (DB エラーを含む) IDecoEmojiConstant.NG_DISP_CNT : 表示数が未設定			
前提					
DecoEmojiManager により初期データが構築されていること。 利用前に、aidl_getCategoryList_ex0を呼び出し、あらかじめ表示数を設定しておくこと (未設定の場合、エラーとなる)。					
概略説明					
引数のカテゴリIDに紐づくデコメ絵文字のURI一覧を返却する。 第3引数(emojiType)にて指定された絵文字種別を含むカテゴリを返却する。					
補足					
現在、emojiTypeにて指定できる対象絵文字タイプ (“emojiType”) は以下のいずれか。					
値	絵文字種別				
0x000F	デコメ絵文字およびデコメピクチャ				
0x000E	デコメピクチャ				
0x0001	デコメ絵文字				
対象絵文字タイプ (“emojiType”) に上記以外の値を指定された場合の動作は不定。					

サンプルコード:

```
IDecoEmojiManager mIDecoEmojiManagerIF = null;
ServiceConnection serviceConn = new ServiceConnection() {
    public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
        mIDecoEmojiManagerIF = IDecoEmojiManager.Stub.asInterface(iBinder);
        :
    }
}
:
Intent intentBindService = new Intent(IDecoEmojiManager.class.getName());
bindService(intentBindService, serviceConn, Context.BIND_AUTO_CREATE);
:
List<DecoEmojiCategoryInfo> categoryInfoList = new
ArrayList<DecoEmojiCategoryInfo>();
List<String> decoEmojiUriList = new ArrayList<String>();
mIDecoEmojiManagerIF.aidl_getCategoryList_ex(60, categoryInfoList, 0x000F); // 60
個まで
mIDecoEmojiManagerIF.aidl_getDecoUriList_ex(categoryInfoList[0].getCategoryId(),
0, decoEmojiUriList, 0x000F);
```

14.6.1.6. デコメ絵文字更新チェック

API 名		aidl_checkDecoEmoji			
説明		引数のデコメ絵文字IDより後に追加更新されたデコメ絵文字を抽出し、追加分のデコメ絵文字辞書更新情報をブロードキャスト送信する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	decoemoji_id	int	I	Must	デコメ絵文字ID … IME側で最後に追加したデコメ絵文字ID デコメ絵文字ID : -1(*1), 0～9,999
2	ime_name	String	I	Must	IMEパッケージ名 … 呼出元IMEのパッケージ名
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK : 正常終了 IDecoEmojiConstant.NG_INITIALIZING : 初期化処理中 IDecoEmojiConstant.NG_DIC_INFO : 読み/品詞の取得中に問題が発生 (DB エラーを含む) IDecoEmojiConstant.NG_ID : 「decoemoji_id」の値が範囲外 IDecoEmojiConstant.NG_IME_NAME : 「ime_name」の値が null			
前提					
DecoEmojiManager により初期データが構築されていること					
概略説明					
引数のデコメ絵文字IDより後に追加されたデコメ絵文字を抽出し、追加分のデコメ絵文字辞書操作情報をブロードキャスト送信する。					
(*1) 全デコメ絵文字データの再取得処理を行う場合は、第一引数(decoemoji_id) に -1 を指定する。					

14.6.1.7. デコメ絵文字拡張情報取得

デコメ絵文字の画像データ (GIF、JPEG) 内に格納されている拡張情報データを取得するために用いる。

この API を使用するためには、デコメ絵文字の画像データ内に取得したい拡張情報データが格納されている必要がある。フォーマットについては「[デコメ絵文字フォーマット](#)」参照のこと。

API 名	aidl_getTagInfo				
説明	デコメ絵文字の画像データ（GIF、JPEG）内に格納されている拡張情報データを取得する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	uri	String	I	Must	取得したいデコメ絵文字URI
2	tag_name	String	I	Must	取得したいカスタムタグ名
3	tag_info	List<String>	O	Must	指定されたデコメ絵文字URIのカスタムタグから取得した拡張情報。 最大 10 件まで取得できる。
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK：正常終了 IDecoEmojiConstant.NG_URI：第 1 引数(uri)が null IDecoEmojiConstant.NG_TAG：第 2 引数(tag_name)が null IDecoEmojiConstant.NG_TAG_INFO_NULL：第 3 引数(tag_info)が null IDecoEmojiConstant.NG_TAG_INFO_READ：拡張情報取得中に問題が発生			
概略説明					
・第1引数(uri)で指定されたコンテンツの付加情報において、第2引数(tag_name)と一致するカスタムタグ名(<tags>タグ内のname属性の値)が存在した場合、要素の内容を第3引数(tag_info)内にセットして返す。 ・第2引数(tag_name)と一致するカスタムタグ名が存在しない場合、IDecoEmojiConstant.NG_TAG_INFO_READ を返す。 ・第1引数(uri)で指定されたコンテンツの付加情報に、第2引数(tag_name)と一致するカスタムタグが複数件存在する場合、最大 10 件まで取得することができる。10 件以上のデータが存在する場合は、前方から 10 件分を取得する。					

14.6.1.8. 履歴情報・表示順更新

API 名	aidl_changeHistory				
説明	デコメ絵文字/デコメピクチャの利用履歴を更新する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	uri	String	I	Must	履歴を更新したいデコメ絵文字URI
2					
3					
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK：正常終了 IDecoEmojiConstant.NG_URI：第 1 引数(uri)が null IDecoEmojiConstant.NG_HISTORY_CHANGE：履歴情報を更新中に問題が発生			
概略説明					
・第1引数(uri) に指定したデコメ絵文字の履歴情報を最新のものに更新する。 更新対象は、デコメ絵文字情報テーブル.最終利用表示順およびデコメ絵文字情報テーブル.利用履歴表示順となる。					

14.6.1.9. 履歴情報・表示順リセット

API 名	aidl_resetHistoryCnt				
説明	デコメ絵文字情報テーブル、利用履歴表示順をリセットする。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	target	Int	I	Must	リセットする対象を指定する (値は概略説明を参照のこと)
2					
3					
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK : 正常終了 IDecoEmojiConstant.NG_TARGET : 第 1 引数(target)が不正 (3 以上、null) IDecoEmojiConstant.NG_HISTORY_RESET : 履歴情報をリセット中に問題が発生			
概略説明					
- デコメ絵文字情報テーブル、利用履歴表示順をリセットする。 - 第1引数(target)に指定したフィールドを対象とする。値の詳細は以下の通り。 0 : 履歴情報を削除し、カテゴリ内表示順を初期化する。 (デコメ絵文字情報テーブル、最終利用表示順 / 利用履歴表示順の両方をリセット) 1 : カテゴリ内表示順を初期化する。 (デコメ絵文字情報テーブル、最終利用表示順をリセット) 2 : 履歴情報を削除する。 (デコメ絵文字情報テーブル、利用履歴表示順をリセット)					

14.6.1.10. 履歴情報URI取得

API 名	aidl_getHistoryUriList				
説明	デコメ絵文字情報テーブル、利用履歴表示順にURIを取得する。				
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	maxGetcnt	int	I	Must	取得対象のデコメ絵文字の最大数
2	emojiType	int	I	Must	表示対象の対象絵文字タイプ（“emojiType”）を指定する (0x0000 <emojiType <= 0x000F)
3	uriList	List<String>	O	-	取得対象のデコメ絵文字のURIのList
返値					
No	型	説明			
1	int	IDecoEmojiConstant.OK：正常終了 IDecoEmojiConstant.NG_MAX_GET_CNT：第 1 引数「maxGetcnt」の値が不正 IDecoEmojiConstant.NG_URI_INFO：URI の取得中に問題が発生した (DB エラーを含む)			
概略説明					
・ 第一引数(maxGetcnt)に指定された最大数の範囲内で、利用履歴の順にデコメ絵文字のURIを取得する。					

14.6.2. 利用クラス

パッケージ名 : jp.co.omronsoft.android.decoemojimanager_<career>.interfacedata

14.6.2.1. DecoEmojiUriInfo クラス

メソッド	引数	内容
DecoEmojiUriInfo()	なし	オブジェクトの生成
DecoEmojiUriInfo(Parcel in)	in:Parcel 指定	in から文字列を読み出し、URI として設定する
getUri()	なし	保持する URI を返す
setUri(String uri)	uri : URI	URI を設定する

14.6.2.2. DecoEmojiCategoryInfo クラス

メソッド	引数	内容
DecoEmojiCategoryInfo()	なし	オブジェクトの生成
DecoEmojiCategoryInfo(Parcel in)	in:Parcel 指定	in から id(int)、日本語カテゴリ名(文字列)、英語カテゴリ名(文字列)、ページ数(int)を読み出して設定する
int getCategoryId()	なし	保持するカテゴリの ID を返す
String getCategoryName_jpn()	なし	保持するカテゴリ名(日本語)を返す
String getCategoryName_eng()	なし	保持するカテゴリ名(英語)を返す
int getCategoryPresetId()	なし	保持するカテゴリプリセット ID を返す
int getPageCnt()	なし	保持するページ数を返す
void setCategoryId(int category_id)	category_id: カテゴリ ID	カテゴリ ID を設定する
void setCategoryName_jpn(String category_name)	category_name : カテゴリ名	カテゴリ名(日本語)を設定する
void setCategoryName_eng(String category_name)	category_name : カテゴリ名	カテゴリ名(英語)を設定する
void setCategoryPresetId(int category_preset_id)	Category_preset_id: カテゴリプリセット ID	カテゴリプリセット ID を設定する
void setPageCnt(int pagecnt)	pagecnt : ページ数	ページ数を設定する

14.6.2.3. DecoEmojiAttrInfo クラス

メソッド	引数	内容
DecoEmojiAttrInfo()	なし	オブジェクトの生成
DecoEmojiAttrInfo(Parcel in)	in:Parcel 指定	in から ID(int)、読み(文字列配列)、品詞(byte 配列)、表記(文字列配列)を読み出して設定する
int getId()	なし	ID を取得する
void setID(int id)	id : ID	ID を設定する
String getName(int idx)	idx:インデックス	idx 番目の読みを取得する
byte getPart(int idx)	idx:インデックス	idx 番目の品詞を取得する
void setName(int idx, String name)	idx:インデックス name : 読み	idx 番目の読みを設定する
void setPart(int idx, byte part)	idx:インデックス part : 品詞	idx 番目の品詞を設定する
String getNote(int idx)	idx:インデックス	idx 番目の表記を取得する
void setNote(int idx, String note)	idx:インデックス note : 表記	idx 番目の表記を設定する

14.6.3. 定数

14.6.3.1. DecoEmojiManagerエラー値

名前	意味	値
IDecoEmojiConstant.OK	正常終了	0
IDecoEmojiConstant.NG_INITIALIZING	エラー:初期データ構築中	-1
IDecoEmojiConstant.NG_DIC_INFO	エラー:読み/品詞の取得中に問題が発生した(DB エラーを含む)	-2
IDecoEmojiConstant.NG_ID	エラー:「decoemoji_id」の値が範囲外	-3
IDecoEmojiConstant.NG_OPERATION	エラー:「flag」の値が範囲外	-5
IDecoEmojiConstant.NG_IME_NAME	エラー:「ime_info」の値が null	-6
IDecoEmojiConstant.NG_URI_INFO	エラー:URIの取得中に問題が発生(DB エラーを含む)	-10
IDecoEmojiConstant.NG_DIC_FULL	エラー:登録済の読み/品詞の件数が上限(20000)に達した	-11
IDecoEmojiConstant.NG_ID_FULL	エラー:登録済のデコメ絵文字の件数が上限(10000)に達した	-12
IDecoEmojiConstant.NG_NOT_REGISTERED_ID	エラー:登録されていない URI を指定	-18
IDecoEmojiConstant.NG_URI	エラー:「uri」の値が不正	-19
IDecoEmojiConstant.NG_REGISTERED_URI	エラー:登録済の URI を指定	-20
IDecoEmojiConstant.NG_TAG	エラー:引数の拡張タグ名が null	-21
IDecoEmojiConstant.NG_TAG_INFO_NULL	エラー:引数の拡張タグ情報リストが null	-22
IDecoEmojiConstant.NG_TAG_INFO_READ	エラー:拡張タグ情報取得中に問題が発生	-23

14.6.3.2. DecoEmojiManager操作種別

名前	意味	値
IDecoEmojiConstant.FLAG_INSERT	追加	0
IDecoEmojiConstant.FLAG_DELETE	削除	2
IDecoEmojiConstant.FLAG_SET_TO_INITIALIZING	初期化	4
IDecoEmojiConstant.FLAG_FINISH	更新データなし	9

14.6.3.3. DecoEmojiManagerブロードキャスト関連

名前	意味	値
IDecoEmojiConstant.BROADCAST_DATA_TAG	データ	"DecoEmojiInformation"
IDecoEmojiConstant.BROADCAST_ID_TAG	ID	"Id"
IDecoEmojiConstant.BROADCAST_TYPE_TAG	タイプ	"Type"
IDecoEmojiConstant.BROADCAST_TIMESTAMPTAG	更新タイムスタンプ	"TimeStamp"
IDecoEmojiConstant.ACTION_DECOEMOJI_RESULT	デコメ絵文字辞書更新情報	"jp.co.omronsoft.android.dec oemoji_manager_<career>"
IDecoEmojiConstant.ACTION_CHANGE_DATABASE	デコメ絵文字情報更新情報	"jp.co.omronsoft.android.dec oemoji_manager_<career>.C HANGE_DATABASE"

14.6.4. DecoEmojiManager発行Intent

DecoEmojiManager から送出される Intent の詳細について説明する。

DecoEmojiManager では、次の二つの Intent を発行する。

(1)デコ絵文字情報更新通知

デコ絵文字情報を保持するアプリに対して、デコ絵文字情報の更新を通知する。

(2)デコ絵文字辞書更新通知

デコ絵文字辞書情報を保持するアプリに対し、デコ絵文字辞書情報の更新を通知する。

14.6.4.1. デコ絵文字情報更新通知

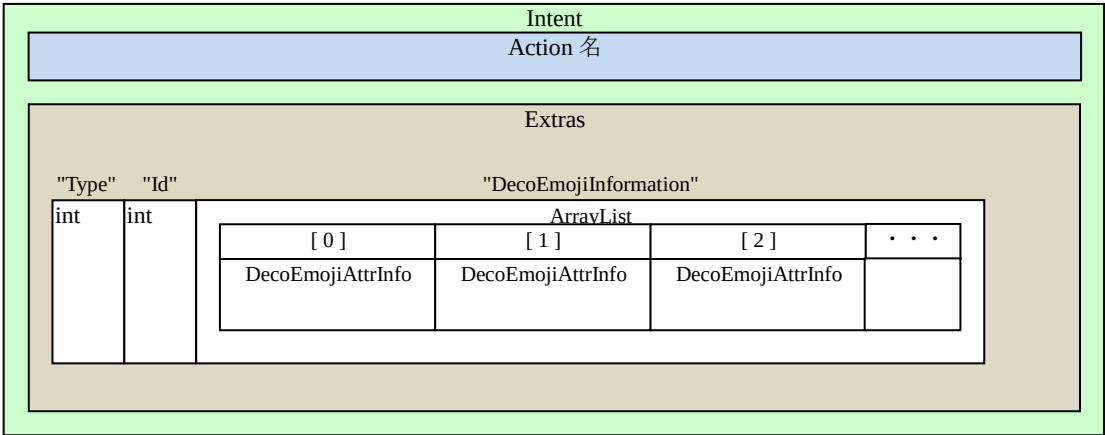
データ名称		入出力内容
Action 名		IDecoEmojiConstant.ACTION_CHANGE_DATABASE
Extras	IDecoEmojiConstant.BROADCAST_TYPE_TAG	更新区分。格納する区分は下記の通り。 追加:0 削除:2 初期化:4
	IDecoEmojiConstant.BROADCAST_URI_TAG	更新画像の URI ★更新区分が初期化時は null
	IDecoEmojiConstant.BROADCAST_TIMESTAMP_TAG	更新画像のタイムスタンプデータ ★更新区分が初期化時は null

サンプルコード:

```
public void onReceive(final Context context, Intent intent) {
    // intent 識別子を取得
    String action = intent.getAction();
    if (action.equals(IDecoEmojiConstant.ACTION_CHANGE_DATABASE)) {
        // IntentBundle を取得する
        Bundle bundle = intent.getExtras();
        // Type を取得する。
        int type = bundle.getInt(IDecoEmojiConstant.BROADCAST_TYPE_TAG);
        if (type != IDecoEmojiConstant.FLAG_SET_TO_INITIALIZING) {
            // URI を取得する。
            String uri = bundle.getString(IDecoEmojiConstant.BROADCAST_URI_TAG);
            // 更新時の TimeStamp を取得する。
            String timestamp =
                bundle.getString(IDecoEmojiConstant.BROADCAST_TIMESTAMP_TAG);
        }
    }
}
```

14.6.4.2. デコメ絵文字辞書更新通知

Intent の構造



データ名称		入出力内容
Action 名		★デコメ絵文字更新チェック呼出に対する応答の場合 呼出元 IME のパッケージ名 ★上記以外の場合 IDecoEmojiConstant.ACTION_DECOEMOJI_RESULT
Extras	Type	更新区分。格納する区分は下記の通り。 追加:0 削除:2 初期化:4 更新データなし:9
	Id	受信済デコメ絵文字 ID。デコメ絵文字更新チェック呼出に対する応答時のみ有効な値を格納する。 ★デコメ絵文字更新チェック呼出に対する応答の場合 デコメ絵文字更新チェック呼出時、引数「decoemoji_id」で渡されたデコメ絵文字 ID ★上記以外の場合 -1 固定
	DecoEmojiInformation	DecoEmojiAttrInfo の ArrayList。 DecoEmojiAttrInfo には、更新対象デコメ絵文字 ID、読み一覧、品詞一覧が含まれる。

サンプルコード:

```
public void onReceive(final Context context, Intent intent) {
    Bundle bundle = intent.getExtras();
    int type = bundle.getInt(IDecoEmojiConstant.BROADCAST_TYPE_TAG);
    if (type != IDecoEmojiConstant.FLAG_FINISH) {
        List<DecoEmojiAttrInfo> receivedatalist =
            bundle.getParcelableArrayList(IDecoEmojiConstant.BROADCAST_DATA_TAG);
    }
}
```

14.7. 絵文字マネージャDecoEmojiManager (Content Provider)インタフェース

14.7.1. 絵文字パッケージで提供する Content Provider 機能

Android フレームワークでは、Content Provider と呼ばれるデータ操作／参照機能が用意されている。

絵文字パッケージでは、絵文字マネージャ (DecoEmojiManager) が管理するデコメ絵文字の DB を参照 (query) できる ContentProvider 機能を提供する。

外部のアプリケーションから Content Provider 経由でのデータ操作(追加(insert)／変更(update)／削除(delete))はできない。

14.7.2. 前提条件

絵文字パッケージでの Content Provider 機能の前提は以下の通り。

- ・ 絵文字パッケージがインストールされていること。
- ・ デコメ絵文字を管理するデータベースが存在すること。

もしデータベースが存在しない状態で本機能を使用した場合は、エラーとなる。

なお、「絵文字マネージャ(DecoEmojiManager)」は、データベースが存在しない場合、起動時にデータベース

を生成する仕組みであるため、起動から一定時間経過後に本機能は使用可能となる。

14.7.3. 取得モード一覧および取得データ一覧

本 ContentProvider 機能において、データ取得インタフェースの種別を「取得モード」と呼ぶ。取得モードごとの取得データは以下の通りである。

取得データの詳細は「14.7.5 取得データ詳細」参照。

No	取得モード	取得モード別(実データ)	取得データ
1	デコメ絵文字情報一覧取得	decoinfo	デコメ絵文字情報テーブル
2	デコメ絵文字情報件数取得	decoinfo	デコメ絵文字情報件数テーブル
3	デコメ絵文字辞書一覧取得	decoiclist	デコメ絵文字辞書テーブル
4	デコメ絵文字辞書件数取得	decoiccount	デコメ絵文字辞書件数テーブル

14.7.4. 使用方法

Content Provider では、ContentURI に基づき、データを取得する。

本 ContentProvider での取得モード別の ContentURI 定数は以下の通り。

No	取得モード	ContentURI 定数
1	デコメ絵文字情報一覧取得	CONTENT_DECOINFO_URI
2	デコメ絵文字情報件数取得	CONTENT_DECOINFO_COUNT_URI
3	デコメ絵文字辞書一覧取得	CONTENT_DECOICLIST_URI
4	デコメ絵文字辞書件数取得	CONTENT_DECOIC_COUNT_URI

なお、No.1、3 については、詳細な条件指定をせずに簡易にデータを取得する方法もある。

詳細は「14.7.6 取得モード詳細」参照。

14.7.5. 取得データ詳細

14.7.5.1. デコメ絵文字情報テーブル

No	項目名称	型	備考
1	デコメ絵文字 ID	INTEGER	
2	デコメ絵文字 URI	TEXT	
3	カテゴリ ID	INTEGER	
4	カテゴリ名 (日本語)	TEXT	
5	カテゴリ名 (英語)	TEXT	
6	カテゴリプリセット ID	INTEGER	カテゴリプリセット ID ・プリセットに設定されているカテゴリは、プリセットの設定順に 0 から順に割り当てられる。 ・プリセット以外の ID については、プリセットカテゴリ数が ID として割り当てられる。 ・未分類については、最大カテゴリ数+1が ID として割り当てられる。
7	ディレクトリ ID	INTEGER	
8	ディレクトリ名	TEXT	
9	更新タイムスタンプ	INTEGER	
10	ファイルサイズ	INTEGER	
11	拡張タグ名	TEXT	「 デコメ絵文字拡張情報取得 」機能に関連。データベース登録時にカスタムタグ名(<tags >タグ内のname属性の値)が登録されている。複数のカスタムタグ名が存在する場合には、カンマ区切りで登録されている。
12	幅	INTEGER	
13	高さ	INTEGER	
14	絵文字種別	INTEGER	1 : 「デコメ絵文字」 2～4 : 「デコメピクチャ」
15	最終利用表示順	INTEGER	
16	利用履歴表示順	INTEGER	
17	デコメ絵文字 pop 識別フラグ	INTEGER	0 : デコメ絵文字pop識別フラグなし 1 : デコメ絵文字pop識別フラグあり

14.7.5.2. デコメ絵文字情報件数テーブル

No	項目名称	型	備考
1	デコメ絵文字情報件数	INTEGER	

14.7.5.3. デコメ絵文字辞書テーブル

No	項目名称	型	備考
1	デコメ絵文字 ID	INTEGER	
2	デコメ絵文字 URI	TEXT	
3	読み	TEXT	
4	品詞	INTEGER	
5	表記	TEXT	
6	更新タイムスタンプ	INTEGER	

14.7.5.4. デコメ絵文字辞書件数テーブル

No	項目名称	型	備考
1	デコメ絵文字辞書件数	INTEGER	

14.7.6. 取得モード詳細

14.7.6.1. デコメ絵文字情報一覧取得I F

インタフェース名	デコメ絵文字情報一覧取得			
取得モード	decoinfohist			
説明	デコメ絵文字情報テーブルから、検索条件に沿ったデコメ絵文字情報データ一覧を取得する。			
引数				
No	引数名	型	Must/Opt.	説明
1	Uri	Uri	Must	・ 詳細取得方式 DecoEmojiContract.CONTENT_DECOINFOHIST_URI ・ 簡易取得方式 DecoEmojiContract.CONTENT_DECOINFOHIST_URI/[デコメ絵文字ID] ※デコメ絵文字IDを指定して取得する場合は[デコメ絵文字ID]にデコメ絵文字IDを指定します。
2	projection	String[]	Opt	取得カラム … 取得するカラム名 ・ 省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・ (*1)取得項目参照
3	selection	String	Opt	フィルタ … フィルタ(Where条件) ・ 省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・ (*2)取得条件参照
4	selectionArgs	String[]	Opt	クエリパラメータ … クエリパラメータ ・ 省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・ (*2)取得条件参照
5	sortOrder	String	Opt	ソート順 … ソート順 ・ 省略する場合はNULLを指定する。その場合、昇順で取得する。 ・ (*3)ソート順参照
戻り値				
No	型	説明		
1	Cursor	デコメ絵文字情報テーブルへのカーソル		
(*1)取得項目				
No	項目名	項目 ID	型	備考
1	デコメ絵文字ID	decoemoji_id	INTEGER	
2	デコメ絵文字URI	uri	TEXT	
3	カテゴリID	category_id	INTEGER	
4	カテゴリ名（日本語）	category_name_jpn	TEXT	カテゴリが存在しない場合、NULLを返す。
5	カテゴリ名（英語）	category_name_eng	TEXT	カテゴリが存在しない場合、NULLを返す。
6	カテゴリプリセットID	category_preset_id	INTEGER	
7	ディレクトリID	directory_id	INTEGER	
8	ディレクトリ名	directory_name	TEXT	ディレクトリが存在しない場合、NULLを返す。
9	更新タイムスタンプ	timestamp	INTEGER	
10	ファイルサイズ	file_size	INTEGER	
11	拡張タグ名	tags	TEXT	
12	幅	width	INTEGER	
13	高さ	height	INTEGER	
14	絵文字種別	kind	INTEGER	1：「デコメ絵文字」 2～4：「デコメピクチャ」
15	最終利用表示順	last_use_cnt	INTEGER	
16	利用履歴表示順	history_cnt	INTEGER	
17	デコメ絵文字pop識別フラグ	decome_pop_flag	INTEGER	0：デコメ絵文字pop識別フラグなし 1：デコメ絵文字pop識別フラグあり

(*2)取得条件				
No	項目名	項目 ID	型	備考
1	デコメ絵文字ID	decoemoji_id	INTEGER	
2	デコメ絵文字URI	uri	TEXT	
3	カテゴリID	category_id	INTEGER	
4	カテゴリ名（日本語）	category_name_jpn	TEXT	
5	カテゴリ名（英語）	category_name_eng	TEXT	
6	カテゴリプリセットID	category_preset_id	INTEGER	
7	ディレクトリID	directory_id	INTEGER	
8	ディレクトリ名	directory_name	TEXT	
9	更新タイムスタンプ	timestamp	INTEGER	
10	ファイルサイズ	file_size	INTEGER	
11	拡張タグ名	tags	TEXT	
12	幅	width	INTEGER	
13	高さ	height	INTEGER	
14	絵文字種別	kind	INTEGER	1：「デコメ絵文字」 2～4：「デコメピクチャ」
15	最終利用表示順	last_use_cnt	INTEGER	
16	利用履歴表示順	history_cnt	INTEGER	
17	デコメ絵文字pop識別フラグ	decome_pop_flag	INTEGER	0：デコメ絵文字pop識別フラグなし 1：デコメ絵文字pop識別フラグあり
(*3)ソート順				
No	項目名	項目 ID	昇順(ASC)／ 降順(DISC)	初期値
1	デコメ絵文字ID	decoemoji_id	昇順・降順	昇順
2	デコメ絵文字URI	uri	昇順・降順	-
3	カテゴリID	category_id	昇順・降順	-
4	カテゴリ名（日本語）	category_name_jpn	昇順・降順	-
5	カテゴリ名（英語）	category_name_eng	昇順・降順	-
6	カテゴリプリセットID	category_preset_id	昇順・降順	
7	ディレクトリID	directory_id	昇順・降順	-
8	ディレクトリ名	directory_name	昇順・降順	-
9	更新タイムスタンプ	timestamp	昇順・降順	-
10	ファイルサイズ	file_size	昇順・降順	-
11	拡張タグ名	tags	昇順・降順	-
12	幅	width	昇順・降順	
13	高さ	height	昇順・降順	
14	絵文字種別	kind	昇順・降順	1：「デコメ絵文字」 2～4：「デコメピクチャ」
15	最終利用表示順	last_use_cnt	昇順・降順	
16	利用履歴表示順	history_cnt	昇順・降順	
17	デコメ絵文字pop識別フラグ	decome_pop_flag	昇順・降順	0：デコメ絵文字pop識別フラグなし 1：デコメ絵文字pop識別フラグあり

サンプルコード(詳細取得方式):

```
String column[] = {DecoEmojiContract.DecoEmojiInfoColumns.USE_FLAG};
String selection = DecoEmojiContract.DecoEmojiInfoColumns.DIRECTORY_NAME + " like ?";
String arg[] = {"%a%"};
String order = DecoEmojiContract.DecoEmojiInfoColumns.DIRECTORY_ID + " DESC";
Cursor cur = resolver.query(DecoEmojiContract.CONTENT_DECOINFOLIST_URI, column, selection, arg, order);
```

サンプルコード(簡易取得方式):デコメ絵文字 ID = 1234 を取得したい場合

```
Cursor cur = resolver.query(DecoEmojiContract.CONTENT_DECOINFOLIST_URI/1234, null, null, null, null);
```

サンプルコード:カテゴリ一覧を取得したい場合

```
// 入力フィールドに設定されている対象絵文字タイプ (“emojiType”) を取得
int emojiType = bundle.getInt(“emojiType”);
// 絵文字タイプを SQL 条件文字列に変換
String kind = DecoEmojiContract.makeStringEmojiKind(emojiType);
// 条件句を作成
// 条件として絵文字種別の設定を行う
String selection = DecoEmojiContract.DecoEmojiInfoColumns.KIND + " IN (" + kind + ")";
// order 句を作成
// プリセットの順番に取得できるように設定
String order = DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_PRESET_ID + " asc" ;
// カテゴリ取得
Cursor cursor = mContext.getContentResolver()
    .query(DecoEmojiContract.CONTENT_DECOINFOLIST_URI, null, selection, null, order);

if (cursor != null) {
    while (cursor.moveToNext()) {
        DecoEmojiCategoryInfo values = new DecoEmojiCategoryInfo();
        int categoryId =
            cursor.getInt(cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_ID));
        values.setCategoryId(categoryId); values.setCategoryName_jpn(cursor.getString(cursor
            .getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_NAME_JPN)));
        values.setCategoryName_eng(cursor
            .getString(cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_NAME_ENG)));
        boolean overlap = false;
        // 1 カテゴリ内に複数のコンテンツがある場合には、データが重複している為、
        // 重複チェックを行う
        for (int cnt = 0; cnt < mCategoryInfoList.size(); cnt++) {
            if (mCategoryInfoList.get(cnt).getCategoryId() == categoryId) {
                overlap = true;
                break;
            }
        }
        // 重複がある場合には、次のリストを取得する
        if (overlap) {
            continue;
        }
        // カテゴリ情報を追加する
        mCategoryInfoList.add(values);
    }
    cursor.close();
}
```

サンプルコード：URI 一覧を取得した場合

```
// 入力フィールドに設定されている対象絵文字タイプ（“emojiType”）を取得
int emojiType = bundle.getInt(“emojiType”);
// 絵文字タイプを SQL 条件文字列に変換
String kind = DecoEmojiContract.makeStringEmojiKind(emojiType);
// カラムを作成
String projection [] = {DecoEmojiContract.DecoEmojiInfoColumns.URI , DecoEmojiContract.DecoEmojiInfoColumns.KIND };
// 条件句を作成
// 条件として絵文字種別の設定を行う
String selection = “ (” + DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_ID + “ = ?) and ( ”
                + DecoEmojiContract.DecoEmojiInfoColumns.KIND + “ in( ” + kind + “ ) ) ”;
// order 句を作成
// 表示順で取得できるようにソート条件を設定する
String order = DecoEmojiContract.DecoEmojiInfoColumns.LAST_USE_CNT + “ desc” ;
String arg[] = {“”};
arg[0]= String.valueOf(categoryInfo.getCategoryId()); // カテゴリ取得
// URI 一覧取得
Cursor cursor = mLocalContext.getContentResolver()
                .query(DecoEmojiContract.CONTENT_DECOINFOLIST_URI, projection, selection, arg, order);

if (cursor != null) {
    while (cursor.moveToNext()) {
        // デコメピクチャの判定を行い、デコメピクチャは最後に一括で追加するようにする
        if (cursor.getInt( cursor.getColumnIndex (DecoEmojiContract.DecoEmojiInfoColumns.KIND)) !=
DecoEmojiContract.KIND_PICTURE ) {

mCurrentDecoEmojiUriList.add(cursor.getString( cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.URI)) );
        } else {
            tmpDecoEmojiUriList.add(cursor.getString( cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.URI)));
        }
    }
    mCurrentDecoEmojiUriList.addAll(tmpDecoEmojiUriList);
    cursor.close();
}
```

14.7.6.2. デコメ絵文字情報件数取得I F

インタフェース名	デコメ絵文字情報件数取得			
取得モード	decoinfocount			
説明	デコメ絵文字情報テーブルから、検索条件に沿った件数を取得する。			
引数				
No	引数名	型	Must/Opt.	説明
1	Uri	Uri	Must	DecoEmojiContract. CONTENT_DECOINFOLIST_COUNT_URI
2	projection	String[]	-	取得カラム … 取得するカラム名 取得カラムの指定は不要
3	selection	String	Opt	フィルタ … フィルタ(Where条件) ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*2)取得条件参照
4	selectionArgs	String[]	Opt	クエリパラメータ … クエリパラメータ ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*2)取得条件参照
5	sortOrder	String	-	ソート順 … ソート順 ソート順の指定は不要
戻り値				
No	型	説明		
1	Cursor	デコメ絵文字情報件数テーブルへのカーソル		
(*1)取得項目				
No	項目名	項目 ID	型	備考
1	デコメ絵文字情報件数	decoemoji_info_cnt	INTEGER	
(*2)取得条件				
No	項目名	項目 ID	型	備考
1	デコメ絵文字ID	decoemoji_id	INTEGER	
2	デコメ絵文字URI	uri	TEXT	
3	カテゴリID	category_id	INTEGER	
4	カテゴリ名（日本語）	category_name_jpn	TEXT	
5	カテゴリ名（英語）	category_name_eng	TEXT	
6	カテゴリプリセットID	category_preset_id	INTEGER	
7	ディレクトリID	directory_id	INTEGER	
8	ディレクトリ名	directory_name	TEXT	
9	更新タイムスタンプ	timestamp	INTEGER	
10	ファイルサイズ	file_size	INTEGER	
11	拡張タグ名	tags	TEXT	
12	幅	width	INTEGER	
13	高さ	height	INTEGER	
14	絵文字種別	kind	INTEGER	1：「デコメ絵文字」 2～4：「デコメピクチャ」
15	最終利用表示順	last_use_cnt	INTEGER	
16	利用履歴表示順	history_cnt	INTEGER	
17	デコメ絵文字pop識別フラグ	decome_pop_flag	INTEGER	0：デコメ絵文字pop識別フラグなし 1：デコメ絵文字pop識別フラグあり
(*3)ソート順				
ソート順の指定は不要				

サンプルコード(詳細取得方式) :

```
String selection = DecoEmojiContract.DecoEmojiInfoColumns.DIRECTORY_NAME + " like ?";
String arg[] = {"%a%"};
Cursor cur = resolver.query(DecoEmojiContract.CONTENT_DECOINFOLIST_COUNT_URI, null, selection, arg,
null);
```

14.7.6.3. デコメ絵文字辞書一覧取得I F

インタフェース名	デコメ絵文字辞書一覧取得
取得モード	decodiclist
説明	デコメ絵文字辞書テーブルから、検索条件に沿ったデコメ絵文字情報データ一覧を取得する。

引数

No	引数名	型	Must/Opt.	説明
1	Uri	Uri	Must	・詳細取得方式 DecoEmojiContract.CONTENT_DECODICLIST_URI ・簡易取得方式 DecoEmojiContract.CONTENT_DECODICLIST_URI/[デコメ絵文字ID] ※デコメ絵文字IDを指定して取得する場合は[デコメ絵文字ID]にデコメ絵文字IDを指定します。
2	projection	String[]	Opt	取得カラム … 取得するカラム名 ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*1)取得項目参照
3	selection	String	Opt	フィルタ … フィルタ(Where条件) ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*2)取得条件参照
4	selectionArgs	String[]	Opt	クエリパラメータ … クエリパラメータ ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*2)取得条件参照
5	sortOrder	String	Opt	ソート順 … ソート順 ・省略する場合はNULLを指定する。その場合、昇順で取得する。 ・(*3)ソート順参照

戻り値

No	型	説明
1	Cursor	デコメ絵文字辞書テーブルへのカーソル

(*1)取得項目

No	項目名	項目 ID	型	備考
1	デコメ絵文字ID	decoemoji_id	INTEGER	
2	デコメ絵文字URI	uri	TEXT	
3	読み	decoemoji_name	TEXT	
4	品詞	decoemoji_part	INTEGER	
5	表記	decoemoji_note	TEXT	
6	更新タイムスタンプ	timestamp	INTEGER	

(*2)取得条件

No	項目名	項目 ID	型	備考
1	デコメ絵文字ID	decoemoji_id	INTEGER	
2	デコメ絵文字URI	uri	TEXT	
3	読み	decoemoji_name	TEXT	
4	品詞	decoemoji_part	INTEGER	
5	表記	decoemoji_note	TEXT	
6	更新タイムスタンプ	timestamp	INTEGER	

(*3)ソート順

No	項目名	項目 ID	昇順(ASC)/降順(DESC)	初期値
1	デコメ絵文字ID	decoemoji_id	昇順・降順	昇順
2	デコメ絵文字URI	uri	昇順・降順	-
3	読み	decoemoji_name	昇順・降順	-
4	品詞	decoemoji_part	昇順・降順	-

5	表記	decoemoji_note	昇順・降順	-
6	更新タイムスタンプ	timestamp	昇順・降順	-

サンプルコード(詳細取得方式):

```
String column[] = {DecoEmojiContract.DecoEmojiDicColumns.DECOEMOJI_NAME};
String selection = DecoEmojiContract.DecoEmojiDicColumns.DECOEMOJI_NOTE + " like ?";
String arg[] = {"%-%"};
String order = DecoEmojiContract.DecoEmojiDicColumns.DECOEMOJI_ID + " DESC";
Cursor cur = resolver.query(DecoEmojiContract.CONTENT_DECODICLIST_URI, column, selection, arg, order);
```

サンプルコード(簡易取得方式):デコメ絵文字 ID = 1234 を取得したい場合

```
Cursor cur = resolver.query(DecoEmojiContract.CONTENT_DECODICLIST_URI/1234, null, null, null, null);
```

14.7.6.4. デコメ絵文字辞書件数取得I F

インタフェース名	デコメ絵文字辞書件数取得			
取得モード	decodiccount			
説明	デコメ絵文字情報テーブルから、検索条件に沿った件数を取得する。			
引数				
No	引数名	型	Must/Opt.	説明
1	Uri	Uri	Must	DecoEmojiContract. CONTENT_DECODICLIST_COUNT_URI
2	projection	String[]	-	取得カラム … 取得するカラム名 取得カラムの指定は不要
3	selection	String	Opt	フィルタ … フィルタ(Where条件) ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*2)取得条件参照
4	selectionArgs	String[]	Opt	クエリパラメータ … クエリパラメータ ・省略する場合はNULLを指定する。その場合、全ての項目を取得する。 ・(*2)取得条件参照
5	sortOrder	String	-	ソート順 … ソート順 ソート順の指定は不要
戻り値				
No	型	説明		
1	Cursor	デコメ絵文字辞書テーブルへのカーソル		
(*1)取得項目				
No	項目名	項目 ID	型	備考
1	件数	decoemoji_info_cnt	INTEGER	
(*2)取得条件				
No	項目名	項目 ID	型	備考
1	デコメ絵文字ID	decoemoji_id	INTEGER	
2	デコメ絵文字URI	uri	TEXT	
3	読み	decoemoji_name	TEXT	
4	品詞	decoemoji_part	INTEGER	
5	表記	decoemoji_note	TEXT	
6	更新タイムスタンプ	timestamp	INTEGER	
(*3)ソート順				
ソート順の指定は不要				

サンプルコード(詳細取得方式)：

```
String selection = DecoEmojiContract.DecoEmojiDicColumns.DECOEMOJI_NAME + " like ?";
String arg[] = {"%a%"};
Cursor cur = resolver.query(DecoEmojiContract.CONTENT_DECODICLIST_COUNT_URI, null, selection, arg, null);
```

14.7.7. 利用クラス（DecoEmojiContractクラス）

パッケージ名:jp.co.omronsoft.android.decoemojimanager_<career>.interfacedata

14.7.7.1. 「対象絵文字タイプ」 To 「絵文字種別」 変換

API 名		makeStringEmojiKind			
説明		対象絵文字タイプ（“emojiType”）の値を絵文字種別の値に変換する。			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	emojiType	int	I	Must	表示対象の対象絵文字タイプ（“emojiType”）を指定する (0x0000 <emojiType <= 0x000F)
2					
返値					
No	型	説明			
1	String	絵文字種別を表す値			
前提 なし					
補足 現在、本機能の利用は非推奨です。					

14.7.7.2. 「絵文字種別」 To 「対象絵文字タイプ」 変換

API 名		convertEmojiType			
説明		絵文字種別の値を対象絵文字タイプ（“emojiType”）の値に変換する。（入力フィールドの対象絵文字タイプ（“emojiType”）と、取得結果の絵文字種別を比較する際などに使用する）			
引数					
No	引数名	型	I/O	Must/Opt.	説明
1	emojiKind	int	I	Must	
2					
返値					
No	型	説明			
1	int	絵文字種別を表す値			
前提					
なし					
概略説明					
- 絵文字種別の値を対象絵文字タイプ（“emojiType”）の値に変換する。（入力フィールドの対象絵文字タイプ（“emojiType”）と、取得結果の絵文字種別を比較する際などに使用する） - 返却値は以下の通り。 emojiKind：1 返却値：0x0001 emojiKind：2 返却値：0x0002 emojiKind：3 返却値：0x0004 emojiKind：4 返却値：0x0008 - emojiKind が 0 以下、もしくは 5 以上の値が指定された場合には、0 を返却する。					

サンプルコード:

```
Int kind = 4;
int emojiType = bundle.getInt( "emojiType" );
int convEmojiType = DecoEmojiContract.convertEmojiType(kind);
if ((convEmojiType & emojiType) != 0) {
    // 以降は任意の処理
}
```

14.7.8. 定数

14.7.8.1. ContentProviderURI

No	定数名	型	定数内容
1	DecoEmojiContract.CONTENT_DECOINFOLIST_URI	Uri	デコメ絵文字情報一覧 URI
2	DecoEmojiContract.CONTENT_DECOINFO_COUNT_URI	Uri	デコメ絵文字情報件数 URI
3	DecoEmojiContract.CONTENT_DECODICLIST_URI	Uri	デコメ絵文字辞書一覧 URI
4	DecoEmojiContract.CONTENT_DECODIC_COUNT_URI	Uri	デコメ絵文字辞書件数 URI

14.7.8.2. デコメ絵文字情報テーブルカラム定数定義

「デコメ絵文字情報テーブル」を(A)と表現する。

No	定数名	型	定数内容
1	DecoEmojiContract.DecoEmojiInfoColumns.DECOEMOJI_ID	String	(A).デコメ絵文字 ID 項目名
2	DecoEmojiContract.DecoEmojiInfoColumns.URI	String	(A).デコメ絵文字 URI 項目名
3	DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_ID	String	(A).カテゴリ ID 項目名
4	DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_NAME_JPN	String	(A).カテゴリ名 (日本語) 項目名
5	DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_NAME_ENG	String	(A).カテゴリ名 (英語) 項目名
6	DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_PRESET_ID	String	(A).カテゴリプリセット ID 項目名
7	DecoEmojiContract.DecoEmojiInfoColumns.DIRECTORY_ID	String	(A).ディレクトリ ID 項目名
8	DecoEmojiContract.DecoEmojiInfoColumns.DIRECTORY_NAME	String	(A).ディレクトリ名 項目名
9	DecoEmojiContract.DecoEmojiInfoColumns.TIMESTAMP	String	(A).更新タイムスタンプ 項目名
10	DecoEmojiContract.DecoEmojiInfoColumns.FILE_SIZE	String	(A).ファイルサイズ 項目名
11	DecoEmojiContract.DecoEmojiInfoColumns.TAGS	String	(A).拡張タグ名 項目名
12	DecoEmojiContract.DecoEmojiInfoColumns.DECOEMOJI_INFO_COUNT	String	(A).カウント 項目名
13	DecoEmojiContract.DecoEmojiInfoColumns.WIDTH	String	(A).幅 項目名
14	DecoEmojiContract.DecoEmojiInfoColumns.HEIGHT	String	(A).高さ 項目名
15	DecoEmojiContract.DecoEmojiInfoColumns.KIND	String	(A).種別 項目名
16	DecoEmojiContract.DecoEmojiInfoColumns.LAST_USE_CNT	String	(A).最終利用表示順 項目名
17	DecoEmojiContract.DecoEmojiInfoColumns.HISTORY_CNT	String	(A).利用履歴表示順 項目名
18	DecoEmojiContract.DecoEmojiInfoColumns.DECOME_POP_FLAG	String	(A). デコメ絵文字 pop 識別フラグ 項目名

14.7.8.3. デコメ絵文字辞書テーブルカラム定数定義

「デコメ絵文字辞書テーブル」を(B)と表現する。

No	定数名	型	定数内容
1	DecoEmojiContract.DecoEmojiDicColumns.DECOEMOJI_ID	String	(B).デコメ絵文字 ID 項目名

2	DecoEmojiContract.DecoEmojiDicColumns.URI	String	(B).デコ絵文字 URI 項目名
3	DecoEmojiContract.DecoEmojiDicColumns.DECOEM OJI_NAME	String	(B).読み項目名
4	DecoEmojiContract.DecoEmojiDicColumns.DECOEM OJI_PART	String	(B).品詞項目名
5	DecoEmojiContract.DecoEmojiDicColumns.DECOEM OJI_NOTE	String	(B).表記項目名
6	DecoEmojiContract.DecoEmojiDicColumns.TIMESTA MP	String	(B).更新タイムスタンプ項目名
7	DecoEmojiContract.DecoEmojiDicColumns.DECOEM OJI_DIC_COUNT	String	(B).カウント項目名

14.7.9. 例外

- (1) query の ContentURI が正しくない場合
例外(IllegalStateException) を throw する。
- (2) データベース処理(SQLite)で例外が発生した場合
SQLite への query 処理で例外が発生した場合、SQLite の例外を throw する。
例外については、Android の SQLite の仕様に準拠する。

15. Data Format

15.1. デコメ絵文字フォーマット

デコメ絵文字に関するフォーマットは以下の通り。

15.1.1. 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)

15.1.2. デコメ絵文字popファイルの判定方法

デコメ絵文字 pop ファイルの判定方法は、5.1.1.GIF ファイルフォーマットに記載している Block1(Application Extension Block)内の Application Identifier を参照して、「HTMLEMJI」が含まれていればデコメ絵文字 pop と判定する。

Block 名		バイト数	内容
Application Extension Block	Extension Introducer	1 byte	0x21
	Extension Label	1 byte	0xFF
	Block Size #1	1 byte	0x0B
	Application Identifier	8 bytes	0x48 0x54 0x4D 0x4C 0x45 0x4D 0x4A 0x49 (HTMLEMJI)
	Application Authentication Code	3 bytes	0x31 0x2E 0x30 (1.0)
	Block Size #2	1 byte	Application Data のデータサイズ (例 : 255 bytes の場合は 0xFF)
	Application Data	n bytes	素材データ
	Block Terminator	1 byte	0x00

15.1.3. 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)

15.1.4. 付加情報定義フォーマット

付加情報定義フォーマットは以下の通り。

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

付加情報の格納例

```
<emoji><label><name>えがお</name><note>笑顔</note></label>
<label><name>わらう</name><part>動詞</part></label></emoji>
```

付加情報の格納例（カスタムタグに<url>、<melody>を定義した場合）

```
<emoji><label><name>えがお</name><note>笑顔</note></label>
<label><name>わらう</name><part>動詞</part></label>
<tags name="url"> http://www.xxx.com</tags>
<tags name="melody">/mnt/sdcard/aaa.ogg</tags>
</emoji>
```


15.1.5. 付加情報の詳細

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 (0-1)	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。 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="カスタム"></tags> カスタムタグは自由に定義できるため、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。 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、1 種類のカスタムタグ名に対して、最大 10 個の要素を記述できる。

15.1.6. 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 >
] >

```

15.1.7. 付加情報文字列の上限

上限は 10240bytes である。

15.1.8. 付加情報文字列の文字コード

文字コードは UTF-8 である。

15.1.9. 付加情報文字列の分割

付加情報の 1 つのタグ文字列は、256 bytes 以内とする。

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。

Block の先頭が<emoji>で始まらない場合、その Block はデコメ絵文字読み情報と見なさない。

15.1.10. 日本語向け品詞一覧

品詞_ID	品詞*	説明
1	名詞	普通名詞
2	名詞-固有名詞	固有名詞
3	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」
4	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」…
5	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」…
6	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」…
7	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」…
8	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」…
1 1	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」「c m」「k g」「カ月」「時間」「時半」…
1 2	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）」
1 4	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）」
1 5	動詞	動詞
1 6	形容詞	形容詞
1 7	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
1 8	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」…
1 9	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」…
2 0	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」…
2 1	記号	一般的な記号. 例：「○」「◎」「@」「\$」「〒」「→」「+」など

*オープンな品詞体系「IPADIC」を参考に抽出したもの。

改定履歴		
V1.0.0	2011/10/17	新規作成
V1.1.0	2012/3/27	デコ絵文字更新通知機能追加 端末スリープ時のアニメーション停止の注意を追加 API 仕様を追加（インタフェース仕様書の参照を不要にした。）
V1.1.1	2012/4/10	「デコメ絵文字読み付加情報フォーマット」に<label>の出現数上限を追記
V1.2.0	2012/9/14	絵文字パッケージ V1.2 対応 絵文字マネージャ ContentProvider 機能追加 デコメ絵文字画像の高精細化 デコメピクチャの追加
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