

絵文字パッケージ対応 IME 開発ガイド

Ver. 3.0.0

オムロン ソフトウェア株式会社
2020/12/07

目次

| | |
|---|----|
| 絵文字パッケージ対応IME開発ガイド | 1 |
| Ver. 3.0.0 | 1 |
| 1. 本ドキュメントについて | 5 |
| 2. 略語・記号の定義 | 5 |
| 3. 概要 | 6 |
| 3.1. 絵文字パッケージでできること | 6 |
| 3.2. 全体構成 | 7 |
| 3.2.1. デコメ絵文字表示機能 | 7 |
| 3.2.2. デコメ絵文字予測/変換入力機能 | 7 |
| 3.2.3. デコメ絵文字管理機能 | 7 |
| 3.3. デコメ絵文字の種別 | 8 |
| 3.3.1. デコメ絵文字種別の定義 | 8 |
| 3.3.2. 入力対象絵文字タイプ(“emojiType”)の詳細 | 9 |
| 4. 絵文字パッケージ連携機能実装・利用手順 (IMEV向け) | 10 |
| 4.1. デコメ絵文字入力機能の実装 | 10 |
| 4.2. 絵文字表示対応 | 11 |
| 4.2.1. 絵文字表示可能な表示フィールドの生成 | 11 |
| 4.2.2. デコメ絵文字コードの生成 | 11 |
| 4.2.3. アニメーション時の注意事項 | 12 |
| 4.2.4. Androidバージョンによる絵文字表示ライブラリの使用注意事項 | 14 |
| 4.3. IMEが受信すべき更新通知 (Broadcast) | 15 |
| 4.4. 絵文字マネージャへのアクセス | 15 |
| 4.5. 利用履歴情報の同期について | 17 |
| 5. 制限事項 | 18 |
| 5.1. ハードウェアアクセラレーションに関する問題について | 18 |
| 5.2. SDカード参照について | 18 |
| 5.3. RuntimePermission機能について | 18 |
| 5.4. targetSdkVersion 26 での動作について | 18 |
| 5.5. AndroidバージョンP以降の端末動作について | 19 |

| | |
|---|-----------|
| 6. 絵文字パッケージのインタフェース | 20 |
| 6.1. インタフェースについて | 20 |
| 6.2. 絵文字表示・入力動作シーケンス | 21 |
| 6.3. 入力フィールド(EditText)の絵文字入力の有効化 | 22 |
| 6.3.1. TextView の絵文字有効化 | 22 |
| 6.3.2. デコメ絵文字対象絵文字タイプ("emojiType")の定義 | 22 |
| 6.3.3. デコメ絵文字対象絵文字タイプ("emojiType")の詳細 | 23 |
| 7. <i>[API]</i> インタフェース仕様 | 24 |
| 7.1. API一覧 | 24 |
| 7.2. EmojiAssist クラス | 26 |
| 7.2.1. インスタンス取得 | 26 |
| 7.2.2. TextView 登録 | 27 |
| 7.2.3. TextView 削除 | 28 |
| 7.2.4. TextView 全削除 | 29 |
| 7.2.5. アニメーション開始 | 30 |
| 7.2.6. アニメーション停止 | 31 |
| 7.2.7. デコメ絵文字 Span 設定 | 32 |
| 7.2.8. 入力文字列の絵文字有無判定 | 35 |
| 7.2.9. デコメ絵文字 pop 識別子表示 | 36 |
| 7.2.10. デコメピクチャ拡張表示設定 | 37 |
| 7.2.11. TextView 登録(拡張版) | 38 |
| 7.3. EmojiAssist.DecoEmojiTextInfo クラス | 39 |
| 7.3.1. URI 設定 | 39 |
| 7.3.2. 対象絵文字タイプ設定 | 40 |
| 7.3.3. 幅設定 | 41 |
| 7.3.4. 高さ設定 | 42 |
| 7.3.5. 絵文字種別設定 | 43 |
| 7.3.6. pop 識別子設定 | 44 |
| 7.3.7. カスタムタグ有無情報設定 | 45 |
| 7.4. DecoEmojiSpan クラス | 46 |
| 7.4.1. URI 取得 | 46 |
| 7.4.2. 幅取得 | 47 |
| 7.4.3. 高さ取得 | 48 |
| 7.4.4. デコメ絵文字 pop 識別子取得 | 49 |
| 7.4.5. カスタムタグ情報取得 | 50 |
| 7.4.6. デコメピクチャ判定 | 51 |
| 8. 絵文字マネージャ | 52 |
| 8.1. 絵文字マネージャ 要求 (Broadcast) インタフェース | 53 |
| 8.1.1. ファイル差分チェック実施要求 | 53 |
| 8.1.2. 絵文字マネージャ通知アプリ登録要求 | 53 |

| | |
|---|-----------|
| 8.2. 絵文字マネージャ 通知 (Broadcast) インタフェース | 55 |
| 8.2.1. デコメ絵文字情報更新通知 | 56 |
| 8.2.2. デコメ絵文字辞書更新通知 | 56 |
| 8.2.3. 絵文字マネージャ状態変更通知 | 57 |
| 8.2.4. ファイル差分チェック要求応答通知 | 57 |
| 8.2.5. 補助クラス | 58 |
| 8.2.6. 定数 | 59 |
| 8.3. 絵文字マネージャContentProviderインタフェース | 60 |
| 8.3.1. 絵文字パッケージで提供する ContentProvider 機能 | 60 |
| 8.3.2. 前提条件 | 60 |
| 8.3.3. 取得モード一覧および取得データ一覧 | 60 |
| 8.3.4. 使用方法 | 60 |
| 8.3.5. ContentProvider利用時のqueryパラメータについて | 61 |
| 8.3.6. 取得データ詳細 | 62 |
| 8.3.7. 取得モード詳細 | 65 |
| 8.3.8. 補助クラス (DecoEmojiContractクラス) | 72 |
| 8.3.9. 定数 | 74 |
| 8.3.10. データ取得時の推奨事項について | 76 |
| 8.3.11. 例外 | 76 |
| 8.4. 絵文字マネージャ サービス(aidl)インタフェース | 77 |
| 8.4.1. 絵文字マネージャ サービス (aidl) インタフェースの接続と切断 | 77 |
| 8.4.2. サービス(aidl)インタフェースのAPI | 78 |
| 8.4.3. 定数 | 81 |
| 9. Data Format | 82 |
| 9.1. デコメ絵文字フォーマット | 82 |
| 9.1.1. GIF ファイルフォーマット | 82 |
| 9.1.2. JPEG ファイルフォーマット | 82 |
| 9.1.3. デコメ絵文字popファイルの判定方法 | 83 |
| 9.1.4. 付加情報定義フォーマット | 84 |
| 9.1.5. 付加情報の詳細 | 85 |
| 9.1.6. 付加情報フォーマットDTD(参考) | 86 |
| 9.1.7. 付加情報文字列の上限 | 86 |
| 9.1.8. 付加情報文字列の文字コード | 86 |
| 9.1.9. 付加情報文字列の分割 | 86 |
| 9.1.10. 日本語向け品詞一覧 | 87 |
| 9.2. DBエンティティ情報 | 88 |
| 9.2.1. デコメ絵文字情報テーブル[decoemoji_info] | 88 |
| 9.2.2. デコメ絵文字辞書テーブル[decoemoji_dic] | 90 |
| 9.2.3. カテゴリ情報テーブル[category_info] | 91 |
| 9.2.4. ディレクトリ情報テーブル[directory_info] | 92 |
| 9.2.5. カスタムタグ情報テーブル[customtag_info] | 92 |
| 10. 商標について | 93 |

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、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. 略語・記号の定義

| 用語 | 意味 |
|----------------|--|
| IMF | Input Method Framework |
| IME | Input Method Editor |
| IMEV | Input Method Editor Vender |
| HSV | Handset Vender |
| APPV | Application Vender |
| 表示フィールド | アプリケーションの用いる文字・画像の表示フィールド(View) |
| 入力フィールド | アプリケーションの用いる文字入力フィールド(View) |
| Unicode 絵文字 | Unicode として標準化された文字コードを持つ絵文字。 |
| デコメ絵文字 | デコレーション絵文字とデコメピクチャの総称。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。デコメコンテンツと呼ぶ場合もある。 |
| デコレーション絵文字 | 文章中に文字と同様に用いられる画像。 |
| 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。 |
| Image Keyboard | 画像やその他のコンテンツを IME からアプリケーションの入力フィールドに直接入力できる機能。API レベル 25 およびリビジョン 25.0.0 の v13 サポートライブラリで追加された。 |

※「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。

3. 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(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。

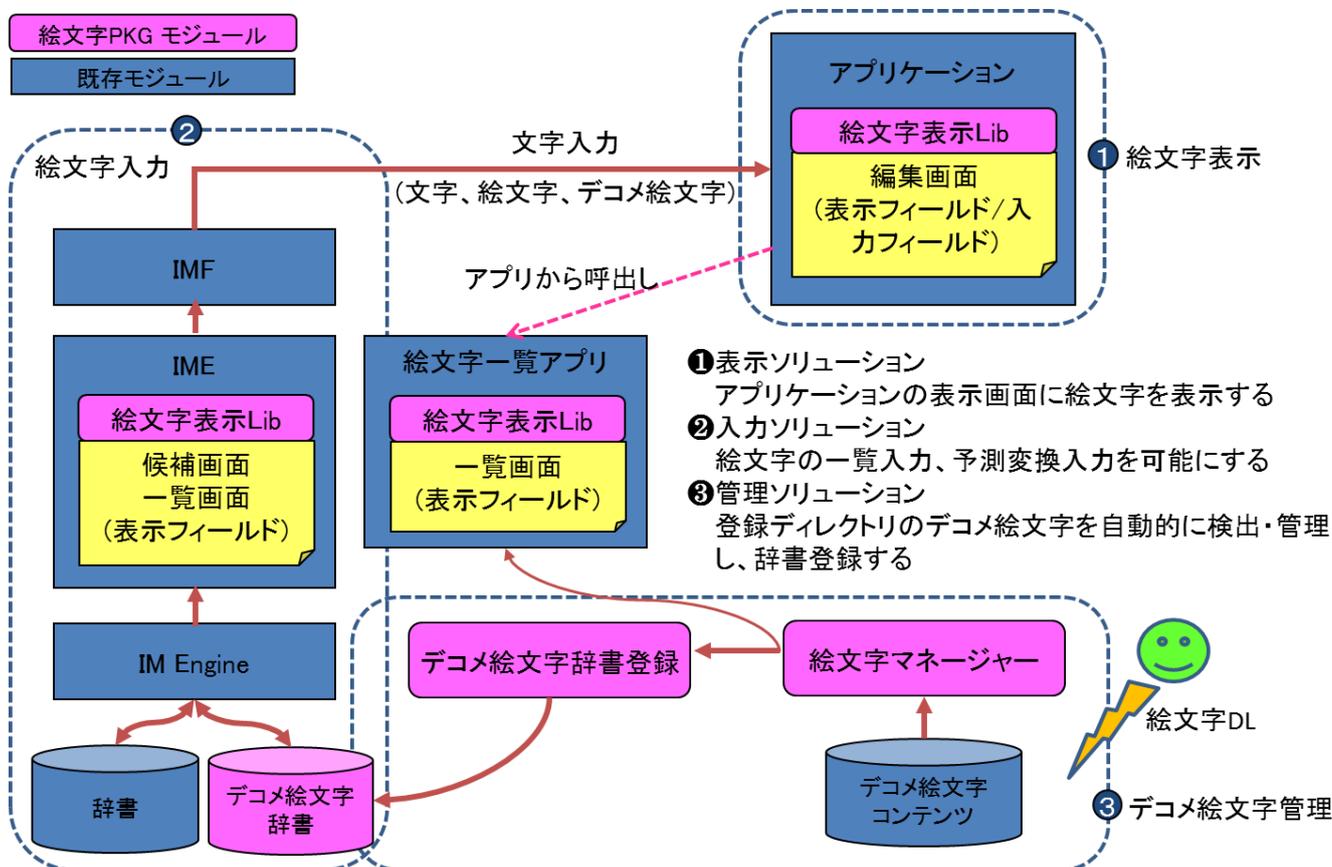
※株式会社 NTT ドコモは 2020-2021 冬春モデル以降、絵文字パッケージの端末プリインストールを廃止します。そのため、該当する機種においてはデコメ絵文字が使用不可となります。

3.1. 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。
 - Android 端末に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デコメ絵文字を利用することができます。
 - 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後からダウンロードしたデコメ絵文字を一覧表示し、アプリから利用することを可能にします。
 - 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。新しくダウンロードしたデコメ絵文字でも、一覧パレットや予測・変換候補に表示されることで、ユーザに気づきの機会を与えることができます。絵文字パッケージが備えるすべての機能を Android 端末で利用可能にするには、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。

3.2. 全体構成

構成は下記の通りとなります。



3.2.1. デコメ絵文字表示機能

絵文字パッケージの提供する「絵文字表示ライブラリ」(Emoji Assist)をアプリに追加することによって、アプリの入力・表示フィールドにおいて通常の文字に交じってデコメ絵文字の入力・表示を実現できます。

3.2.2. デコメ絵文字予測/変換入力機能

IMEの絵文字一覧パレット、アプリによる絵文字一覧表示を用いて、通常文字に交じってデコメ絵文字を入力することができます。

また、デコメ絵文字一覧入力に加え、通常文字と同じようにIMEによるデコメ絵文字予測/変換入力を実現できます。なお、デコメ絵文字の予測変換入力を実現するには、デコメ絵文字予測/変換入力機能をIMEに追加する必要があります。

3.2.3. デコメ絵文字管理機能

絵文字パッケージは、サービスアプリ「絵文字マネージャ」によって、所定の場所に登録、またはダウンロードされたデコメ絵文字を自動認識し、アプリで共通に利用できる絵文字として管理します。デコメ絵文字に読みがあれば、IMEにその情報(辞書登録情報)を通知します。

3.3. デコメ絵文字の種別

3.3.1. デコメ絵文字種別の定義

絵文字パッケージでは、拡張したデコメ絵文字を以下の種別に定義します。

デコメ絵文字(デコメコンテンツ)

- ー デコレーション絵文字(正方形)
 - ー デコレーション絵文字(20*20ピクセル) ... ①
 - ー デコレーション絵文字(20*20ピクセルを除く正方形) ... ②
- ー デコレーション絵文字(長方形) ... ③
- ー デコメピクチャ ... ④

「デコレーション絵文字」と「デコメピクチャ」の違いは、デコレーション絵文字があくまで通常の文字と同様に、文字列の一部として通常文字と同サイズ(高さ)での表示が行われるのに対して、デコメピクチャは、元の画像サイズで表示が行われるという点にあります。

なお、既存アプリケーションの互換性が保てるように、より詳細な種別も定義し、アプリから指定ができるようにしました。

- ・ 従来のデコレーション絵文字だけが扱えるアプリのために、「デコレーション絵文字(20*20ピクセル)」を指定することができます。
- ・ 20*20ピクセルには限らないが、通常文字と同様の正方形の絵文字だけを、「デコレーション絵文字(正方形)」として指定することができます。(「ミニデコ」を除外する指定)

なお、①②③④は、**unique** な(他と重なりのない)基本種別の定義であり、これらを組み合わせて、特殊な組み合わせパターンを指定することも可能です。

デコメ絵文字の種別は、画像サイズと画像ファイルに格納される付加情報(<emoji>タグ)によって判定されます。

また、絵文字パッケージがデコレーション絵文字、デコメピクチャとして扱える画像のサイズには次の上限があります。

- ・ 画像サイズ : 960*960
- ・ ファイルサイズ : 2MBytes (2 * 1024 * 1024 Bytes)

※今後、変更の可能性あり。

GIF形式のデコメ絵文字には、デコメ絵文字 pop、デコメピクチャ pop という拡張情報(pop 情報)を持つものがあります。

デコメ絵文字/ピクチャ pop は、単独の GIF ファイルで可能なアニメーションの範囲を超えた、(広範囲、デコメ間の連動など)多様なアニメーションを行う機能を持つコンテンツです。

3.3.2. 入力対象絵文字タイプ("emojiType")の詳細

絵文字利用アプリは、入力フィールドごとに、入力可能な絵文字種別を、入力対象絵文字タイプとして指定することができます。対象絵文字タイプ("emojiType")は、表示対象となる絵文字種別を下記の通り組み合わせで定義します。

| No | 定義名 | 定義値 | 定義値 (数値) | 絵文字種別 |
|----|---|---|-------------|---------------------|
| 1 | EMOJITYPE_COMB_ANY (デコメ絵文字) | EMOJITYPE_PICTD_SQ20 +EMOJITYPE_PICTD_SQNOT20 +EMOJITYPE_PICTD_RECT +EMOJITYPE_PICTURE | 0x000F | 「①」、「②」、「③」、 「④」 |
| 2 | EMOJITYPE_COMB_PICTD (デコレーション絵文字) | EMOJITYPE_PICTD_SQ20 +EMOJITYPE_PICTD_SQNOT20 +EMOJITYPE_PICTD_RECT | 0x0007 | 「①」、「②」、「③」 |
| 3 | EMOJITYPE_COMB_SQ (デコレーション絵文字 (正方形)) | EMOJITYPE_PICTD_SQ20 +EMOJITYPE_PICTD_SQNOT20 | 0x0003 | 「①」、「②」 |
| 4 | EMOJITYPE_PICTURE (デコメピクチャ) | - | 0x0008 | 「④」 |
| 5 | EMOJITYPE_PICTD_RECT (デコレーション絵文字 (長方形)) | - | 0x0004 | 「③」 |
| 6 | EMOJITYPE_PICTD_SQNOT20 (デコレーション絵文字 (20*20ピクセルを除く 正方形)) | - | 0x0002 | 「②」 |
| 7 | EMOJITYPE_PICTD_SQ20 (デコレーション絵文字 (20*20ピクセル)) | - | 0x0001 | 「①」 |

通常は、No.1-4 の値を使用します。

No.5-7 は、すべての組み合わせパターンを網羅することができるよう、既存アプリの互換性のために用意した値であり、必要に応じて単独で使用することも、それぞれを組み合わせで特殊な絵文字種別を設定することも可能です。

例) `b.putInt("emojiType",EMOJITYPE_PICTD_SQ20|EMOJITYPE_PICTURE)` …①と④の組合せ

もし、対象絵文字タイプ("emojiType")に上記以外の値を指定された場合は、「①」のみを取得します。

4. 絵文字パッケージ連携機能実装・利用手順(IMEV 向け)

絵文字パッケージを利用するにあたっての IME 向けの事前準備および利用手順について記述しています。

4.1. デコメ絵文字入力機能の実装

IME は、デコメ絵文字の入力を可能にするため、絵文字パッケージと連携して、次の機能を実装する必要があります。

(1) デコメ絵文字一覧表示・入力機能

- (a) デコメ絵文字の入力が可能な入力フィールドであるとき、デコメ絵文字の一覧(パレット)を表示する。
- (b) 入力フィールドの絵文字タイプの指定に従って、一覧表示するデコメ絵文字のタイプを選ぶ。
- (c) デコメ絵文字はカテゴリ別に一覧表示する。
なお、カテゴリ内にデコレーション絵文字とデコメピクチャが存在する場合は、デコレーション絵文字とデコメピクチャはその順序で、それぞれをまとめて(群として)表示する。
- (d) デコメ絵文字の選択確定を受けて、文章中に入力する。
- (e) 利用履歴を管理し、表示する。
 - i. 利用履歴は、ユーザが再利用しやすいように、最も新しく利用されたものから順に、一定の数(たとえば最新利用の 20 個)を、常時、または先頭のカテゴリとして表示する。
 - ii. カテゴリ内では、最も新しく利用されたものから順に並べて表示する。デコレーション絵文字とデコメピクチャがある場合は、それぞれの群の中で利用順に表示する。
- (f) 利用履歴は IME その他入力アプリ間で共有する。

(2) デコメ絵文字予測・変換入力機能

- (a) デコメ絵文字の入力が可能な入力フィールドであるとき、デコメ絵文字を通常文字の候補とともに予測変換候補に表示する。
- (b) 入力フィールドの絵文字タイプの指定に従って、候補表示するデコメ絵文字のタイプを選ぶ。
- (c) デコメ絵文字の候補の採用確定を受けて、文章中に入力する。
- (d) 採用履歴を学習し、候補表示の優先度に反映する。

(3) デコメ絵文字出力機能

- (a) 入力文字列は、一覧入力または予測変換入力されたデコメ絵文字と通常文字を混在させた文字列として、アプリの入力フィールドに出力する。
- (b) 入力文字列は、属性情報を含むリッチテキスト形式の CharSequence 型で記述される。デコメ絵文字は、次の属性情報を持つ文字として記述される。
 - ・ デコメ絵文字 ID (URI)
 - ・ 画像サイズ(縦*横:pixel)
 - ・ 絵文字タイプ
※Image Keyboard 機能を利用して入力する場合"デコメピクチャ"固定
 - ・ pop 情報の有無
※Image Keyboard 機能を利用して入力する場合"無"固定
 - ・ カスタムタグの有無
※Image Keyboard 機能を利用して入力する場合"無"固定

(4) デコメ絵文字辞書更新機能

- (a) IME が初期起動されたとき、絵文字マネージャにすべてのデコメ絵文字の読み情報を問い合わせ、デコメ絵文

字辞書を作成する。

- (b) デコモ絵文字の辞書情報更新通知を受け付けて、デコモ絵文字辞書を更新する。
- (c) 絵文字マネージャの DB が構築または再構築されたとき、デコモ絵文字辞書を再構築する。

デコモ絵文字の入力方法は以下の 2 通りあります。入力フィールドの設定に合わせて入力方法を切り替えることも可能です。

(1) デコモ絵文字コードを生成して text として入力

「4.2.2 デコモ絵文字コードの生成」により、デコモ絵文字コードを生成し、文字列を入力します。

(2) Image Keyboard 機能を利用して画像として入力

入力フィールドの MIME タイプ設定のチェックや入力する画像を IME のプライベート領域にコピーする処理が必要です。

※Image Keyboard の詳細、実装方法は下記 URL を参照

[URL] <https://developer.android.com/guide/topics/text/image-keyboard.html>

4.2. 絵文字表示対応

IME が表示するデコモ絵文字一覧パレット、予測変換候補画面には、デコモ絵文字の表示が必要です。デコモ絵文字の表示が可能な表示フィールドを作るためには、以下の手続きが必要です。

4.2.1. 絵文字表示可能な表示フィールドの生成

絵文字を表示する TextView を生成し、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ライブラリであり、次の機能を持ちます。

- (1) 設定した TextView を定期的に更新(invalidate)し、アニメーション絵文字をアニメ動作させます。
- (2) 設定した TextView の文字列内容にデコモ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: DynamicDrawableSpan 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、デコモ絵文字の表示を行います。
- (3) デコモ絵文字 pop の識別子の表示可否を判断して、デコモ絵文字がデコモ絵文字 pop であるとき、識別子の表示を行います。

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);

// TextView を EmojiAssist に割り当てる
EmojiAssist ea = EmojiAssist.getInstance();

ea.setDisplayPop(true);      // デコモ絵文字 pop の識別子を表示させる

ea.addView(emojiView1);
ea.addView(emojiView2);

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

4.2.2. デコモ絵文字コードの生成

一覧入力、または予測変換入力された文字列にデコモ絵文字が含まれる場合、この文字列を IME からアプリケーションに入力文字列として引き渡す場合は、デコモ絵文字を所定の表現形式(デコモ絵文字コード)に変換する必要があります。このことによつて、デコモ絵文字は通常文字と区別なく文字列として入力され、アプリケーションの入力・表示フィールドではデコモ絵文字として

画像表示が行われます。

Image Keyboard 機能を利用して入力を行う場合は、デコメ絵文字コードに変換する必要はありません。アプリケーションの入力フィールドが `EmojiAssistEditText` であれば、自動で通常文字と区別のない文字列に変換し、画像表示が行われます。

4.2.2.1. デコメ絵文字の属性

デコメ絵文字は、次のような属性を持ちます。

- (1) URI
- (2) 横幅
- (3) 高さ
- (4) 絵文字種別
- (5) デコメ絵文字 pop の有無
- (6) カスタムタグの有無

これらの属性を表現するために、属性情報は、文字 (`CharSequence`) のスパン情報として付加されます。IME からアプリに渡される場合は `Annotation` スパンに記述され、アプリケーションの表示フィールド (`TextView`) では `DecoEmojiSpan` と呼ぶ絵文字パッケージの提供するスパンに転記されます。なお、文字コードには規定の文字コード(ゲタ文字"="(Unicode 0x3013))を用います。

4.2.2.2. デコメ絵文字コード生成 API

デコメ絵文字コードは以下の API を用いて生成します。

・`getDecoEmojiText(DecoEmojiTextInfo decoEmojiTextInfo)`

4.2.3. アニメーション時の注意事項

絵文字パッケージでは Android OMR1 以前用の `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。

※Android P 以降の `EmojiAssist (jar)` では、以下の処理を行う必要はありません。

IME がスリープ時に行うべき処理を以下に記載します。

4.2.3.1. スリープ時対応策 (サービスアプリケーション向け)

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();
            }
        }
    }
}
```

4.2.3.2. EmojiAssist への組込みを動的に実施する方法

IME では、予測変換候補画面のように、動的に生成された候補文字列を `TextView` の継承部品であるボタンに割り当てること
があります。この `TextView` 部品には、デコメ絵文字が含まれる場合も、含まれない場合もあります。これらのすべてを `EmojiAssist`
に組み込むと、アニメーション時の処理負荷が無駄に増えることが起こります。

このような場合は、`TextView` 部品にデコメ絵文字が含まれているかどうかを動的に判定し、含まれている場合のみ `EmojiAssist`
に組み込むことが効率的です。

このように、`EmojiAssist` への組込みを動的に判定して実施する場合は下記 API を利用することで対応が可能です。

サンプルコード：

```
//addView()した view の管理用 List
ArrayList<TextView> mAddViewList = new ArrayList<TextView>();
// EmojiAssist のインスタンス取得
EmojiAssist ea = EmojiAssist.getInstance();
// TextView を EmojiAssist に割り当てるかどうかを checkTextData() にてチェック
Spannable text = new SpannableString(textView.getText());
int ret = ea.checkTextData(text);
if (ret > 0) {
    ea.addView(textView);
    mAddViewList.add(textView);
}

...
中略
...

// デコメ絵文字が割り当てた View に存在するかのチェック
Boolean isDeco = false;
if (mAddViewList != null && mAddViewList.size() > 0) {
    for (int i = 0; mAddViewList.size() > i; i++) {
        Spannable text = new SpannableString(mAddViewList.get(i).getText());
        int ret = ea.checkTextData(text);
        //デコメ絵文字をアニメーションする場合
        if (ret > 1) {
            isDeco = true;
            break;
        }
    }
    if (isDeco) {
        ea.startAnimation();
    } else {
        ea.stopAnimation();
    }
} else {
    ea.stopAnimation();
}
```

4.2.4. Android バージョンによる絵文字表示ライブラリの使用注意事項

絵文字表示ライブラリ(emojisl_type2.jar/ emojisl_type2-OMR1.jar) について、AndroidP、OMR1 以前両方で動作するアプリ
を作成する場合は、Android P 以降用と OMR1 以前用の 2 ファイルを内包し、SDK バージョンの判定により使用する API を分岐
してください

```
if (Build.VERSION.SDK_INT > Build.VERSION_CODES.O_MR1) {  
    // emojis1_type2.jar の API を使用した処理  
} else {  
    // emojis1_type2-OMR1.jar の API を使用した処理  
}
```

絵文字表示ライブラリの API については、絵文字パッケージ インタフェース仕様書(javadoc)を参照ください。

4.3. IME が受信すべき更新通知(Broadcast)

IME が受信すべき更新通知の種別は以下の 3 つです。

(1) デコモ絵文字情報更新通知

- ・ デコモ絵文字情報を保持するアプリに対して、デコモ絵文字情報の更新を通知します。
- ・ 自アプリ内にデコモ絵文字の情報を保持するアプリは、この通知を受けて、自アプリ内の情報を更新しなければなりません。

(2) デコモ絵文字辞書更新通知

- ・ デコモ絵文字辞書情報を保持するアプリに対し、デコモ絵文字辞書情報の更新を通知します。
- ・ デコモ絵文字辞書情報を保持するアプリ(IME)は、この通知を受けて、自アプリ内の情報(デコモ絵文字辞書)を更新しなければなりません。

(3) 絵文字マネージャ状態変更通知

- ・ 絵文字マネージャのサービス状態変更を通知します。
- ・ 絵文字マネージャのサービス状態には、初期処理中(ファイルチェック中)のサービス不能状態とサービス可能状態があり、前者の場合はアプリ(IME)は絵文字マネージャのサービス(データ)を利用することができないので、初期処理(ファイルチェック処理)の終了が通知されるまで、処理を待たなければなりません。
- ・ 初期処理中であることは、絵文字マネージャにアクセスしたときに戻り値で知ることができます。また、それが終了した(状態が変化した)ことは、本通知で Broadcast されます。
- ・ 本通知によって、絵文字マネージャの持つ DB が初期化されたことが通知された場合は、デコモ絵文字情報を保持するアプリ(IME)、デコモ絵文字辞書情報を保持するアプリ(IME)は自アプリ内の情報(辞書)を初期化、更新しなくてはなりません。

(4) 実装上の注意点

- ・ デコモ絵文字情報更新通知、デコモ絵文字辞書情報更新通知については絵文字種別に関係なく通知が行われるため、IME 側で入力可能な絵文字種別を制限している場合は、通知されたデコモ絵文字の絵文字種別を確認し、入力可能とするかを判定するよう実装してください。

4.4. 絵文字マネージャへのアクセス

- (1) IME は、デコモ絵文字の一覧表示、デコモ絵文字辞書の登録・更新および候補ウィンドウへの表示のために、絵文字マネージャの DB にアクセスする必要があります。

- (2) 絵文字マネージャから取得すべき情報は以下の通りです。
- ・デコメ絵文字情報テーブル
 - ・デコメ絵文字情報件数テーブル
 - ・デコメ絵文字辞書テーブル
 - ・デコメ絵文字辞書件数テーブル
- (3) DB へのアクセスには ContentProvider 機能を利用します。取得方法については「 8.3 絵文字マネージャ ContentProvider 」を参照ください。

4.5. 利用履歴情報の同期について

デコメ絵文字の利用履歴については、絵文字パッケージの DB を参照し更新および取得する API を用いて、アプリケーション間で共有することができます。

絵文字パッケージは、各アプリがデコメ絵文字の利用履歴を、絵文字パッケージに通知し、同期された履歴情報を取得することのできる API を提供しています。

利用履歴情報の更新には、サービス (aidl) インタフェースの `aidl_changeHistory()` を利用し、取得には `Content Provider` インタフェースを利用ください。

5. 制限事項

5.1. ハードウェアアクセラレーションに関する問題について

絵文字パッケージでは、特定の Android バージョンにおいて、ハードウェアアクセラレーション が有効の場合には、EmojiAssist がデコメ絵文字を正しく描画できません。

[発生条件]

以下のいずれかの条件に該当する場合は対象となります。

1. 設定→開発者向けオプション→GPU レンダリングを使用 のチェックを ON に設定した場合。
2. アプリ側の「android:minSdkVersion」、または「android:targetSdkVersion」が「14」以上の場合。
3. アプリ側の「android:hardwareAccelerated="true"」の記述がある場合。

[原因]

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

[調査環境]

Android 4.0.4、4.2.2、5.0.0

[対策]

この問題の対策としては、アプリ側で以下の対応を行って頂く必要があります。

- ・ TextView に対して `setLayerType(View.LAYER_TYPE_SOFTWARE, null)` を設定。

5.2. SD カード参照について

以下のような場合に、SD カードのデコメコンテンツが表示できません。

- ・ SD カードを参照する権限が付与されていない場合

上記問題を回避するために、SD カード内のデコメコンテンツを使用するアプリケーションでは、AndroidManifest.xml 内に以下の記述を行う対応が必要です。

サンプルコード：

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

5.3. RuntimePermission 機能について

Android 6.0 以降については、デフォルトで SD カードを参照する権限が付与されません。

そのため SD カードのデコメコンテンツを使用するために、権限を有効にする対応の実装を推奨します。

5.4. targetSdkVersion 26 での動作について

- ・ 絵文字マネージャーからの通知 (Broadcast) 受信について

targetSdkVersion 26 以降のアプリでは、暗黙的な Broadcast を受信する動作が制限されるため、絵文字マネージャ

ーから Broadcast を送信する際に送信先のクラス名指定が必要になります。targetSdkVersion が 26 以降で絵文字マネージャーからの通知を受信するアプリは、以下のいずれかの設定が必要です。

- (ア) 絵文字管理追加設定ファイルに記載
- (イ) 絵文字マネージャー通知アプリ登録要求実施

・ファイル差分チェック要求について

targetSdkVersion 26 以降では、暗黙的 Broadcast の受信が制限されたため、「8.1絵文字マネージャー 要求 (Broadcast) インタフェース」利用時に、絵文字マネージャーのパッケージ名および BroadcastReceiver のクラス名の指定が必要です。指定方法は各要求のサンプルコードを参照ください。

5.5. Android バージョン P 以降の端末動作について

・描画について

動作端末 P 以降では、アプリ側で TextView に対して、ソフトウェアレンダリングの設定を行わないでください。ソフトウェアレンダリングの設定を行うと、EmojiAssist がデコメ絵文字を正しく描画できません。

```
if (Build.VERSION.SDK_INT > Build.VERSION_CODES.O_MR1) {  
    // emojis_type2.jar の API を使用した処理  
    //デコメ絵文字の描画をソフトウェアレンダリングに設定  
    TextViewに対して setLayerType(View.LAYER_TYPE_SOFTWARE, null)  
} else {  
    // emojis_type2-OMR1.jar の API を使用した処理  
    //デコメ絵文字の描画をソフトウェアレンダリングに設定  
    TextViewに対して setLayerType(View.LAYER_TYPE_SOFTWARE, null)  
}
```

6. 絵文字パッケージのインタフェース

6.1. インタフェースについて

絵文字パッケージでは、絵文字(デコメ絵文字)を利用するアプリケーション(IMEを含む)のために、次のインタフェースを提供します。

- (1) 入力フィールド(**EditText**)パラメータ
入力フィールド(**EditText**)において、絵文字の入力を可能にするためのパラメータの定義です。
- (2) **Image Keyboard** 対応入力フィールド(**EmojiAssistEditText**)
Image Keyboard 対応した **IME** からのコンテンツ入力可能な入力フィールドです。
- (3) 絵文字表示ライブラリ(**emojisl_type2.jar / emojisl_type2-OMR1.jar**)
表示フィールド(**TextView/EditText**)で絵文字の表示を可能にするためのライブラリです。
- (4) 絵文字マネージャインタフェースライブラリ(**DecoEmojiInterface.jar**)
絵文字の情報を管理するサービスアプリ「絵文字マネージャ」の各種機能を利用するためのインタフェースライブラリです。
- (5) 絵文字マネージャ **ContentProvider** インタフェース
絵文字の ID、付加情報(メタデータ)を格納するデータベースへのアクセスインタフェースです。
- (6) 絵文字マネージャサービス(**aidl**)インタフェース
絵文字マネージャが持つサービス機能を利用するためのアクセスインタフェースです。
- (7) 絵文字マネージャ通知(**Broadcast**)インタフェース
絵文字マネージャが、コンテンツの更新や、DB 状況の更新を通知するためのインタフェースです。

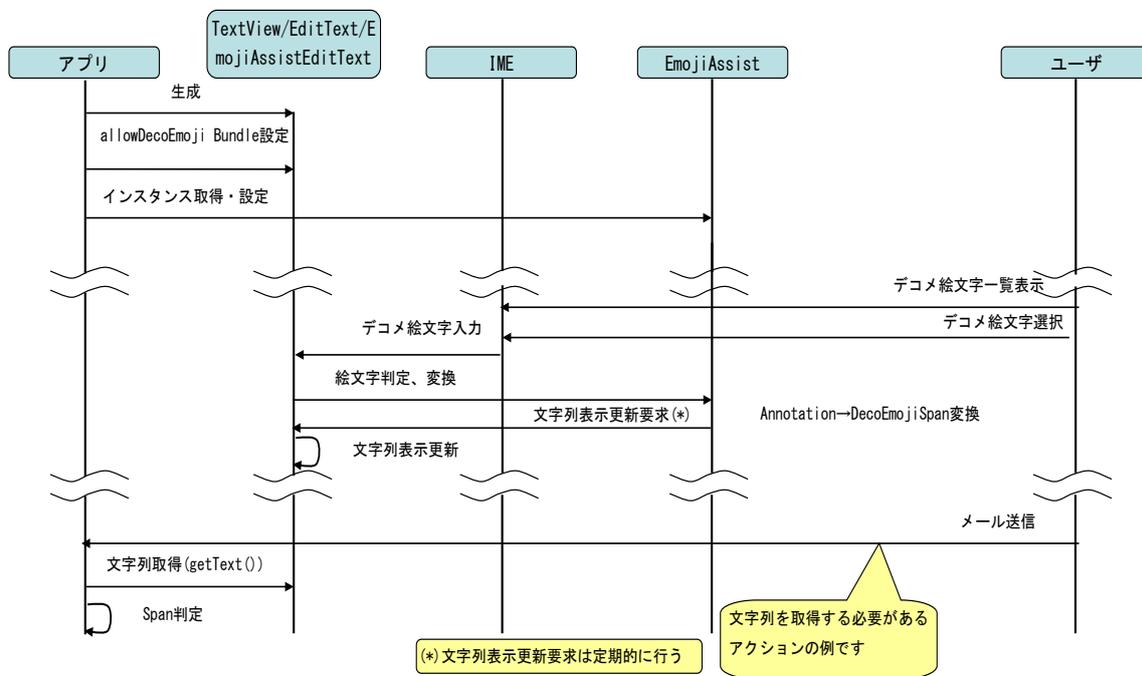
絵文字を入力および表示するアプリケーションは、(1)と(2)を利用します。

さらに、絵文字のより詳細な情報を利用するアプリケーションは、絵文字マネージャインタフェース(3)から(6)を利用します。

(2)絵文字表示ライブラリ(**emojisl_type2.jar / emojisl_type2-OMR1**)の API については「7. 【API】インタフェース仕様」を、
(3)絵文字マネージャインタフェースライブラリの API については、「8.絵文字マネージャ」を参照ください。

6.2. 絵文字表示・入力動作シーケンス

絵文字の入力と表示についてのシーケンス例は、次の通りです。



- (1) アプリは、絵文字表示/入力フィールド(TextView/EditText/EmojiAssistEditText)を生成し、そのインスタンスを絵文字表示用のライブラリであるEmojiAssistに組み込みます。
- (2) IMEでは絵文字一覧および予測変換候補に絵文字が表示され、ユーザが絵文字を選択・入力します。
- (3) 入力した結果はIME経由でEditText/EmojiAssistEditTextに通知されます。
- (4) EditTextでは文字列更新時のイベントハンドラとしてEmojiAssistを呼び出します。EmojiAssistは、デコメ絵文字情報を発見すると、TextView/EditText/EmojiAssistEditTextで表示可能な形式に変換します。
- (5) EmojiAssistはTextViewに対して文字列表示更新要求(invalidate)を送出します。定期的に送出することで、アニメーション表示を可能にします。
- (6) アプリは、編集された結果について変換処理が必要になった局面(メールの送信、文書の保存など)で、EditText/EmojiAssistEditTextからgetText()により文字列を取得します。その際、文字列にデコメ絵文字が含まれるかどうかを判定し、必要に応じてHTML形式への変換などの処理を実行します。

6.3. 入力フィールド(EditText)の絵文字入力の有効化

6.3.1. TextView の絵文字有効化

アプリで使用する TextView(EditText)で、デコメ絵文字を入力したい場合は、下記のようにデコメ絵文字有効フラグ (allowDecoEmoji) を Bundle する必要がある。

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

デコメ絵文字有効フラグ(allowDecoEmoji)が true の場合、対象絵文字タイプ("emojiType")を指定することができ、入力フィールドで指定された対象絵文字タイプ("emojiType")のみが IME に表示される。

```
b.putInt("emojiType", EMOJITYPE_COMB_ANY);
```

対象絵文字タイプ("emojiType")の指定を省略した場合には、「①デコメ絵文字(20*20)」を指定した場合と同じ取得結果となる。

EmojiAssistEditText で、コンテンツを入力したい場合は、デコメ絵文字有効フラグ、対象絵文字タイプの設定は不要です。入力されたコンテンツは全て「④デコメピクチャ」として扱われます。入力サービス(IME)からは、入力フィールドで指定された MIME タイプ("image/gif")を取得することで、コンテンツ入力可能か判定できます。

6.3.2. デコメ絵文字対象絵文字タイプ("emojiType")の定義

絵文字パッケージでは、拡張したデコメ絵文字を以下の種別に定義する。

デコメ絵文字(デコメコンテンツ)

- デコメ絵文字
 - デコメ絵文字(正方形)
 - デコメ絵文字(20*20ピクセル) … ①
 - デコメ絵文字(20*20ピクセルを除く正方形) … ②
 - デコメ絵文字(長方形) … ③
- デコメピクチャ … ④

下記の4つは、重複のない基本種別である。

「①デコメ絵文字(20*20)」

「②デコメ絵文字(正方形、20*20以外)」

「③デコメ絵文字(長方形、正方形を除く)」

「④デコメピクチャ」

6.3.3. デコメ絵文字対象絵文字タイプ(“emojiType”)の詳細

対象絵文字タイプ(“emojiType”)の詳細は、以下の様に定義する。

| No | 絵文字タイプ | 定義値 (シンボル) | 定義値 (数値) | 絵文字種別 |
|----|----------------------|-------------------------|-------------|-------|
| 1 | デコメコンテンツすべて | EMOJITYPE_COMB_ANY | 0x000F | ①②③④ |
| 2 | デコメ絵文字 | EMOJITYPE_COMB_PICTD | 0x0007 | ①②③ |
| 3 | デコメ絵文字 (正方形) | EMOJITYPE_COMB_SQ | 0x0003 | ①② |
| 4 | デコメピクチャ | EMOJITYPE_PICTURE | 0x0008 | ④ |
| 5 | デコメ絵文字 (長方形) | EMOJITYPE_PICTD_RECT | 0x0004 | ③ |
| 6 | デコメ絵文字 (20*20以外の正方形) | EMOJITYPE_PICTD_SQNOT20 | 0x0002 | ② |
| 7 | デコメ絵文字 (20*20) | EMOJITYPE_PICTD_SQ20 | 0x0001 | ① |

通常は、No.1-4 の値を使用する。

No.5-7 は、すべての組み合わせパターンを網羅することができるよう、既存アプリの互換性のために用意した値であり、必要に応じて単独で使用することも、それぞれを組み合わせる特殊な絵文字種別を設定することも可能である。

例)

`b.putInt(“emojiType”,EMOJITYPE_PICTD_SQ20|EMOJITYPE_PICTURE) ...①と④の組合せ`

もし、対象絵文字タイプ(“emojiType”)に上記以外の値を指定された場合は、「①」のみを取得する。

7. 【API】インタフェース仕様

7.1. API 一覧

絵文字パッケージ対応のアプリケーション(IME)は、以下の API を利用するために、次の JAR ライブラリを Link することが必要である。

- 絵文字描画ライブラリ (emojisl_type2.jar / emojisl_type2-OMR1.jar)
ライブラリに含まれるクラスは次の通り。
 - (A) EmojiAssist
 - (B) DecoEmojiSpan

| クラス名 | API 名称 | API 概要 |
|-------------|---|--------------------------------------|
| EmojiAssist | getInstance() | EmojiAssist のインスタンスを取得する。 |
| | addView(TextView view) | 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(DecoEmojiTextInfo decoEmojiTextInfo) | デコメ絵文字属性(Annotation)を設定した文字を取得する。 |
| | 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。 |
| | setPictureScale() | デコメピクチャに対し、デコメ絵文字と同様の拡大縮小表示を行う設定を行う。 |
| | addView(TextView view, boolean scale) | TextView を登録する。(view の横幅に合わせた縮小対応用) |

| クラス名 | API 名称 | API 概要 |
|-------------------------------|-----------------------------------|---------------------------------------|
| EmojiAssist.DecoEmojiTextInfo | setUri(String uri) | デコメ絵文字の画像ファイルを指す URI を設定する。 |
| | setEmojiType(int emojiType) | 対象絵文字タイプを設定する。 |
| | setContext(Context context) | コンテキストを設定する。 |
| | setWidth(int width) | デコメ絵文字の画像ファイルの幅を設定する。 |
| | setHeight(int height) | デコメ絵文字の画像ファイルの高さを設定する。 |
| | setKind(int kind) | デコメ絵文字の画像ファイルの種別を設定する。 |
| | setPop(int pop) | デコメ絵文字の画像ファイルの pop 識別子を設定する。 |
| | setCustomTags(boolean customTags) | デコメ絵文字の画像ファイルのカスタムタグの有無を設定する。 |
| 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。 |
| | getClickTagFlag() | デコメ絵文字の画像ファイルのデコメ絵文字カスタムタグ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。 |

以下、各コンポーネントについて、インタフェース仕様を記述する。

7.2. EmojiAssist クラス

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

7.2.1. 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。 | | | | | |

7.2.2. TextView 登録

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

7.2.3. 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。 | | | | | |

7.2.4. 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。 | | | | | |

7.2.5. 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。 | | | | | |

7.2.6. 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。 | | | | | |

7.2.7. 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 | decoEmojiTextInfo | DecoEmojiTextInfo | I | Must | decoEmojiText情報 |
| | | | | | |
| | | | | | |
| 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。 文字には次のAnnotationが設定される。 Annotation[decoEmoji]には複数の情報が、“ <code>¥t</code> ” 区切りで設定される。 <ul style="list-style-type: none"> ・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 Annotation[decoEmojiPop]には以下の情報が設定される。 <ul style="list-style-type: none"> ・デコメ絵文字pop識別子 Annotation[decoEmojiTags]には以下の情報が設定される。 <ul style="list-style-type: none"> ・カスタムタグ有無情報 <ul style="list-style-type: none"> ・第1引数(DecoEmojiTextInfo)の設定項目一覧 | | | | | |
| No | メンバ変数 (private) | | 設定 API | | 設定 API 詳細 |
| | 変数名 | 初期値 | | | |
| 1 | mEmojiType | 0 | setEmojiType(int emojiType) | | Annotation の設定時に必要な情報。入力フィールドの Bundle に設定されている emojiType を設定する。 |
| 2 | mContext | null | setContext(Context context) | | Annotation の設定時に必要な情報。通常は null のままとする。 |
| 3 | mUri | "" (空文字) | setUri(String uri) | | Annotation の「コンテンツの URI」にセットする。 |
| 4 | mWidth | 0 | setWidth(int width) | | Annotation の「画像ファイルの幅」にセットする。 |
| 5 | mHeight | 0 | setHeight(int height) | | Annotation の「画像ファイルの高さ」にセットする。 |
| 6 | mKind | 0 | setKind(int kind) | | Annotation の「絵文字種別」にセットする。 |
| 7 | mPop | 0 | setPop(int pop) | | Annotation の「デコメ絵文字 pop 識別子」にセットする。 |
| 8 | mTags | false | setCustomTags(boolean tags) | | Annotation の「カスタムタグの有無情報」にセットする。 |

返値の設定条件

- 1) 上記 No.1のメンバ変数の値 が 0 以下の場合 (emojiTypeが設定されない場合)
デコメ絵文字属性(Annotation[decoEmoji]) にはURI のみが設定される。(Width,Height,Kindは設定されない。)
- 2) 上記 No.4、5のメンバ変数の値が1以上の場合 (Width,Heightが設定されている場合)
No.3~7のメンバ変数に設定された値がデコメ絵文字属性(Annotation[decoEmoji]) に設定される。
- 3) 上の条件にかかわらず、Annotationの[decoEmojiPop],[decoEmojiTags]には、メンバ変数に設定がある場合はその値が設定され、ない場合 (初期値のままの場合) はAnnotationそのものが省略される。

補足

第1引数(DecoEmojiTextInfo)のメンバ変数のmUriには、絵文字マネージャから取得したURIをそのまま設定すること。

(例)

絵文字マネージャから取得できる値 : /storage/emulated/0/Download/test/test.gif

サンプルコード:

```
// カラムを作成
// 条件として{幅、高さ、絵文字種別、デコメ絵文字 pop 識別フラグ}の設定を行う
String projection[] = {DecoEmojiContract.DecoEmojiInfoColumns.WIDTH,
    DecoEmojiContract.DecoEmojiInfoColumns.HEIGHT,
    DecoEmojiContract.DecoEmojiInfoColumns.KIND,
    DecoEmojiContract.DecoEmojiInfoColumns.DECOME_POP_FLAG};

// フィルタの設定
String where = DecoEmojiContract.DecoEmojiInfoColumns.URI + " IN (" + uri + ")";

// コンテンツプロバイダから取得する
Cursor cursor = context.getContentResolver().query(DecoEmojiContract.CONTENT_DECOINFOLIST_URI,
    projection, where, null, null);

// 絵文字情報を利用者側で取得
int widthIndex = cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.WIDTH);
int heightIndex = cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.HEIGHT);
int kindIndex = cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.KIND);
int popIndex = cursor.getColumnIndex(DecoEmojiContract.DecoEmojiInfoColumns.DECOME_POP_FLAG);
int decoEmojiWidth = cursor.getInt(widthIndex);
int decoEmojiHeight = cursor.getInt(heightIndex);
int kind = cursor.getInt(kindIndex);
int pop = cursor.getInt(popIndex);

// 設定クラスを生成
EmojiAssist.DecoEmojiTextInfo info = new EmojiAssist.DecoEmojiTextInfo();
// コンテンツ URI
info.setUri(uri);
// 対象絵文字タイプ(値が0以下の場合は下位互換のため Annotation が URI のみとなる)
int emojiType = bundle.getInt("emojiType"); // 入力フィールドのBundle から対象絵文字タイプを
取得
info.setEmojiType(emojiType);

// 設定クラスに情報をセット
info.setWidth(decoEmojiWidth);
info.setHeight(decoEmojiHeight);
info.setKind(kind);
```

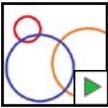
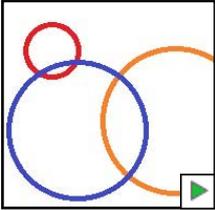
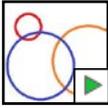
```
info.setPop(pop);
```

```
// デコメ絵文字属性 (Annotation) を設定した文字を取得する  
EmojiAssist.getInstance().getDecoEmojiText(info);
```

7.2.8. 入力文字列の絵文字有無判定

| API 名 | | checkTextData | | | |
|--|------|--|-----|-----------|----------------|
| 説明 | | 入力文字列にデコメ絵文字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) : デコメ絵文字が含まれていない。 EmojiAssist.TYPE_TEXT_DECOEMOJI(2) : デコメ絵文字が含まれている。 | | | |
| 前提 | | | | | |
| なし | | | | | |
| 概略説明 | | | | | |
| 入力文字列にデコメ絵文字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。 | | | | | |

7.2.9. 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デコメ絵文字のサイズに比例し、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 | | | | | |
|  | | | | | |
| ④ デコメ絵文字popのサイズが40px×40pxを20px×20px に縮小した場合 | | | | | |
|  | | | | | |

7.2.10. デコメピクチャ拡縮表示設定

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

7.2.11. TextView 登録(拡張版)

| API 名 | | addView | | | |
|---|-------------------|---|---|-----------|---|
| 説明 | | TextView を登録する。(view の横幅に合わせた縮小表示対応版) | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | view | TextView | I | Must | 登録するTextView |
| 2 | scale | boolean | I | Must | viewの横幅を超えた時のデコメピクチャやデコメ絵文字の縮小設定を指定 true : viewの横幅に合わせて縮小する。 false : viewの横幅に合わせて縮小しない。 |
| 戻値 | | | | | |
| 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。また、デコメピクチャやデコメ絵文字がviewの横幅を超えた場合は縮小して表示させることもできる。 | | | | | |
| 補足 | | | | | |
| 本APIとsetPictureScale()を組み合わせた際の動作は以下のようになる。 | | | | | |
| scale | setPictureScale() | 挙動 | 補足 | | |
| true | true | デコメピクチャはフォントサイズに合わせて拡張 viewのサイズを超えた場合は、縮小 | | | |
| true | false | デコメピクチャは等倍表示 viewのサイズを超えた場合でも縮小しない | setPictureScaleがfalseの場合は、addViewのscaleがtrueであってもデコメピクチャが描画フィールドの幅を超えても等倍で表示する。 | | |
| false | true | デコメピクチャはフォントサイズに合わせて拡張 viewのサイズを超えた場合でも縮小しない | | | |
| false | false | デコメピクチャは等倍表示 viewのサイズを超えた場合でも縮小しない | | | |
| 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。 | | | | | |
| 制限事項 | | | | | |
| 一つのviewに複数の文字列やデコメピクチャ、デコメ絵文字(長方形)が混在する場合には、動作保証外。 | | | | | |
| 縮小の結果、画像の高さが1ピクセル未満になる場合は、高さが1ピクセルとなる縮小率で描画を行う。 その際には、画像がviewからはみ出る可能性がある。 | | | | | |

7.3. EmojiAssist.DecoEmojiTextInfo クラス

デコメ絵文字属性を設定した文字を取得する(getDecoEmojiText()を実行する)ための情報設定クラスであり、EmojiAssist クラスの内部クラスである。

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

7.3.1. URI 設定

| API 名 | | setUri | | | |
|---|-----|---------------------------|-----|-----------|-----|
| 説明 | | デコメ絵文字の画像ファイルを指すURIを設定する。 | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | uri | String | I | Must | URI |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 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を設定する。 初期値は""(空文字)が設定されている。 | | | | | |

7.3.2. 対象絵文字タイプ設定

| | | | | | |
|---|----------------|-----------|------------|------------------|-----------|
| API名 | setEmojiType | | | | |
| 説明 | 対象絵文字タイプを設定する。 | | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | emojiType | int | I | Must | 対象絵文字タイプ |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 前提 なし | | | | | |
| 概略説明 対象絵文字タイプを設定する。 初期値は0が設定されている。 | | | | | |

7.3.3. 幅設定

| API 名 | | setWidth | | | |
|--|-------|-----------------------|-----|-----------|----|
| 説明 | | デコメ絵文字の画像ファイルの幅を設定する。 | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | width | int | I | Must | 幅 |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 前提 なし | | | | | |
| 概略説明 デコメ絵文字の画像ファイルの幅を設定する。 初期値は0が設定されている。 | | | | | |

7.3.4. 高さ設定

| API 名 | | setHeight | | | |
|---|--------|------------------------|-----|-----------|----|
| 説明 | | デコメ絵文字の画像ファイルの高さを設定する。 | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | height | int | I | Must | 高さ |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 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設定する。 初期値は0が設定されている。 | | | | | |

7.3.5. 絵文字種別設定

| API 名 | | setKind | | | |
|---|------|------------------------|-----|-----------|-------|
| 説明 | | デコメ絵文字の画像ファイルの種別を設定する。 | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | kind | int | I | Must | 絵文字種別 |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 前提 なし | | | | | |
| 概略説明 デコメ絵文字の画像ファイルの種別を設定する。 初期値は0が設定されている。 | | | | | |

7.3.6. pop 識別子 設定

| API 名 | | setPop | | | |
|---|-----|----------------------------|-----|-----------|--------|
| 説明 | | デコメ絵文字の画像ファイルのpop識別子を設定する。 | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | pop | int | I | Must | pop識別子 |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 前提 なし | | | | | |
| 概略説明 デコメ絵文字の画像ファイルのpop識別子を設定する。 初期値は0が設定されている。 | | | | | |

7.3.7. カスタムタグ有無情報設定

| | | | | | |
|--|-------------------------------|-----------|------------|------------------|-------------------|
| API名 | setCustomTags | | | | |
| 説明 | デコメ絵文字の画像ファイルのカスタムタグの有無を設定する。 | | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| 1 | customTags | boolean | I | Must | カスタムタグ有無情報 |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| | | | | | |
| 前提 なし | | | | | |
| 概略説明 デコメ絵文字の画像ファイルのカスタムタグの有無を設定する。 初期値はfalseが設定されている。 | | | | | |

7.4. 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

7.4.1. 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();
```

7.4.2. 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。 | | | | | |

7.4.3. 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。 | | | | | |

7.4.4. 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 | | | | | |

7.4.5. カスタムタグ情報取得

| API 名 | getClickTagFlag | | | | |
|---|--------------------------------|-----------|-----|-----------|----|
| 説明 | デコメ絵文字の画像ファイルのカスタムタグ有無情報を取得する。 | | | | |
| 引数 | | | | | |
| No | 引数名 | 型 | I/O | Must/Opt. | 説明 |
| | | | | | |
| | | | | | |
| | | | | | |
| 返値 | | | | | |
| No | 型 | 説明 | | | |
| 1 | boolean | カスタムタグの有無 | | | |
| 前提 なし | | | | | |
| 概略説明 デコメ絵文字の画像ファイルのカスタムタグの有無情報を取得する。 カスタムタグ有無情報の値と定義は以下の通り。 false : カスタムタグなし true : カスタムタグあり | | | | | |

7.4.6. 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();
```

8. 絵文字マネージャ

絵文字マネージャ (DecoEmojiManager) は、デコモ絵文字の画像ファイルと ID、および付加情報を管理するサービスアプリです。

(1) デコモ絵文字の登録

システムで共通に利用するデコモ絵文字は、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ります。登録先ディレクトリに登録されたデコモ絵文字は、絵文字マネージャによって検出・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)することができます。

端末の別アプリ(メール、ブラウザやコンテンツ管理アプリ)が新規にデコモ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ます。

(2) アプリケーションへの通知 (Broadcast)

デコモ絵文字の追加、または削除が検出された場合、絵文字マネージャは **Broadcast** によりデコモ絵文字更新情報を利用アプリに通知します。また、IME がデコモ絵文字予測/変換入力機能を含む場合は、さらに **Broadcast** によりデコモ絵文字辞書情報を IME に通知します。

また、絵文字マネージャ自身のサービス状態が変化した場合に、**Broadcast** により状態を通知します。

(3) アプリケーションインタフェース

デコモ絵文字の一覧を表示したり、デコモ絵文字をアプリケーションに入力したりするサービスを提供するアプリ(IME)や、デコモ絵文字の付加情報を利用するアプリは、次の二つのインタフェースを通じて、絵文字マネージャの持つ情報にアクセスすることができます。

- (A) 絵文字マネージャ ContentProvider インタフェース
- (B) 絵文字マネージャ サービス(aidl)インタフェース

なお、絵文字マネージャ (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、次のライブラリをリンクする必要があります。

- 絵文字マネージャインタフェースライブラリ (DecoEmojiInterface.jar)
ライブラリに含まれるクラスは次の通りです。
 - (A) IDecoEmojiManager
 - (B) DecoEmojiContract

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>"として表記しています。

例: "jp.co.omronsoft.android.decoemojimanager_<carrier>"

API Level は 23 に設定する必要があります。

8.1. 絵文字マネージャ 要求(Broadcast)インタフェース

絵文字マネージャが要求を受信する広報通知(Broadcast)の詳細について説明します。
絵文字マネージャでは、次の Broadcast を受信します。

(1) ファイル差分チェック実施

- ・ デコメ絵文字情報を保持するアプリから、ファイル差分チェックの実施を要求します
- ・ ファイル差分チェック完了の通知については「8. 2. 3絵文字マネージャ状態変更通知」を参照。

(2) 絵文字マネージャ通知アプリ登録

- ・ 絵文字マネージャからの各種通知 (Broadcast) を受信する BroadcastReceiver の登録を要求します。
※Android 0 から、暗黙的な Broadcast を受信する動作が制限されたためクラスの指定が必要

8.1.1. ファイル差分チェック実施要求

| データ名称 | 入出力内容 | |
|----------|--|--|
| Action 名 | IDecoEmojiConstant. ACTION_FILE_CHECK ("jp.co.omronsoft.android.decoemoji.manager.docomo.FILE_CHECK") | |
| Extras | IDecoEmojiConstant. BROADCAST_PACKAGE_NAME_TAG | 要求元アプリのパッケージ名。(任意) パッケージ名を格納してから Intent を送信する必要がある。 ※指定がない場合には、「8. 2. 4ファイル差分チェック要求応答通知」の通知が行われない。 |
| | IDecoEmojiConstant. BROADCAST_PATH_TAG | 指定フォルダ。(任意) ファイル差分チェックを行うフォルダのパスを指定できる。 パスには必ず末尾に「/」を付けること。 ※登録先フォルダ配下でない場合、チェックは行われない。 通知については「8. 2. 4ファイル差分チェック要求応答通知」を参照。 |

サンプルコード(「/storage/emulated/0/Download/」配下をファイル差分チェックする場合):

```
// パッケージ名取得
String packageName = context.getPackageName();
// Intent のインスタンス生成
Intent intent = new Intent(IDecoEmojiConstant. ACTION_FILE_CHECK);
intent.setClassName(IDecoEmojiConstant. PACKAGE_NAME,
                    IDecoEmojiConstant. CLASS_NAME_RECEIVER);
// Bundle に情報格納
Bundle bundle = new Bundle();
bundle.putString(IDecoEmojiConstant. BROADCAST_PACKAGE_NAME_TAG, packageName);
bundle.putString(IDecoEmojiConstant. BROADCAST_PATH_TAG, "/storage/emulated/0/Download/");
// Intent に Bundle 格納
intent.putExtras(bundle);
// Intent 送信
sendBroadcast(intent);
```

8.1.2. 絵文字マネージャ通知アプリ登録要求

要求を受信し、絵文字マネージャからの通知を行うアプリは最大 10 件です。11 件目の要求を受けた時、要求を受けたのが最も古いアプリが、絵文字マネージャから通知されなくなります。

| データ名称 | | 入出力内容 |
|----------|--|--|
| Action 名 | | IDecoEmojiConstant. ACTION_ADD_BROADCAST_APP ("jp. co. omronsoft. android. decoemojimanager_docomo. ADD_BROADCAST_APP") |
| Extras | IDecoEmojiConstant. BROADCAST_PACKAGE_NAME_TAG | 登録するアプリのパッケージ名。 |
| | IDecoEmojiConstant. BROADCAST_CLASS_NAME_TAG | 登録するアプリの BroadcastReceiver のクラス名。 |

サンプルコード(「com. example. SampleBroadcatReceiver」を登録要求する場合) :

```
// パッケージ名取得
String packageName = context.getPackageName();
// Intent のインスタンス生成
Intent intent = new Intent(IDecoEmojiConstant. ACTION_ADD_BROADCAST_APP);
intent.setClassName(IDecoEmojiConstant. PACKAGE_NAME,
                    IDecoEmojiConstant. CLASS_NAME_RECEIVER);
// Bundle に情報格納
Bundle bundle = new Bundle();
bundle.putString(IDecoEmojiConstant. BROADCAST_PACKAGE_NAME_TAG, packageName);
bundle.putString(IDecoEmojiConstant. BROADCAST_CLASS_NAME_TAG, packageName + " SampleBroadcatReceiver");
// Intent に Bundle 格納
intent.putExtras(bundle);
// Intent 送信
sendBroadcast(intent);
```

8.2. 絵文字マネージャ 通知(Broadcast)インタフェース

絵文字マネージャから送出される広報通知 (Broadcast) の詳細について説明します。
絵文字マネージャでは、次の4つの Broadcast を発行します。

(1) デコメ絵文字情報更新通知

- ・デコメ絵文字情報を保持するアプリに対して、デコメ絵文字情報の更新を通知します。

(2) デコメ絵文字辞書更新通知

- ・デコメ絵文字辞書情報を保持するアプリに対し、デコメ絵文字辞書情報の更新を通知します。

(3) 絵文字マネージャ状態変更通知

- ・絵文字マネージャのサービス状態変更を通知します。

(4) ファイル差分チェック要求応答通知

- ・ファイル差分チェック要求時の応答状態を通知します。

8.2.1. デコメ絵文字情報更新通知

デコメ絵文字（GIF、JPG ファイル）の情報更新通知

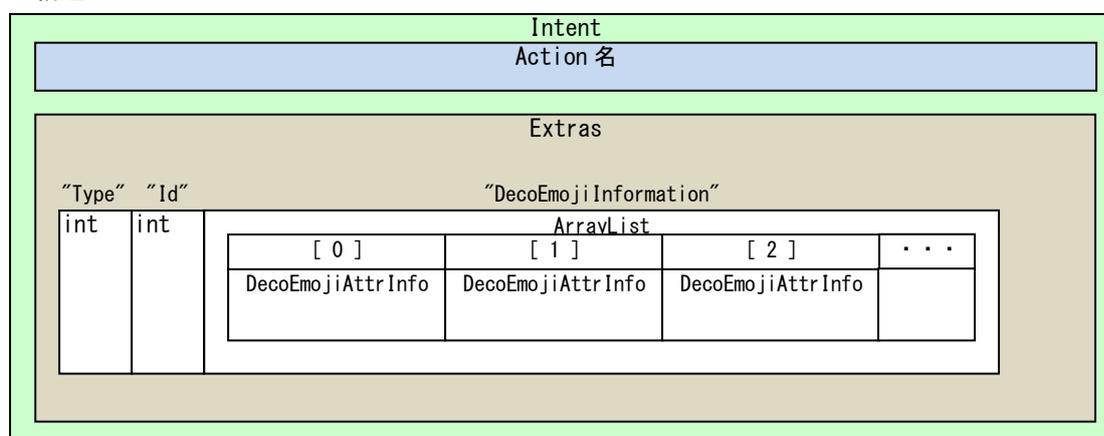
| データ名称 | | 入出力内容 |
|----------|---|--|
| Action 名 | | IDecoEmojiConstant. ACTION_CHANGE_DATABASE |
| Extras | IDecoEmojiConstant. BROADCAST_TYPE_TAG | 更新区分。格納する区分は下記の通り。 追加： IDecoEmojiConstant. FLAG_INSERT 削除： IDecoEmojiConstant. FLAG_DELETE 初期化： IDecoEmojiConstant. FLAG_SET_TO_INITIALIZING |
| | IDecoEmojiConstant. 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 |

8.2.2. デコメ絵文字辞書更新通知

デコメ絵文字（GIF、JPG ファイル）の辞書情報更新通知

| データ名称 | | 入出力内容 |
|----------|--|---|
| Action 名 | | IDecoEmojiConstant. ACTION_DECOEMOJI_RESULT |
| Extras | IDecoEmojiConstant. BROADCAST_TYPE_TAG (Type) | 更新区分。格納する区分は下記の通り。 追加： IDecoEmojiConstant. FLAG_INSERT 削除： IDecoEmojiConstant. FLAG_DELETE 初期化： IDecoEmojiConstant. FLAG_SET_TO_INITIALIZING 更新データなし： IDecoEmojiConstant. FLAG_FINISH |
| | IDecoEmojiConstant. BROADCAST_ID_TAG (Id) | IDecoEmojiConstant. NO_CHECKEMOJI |
| | IDecoEmojiConstant. BROADCAST_DATA_TAG (DecoEmojiInformation) | DecoEmojiAttrInfo の ArrayList。(最大 100 件) DecoEmojiAttrInfo には、更新対象デコレーション絵文字 ID、読み一覧、品詞一覧が含まれる。 ★更新区分が初期化、更新データなし時は null |

Intent の構造



8.2.3. 絵文字マネージャ状態変更通知

| データ名称 | | 入出力内容 |
|---|---|---|
| Action 名 | | IDecoEmojiConstant. ACTION_STATE_CHANGED |
| Extras | IDecoEmojiConstant. BROADCAST_STATE_TAG | 状態変更区分。格納する区分は下記の通り。 初期化完了： IDecoEmojiConstant. STATE_INIT_COMPLETE ファイルチェック完了(DB更新あり)： IDecoEmojiConstant. STATE_FILE_CHECK_COMPLETE ファイルチェック完了(DB更新なし)： IDecoEmojiConstant. STATE_FILE_CHECK_COMPLETE_NO_UPDATE |
| 状態 | | 通知タイミング |
| IDecoEmojiConstant. STATE_INIT_COMPLETE | | デコメ絵文字を管理するDBが構築、もしくは再構築され、DBが利用可能となった際に通知される |
| IDecoEmojiConstant. STATE_FILE_CHECK_COMPLETE | | 絵文字マネージャ起動時に、デコメ絵文字を管理するDBと登録先ディレクトリに格納されているデコメ絵文字の差分が100件以下の場合には、DBの差分更新が行われ、DBが利用可能となった際に通知される ※101件以上の場合には、DBの再構築が行われるため、IDecoEmojiConstant. STATE_INIT_COMPLETEが通知される |
| IDecoEmojiConstant. STATE_FILE_CHECK_COMPLETE_NO_UPDATE | | 絵文字マネージャ起動時に、デコメ絵文字を管理するDBと登録先ディレクトリに格納されているデコメ絵文字の差分が0件で、DBが更新されず利用可能となった際に通知される |
| | | 値 |
| | | 0 |
| | | 1 |
| | | 2 |

8.2.4. ファイル差分チェック要求応答通知

| データ名称 | | 入出力内容 |
|--|---|--|
| Action 名 | | "[ファイル差分チェック要求元パッケージ名]. MANAGER_STATE" ※定数ではなく文字列 |
| Extras | IDecoEmojiConstant. BROADCAST_MANAGER_STATE_TAG | ファイル差分チェック要求時の絵文字マネージャの状態区分。格納する区分は下記の通り。 初期化中状態： IDecoEmojiConstant. INITIALIZING 指定フォルダ不正： IDecoEmojiConstant. NG_FILE_CHECK_PATH |
| 状態 | | 通知タイミング |
| IDecoEmojiConstant. INITIALIZING | | Intent 受信(ファイル差分チェック実施要求 参照)により、ファイル差分チェックが要求された際、絵文字マネージャが初期化中状態で、ファイル差分チェックが即時に実施されない場合、通知する。 なお、初期化処理完了後に、自動的に要求されたファイル差分チェックが実施される。 |
| IDecoEmojiConstant. NG_FILE_CHECK_PATH | | Intent 受信(ファイル差分チェック実施要求 参照)により、ファイル差分チェックが要求された際、ファイル差分チェック対象に指定されたフォルダが登録先フォルダ配下でない場合、通知する。 この時、ファイル差分チェックは実施されない。 |

8.2.5. 補助クラス

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

8.2.5.1. 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: DecoEmojiInformation | in から DecoEmojiAttrInfo の一つのデータセット: 絵文字 ID(int)、読み(文字列配列)、品詞(byte 配列)、表記(文字列配列)を読み出す |
| int getId() | なし | 絵文字 ID を取得する |
| void setID(int id) | id : ID | 絵文字 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(idx は 0-10、以下同じ) |
| 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 |
| byte getPart(int idx) | idx:インデックス | idx 番目の品詞を取得する |
| 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 |

8.2.6. 定数

8.2.6.1. 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(初期化処理の開始を通知) | 4 |

8.2.6.2. DecoEmojiManager 状態種別

| 名前 | 意味 | 値 |
|--|---|---|
| IDecoEmojiConstant.STATE_INIT_COMPLETE | 初期化完了 DB が初期生成、再構築された状態を示す。 | 0 |
| IDecoEmojiConstant.STATE_FILE_CHECK_COMPLETE | ファイルチェック完了 DB は初期化されず、画像ファイルとの一致確認と一部更新が行われた状態を示す。 | 1 |

8.2.6.3. 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.decoemoji_manager_<carrier>" |
| IDecoEmojiConstant.ACTION_CHANGE_DATABASE | デコメ絵文字情報更新情報 | "jp.co.omronsoft.android.decoemoji_manager_<carrier>.CHANGE_DATABASE" |
| IDecoEmojiConstant.BROADCAST_STATE_TAG | 絵文字マネージャ状態 | "State" |
| IDecoEmojiConstant.ACTION_STATE_CHANGED | 絵文字マネージャ状態変更通知 | "jp.co.omronsoft.android.decoemoji_manager.STATE_CHANGED" |

8.3. 絵文字マネージャ ContentProvider インタフェース

絵文字マネージャ (DecoEmojiManager) の持つ DB の内容を、ContentProvider のインタフェースを通じてアプリケーションから参照することができます。

8.3.1. 絵文字パッケージで提供する ContentProvider 機能

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

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 機能を提供します。

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

8.3.2. 前提条件

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

- ・ 絵文字ライブラリ、絵文字マネージャ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 DB が存在すること。

もし DB が存在しない状態で本機能を使用した場合は、エラーとなります。

なお、「絵文字マネージャ (DecoEmojiManager)」は、DB が存在しない場合、起動時に DB を生成する仕組みであるため、アプリ起動から一定時間経過後に本機能は使用可能となります。なお、絵文字マネージャは、アプリからサービス (ContentProvider または aidl) への最初のアクセスによって起動されます。

8.3.3. 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」と呼びます。取得モードごとの取得データは以下の通りです。

取得データの詳細内容は「8.3.6 取得データ詳細」を参照してください。

| 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 list | 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 count | 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 | decodic list | 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 | decodic count | デコメ絵文字辞書件数テーブル |
| 5 | デコメ絵文字カスタムタグ情報一覧取得 | customtag list | デコメ絵文字カスタムタグテーブル |

8.3.4. 使用方法

Content Provider では、ContetURI に基づき、データを取得します。

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

| No | 取得モード | ContetURI 定数 |
|----|------------------|---------------------------------|
| 1 | デコメ絵文字情報一覧取得 | CONTENT_DECOINFO LIST_URI |
| 2 | デコメ絵文字情報件数取得 | CONTENT_DECOINFO LIST_COUNT_URI |
| 3 | デコメ絵文字辞書一覧取得 | CONTENT_DECODIC LIST_URI |
| 4 | デコメ絵文字辞書件数取得 | CONTENT_DECODIC LIST_COUNT_URI |
| 5 | デコメ絵文字カスタムタグ情報取得 | CONTENT_CUSTOMTAG LIST_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ります。
詳細は「 8.3.7 取得モード詳細 」を参照してください。

8.3.5. ContentProvider 利用時の query パラメータについて

引数 selection に文字列定義を行わず、selectionArgs に文字列定義を行ってください。

サンプルコード:

```
// 条件句を作成
// 条件として path の設定を行う
String selection = "path = ?";
//条件句のパラメータを作成
String selectionArgs [] = {""};
selectionArgs [0] = "/storage/emulated/0/Download/test.gif";
//デコメ絵文字情報件数取得
try {
    Cursor cur = resolver.
        query(DecoEmojiContract.CONTENT_DECOINFOLIST_COUNT_URI, null, selection,selectionArgs,null);
} catch (SQLiteException e) {
    #エラー処理
} finally {
}
}
```

8.3.6. 取得データ詳細

| 項目名称 | 項目 ID | 型 | デコメ絵文字情報一覧取得 | | デコメ絵文字情報件数取得 | | デコメ絵文字辞書一覧取得 | | デコメ絵文字辞書件数取得 | | デコメ絵文字カスタムタグ情報一覧取得 | | 備考 |
|--------------|--------------------|---------|--------------|----|--------------|----|--------------|----|--------------|----|--------------------|----|--|
| | | | 取得 | 条件 | 取得 | 条件 | 取得 | 条件 | 取得 | 条件 | 取得 | 条件 | |
| デコメ絵文字 ID | decoemoji_id | INTEGER | ○ | ○ | | ○ | ○ | ○ | | ○ | ○ | ○ | |
| デコメ絵文字 URI | uri | TEXT | ○ | ○ | | ○ | ○ | ○ | | ○ | ○ | ○ | ファイルの URI (file:// or content://) |
| デコメ絵文字 PATH | path | TEXT | ○ | ○ | | ○ | ○ | ○ | | ○ | ○ | ○ | ファイルの絶対パス |
| カテゴリ ID | category_id | INTEGER | ○ | ○ | | ○ | | | | | | | |
| カテゴリ名 (日本語) | category_name_jpn | TEXT | ○ | ○ | | ○ | | | | | | | |
| カテゴリ名 (英語) | category_name_eng | TEXT | ○ | ○ | | ○ | | | | | | | |
| カテゴリプリセット ID | category_preset_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。 |
| ディレクトリ ID | directory_id | INTEGER | ○ | ○ | | ○ | | | | | | | |
| ディレクトリ名 | directory_name | TEXT | ○ | ○ | | ○ | | | | | | | |
| 更新タイムスタンプ | timestamp | INTEGER | ○ | ○ | | ○ | ○ | ○ | | ○ | | | 1970 年 1 月 1 日午前 0 時からの秒数 |
| ファイルサイズ | file_size | INTEGER | ○ | ○ | | ○ | | | | | | | |

| | | | | | | | | | | | | | |
|------------------|---------------------|---------|---|---|---|---|---|---|---|--|--|--|--|
| 拡張タグ名 | tags | TEXT | ○ | ○ | ○ | | | | | | | | DB登録時に拡張タグ名(<tags>タグ内のname属性の値)が登録されている。 複数の拡張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。 |
| 幅 | width | INTEGER | ○ | ○ | ○ | | | | | | | | |
| 高さ | height | INTEGER | ○ | ○ | ○ | | | | | | | | |
| 絵文字種別 | kind | INTEGER | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | 1:「①デコレーション絵文字(20*20)」 2:「②デコレーション絵文字(正方形)」 3:「③デコレーション絵文字(長方形)」 4:「④デコメピクチャ」 |
| 最終利用表示順 | last_use_count | INTEGER | ○ | ○ | ○ | | | | | | | | |
| 利用履歴表示順 | history_count | INTEGER | ○ | ○ | ○ | | | | | | | | |
| 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 |
| ファイル最終更新日時 | file_lastmodified | INTEGER | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | 1970年1月1日午前0時からの秒数 |
| デコメ絵文字情報件数 | decoemoji_dic_count | INTEGER | | | ○ | | | | | | | | |
| 読み | decoemoji_name | TEXT | | | | | ○ | ○ | ○ | | | | |
| 品詞 | decoemoji_part | INTEGER | | | | | ○ | ○ | ○ | | | | |
| 表記 | decoemoji_note | TEXT | | | | | ○ | ○ | ○ | | | | |
| デコメ絵文字辞書件数 | decoemoji_dic_count | INTEGER | | | | | | | ○ | | | | |

| | | | | | | | | | | | | | | |
|-----------|-------------------|---------|--|--|--|--|--|--|--|--|--|---|---|--|
| カスタムタグ ID | customtag_id | INTEGER | | | | | | | | | | ○ | ○ | |
| カスタムタグ名 | customtag_name | TEXT | | | | | | | | | | ○ | ○ | |
| カスタムタグ要素 | customtag_element | TEXT | | | | | | | | | | ○ | ○ | |

8.3.7. 取得モード詳細

8.3.7.1. デコメ絵文字情報一覧取得 IF

| インターフェース名 | デコメ絵文字情報一覧取得 | | | |
|------------|--|--------------------|----------|---|
| 取得モード | Decoinfolist | | | |
| 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 | <ul style="list-style-type: none"> ・詳細取得方式 DecoEmojiContract.CONTENT_DECOINFOLIST_URI ・簡易取得方式 DecoEmojiContract.CONTENT_DECOINFOLIST_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。 <ul style="list-style-type: none"> ●重複除外を指定して取得する場合 DecoEmojiContract.CONTENT_DECOINFOLIST_URI .buildUpon() .appendQueryParameter (DecoEmojiContract. QUERY_PARAM_DISTINCT, "true") .build() ※レコード結果が完全一致する場合のみ重複の結果が除外されるため、レコードに一つでも異なる項目がある場合重複除外されない。 |
| 2 | projection | String[] | Opt | 取得カラム … 取得するカラム名 <ul style="list-style-type: none"> ・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。 ・「8.3.6取得データ詳細」参照 |
| 3 | selection | String | Opt | フィルタ … フィルタ (Where条件) <ul style="list-style-type: none"> ・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。 ・「8.3.6取得データ詳細」参照 |
| 4 | selectionArgs | String[] | Opt | クエリパラメータ … クエリパラメータ <ul style="list-style-type: none"> ・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。 ・「8.3.6取得データ詳細」参照 |
| 5 | sortOrder | String | Opt | ソート順 … ソート順 <ul style="list-style-type: none"> ・省略する場合はNULLを指定する。その場合、昇順で取得する。 ・「8.3.6取得データ詳細」条件を利用可能 |
| 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 | | |

サンプルコード (詳細取得方式) :

```
String column[] = {DecoEmojiContract.DecoEmojiInfoColumns.URI};
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(Uri.withAppendedPath(DecoEmojiContract.CONTENT_DECOINFOLIST_URI, "/1234"), null, null, null, null);
```

サンプルコード : カテゴリー一覧を取得したい場合 (重複除外を指定)

```
// 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);
// デコメ絵文字情報一覧取得 URI
// パラメータに重複除外を設定
Uri uri = DecoEmojiContract.CONTENT_DECOINFOLIST_URI
    .buildUpon()
    .appendQueryParameter(DecoEmojiContract.QUERY_PARAM_DISTINCT, "true")
    .build();
// カラムを作成
// 条件として {カテゴリー ID、カテゴリー名(日本語)、カテゴリー名(英語)} の設定を行う
String projection [] = {DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_ID
    , DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_NAME_JPN
    , DecoEmojiContract.DecoEmojiInfoColumns.CATEGORY_NAME_ENG };
// 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";
// カテゴリー取得
Cursor cursor = mContext.getContentResolver().query(uri, projection, 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)));
        // カテゴリー情報を追加する
        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 = mContext.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();
}
```

8.3.7.2. デコメ絵文字情報件数取得 IF

| 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。 ・「 8.3.6取得データ詳細 」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。 ・「 8.3.6取得データ詳細 」参照 |
| 5 | sortOrder | String | - | ソート順 … ソート順 ・「 8.3.6取得データ詳細 」条件を利用可能 |
| 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 | | |

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);
```

8.3.7.3. デコメ絵文字辞書一覧取得 IF

| 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 | <ul style="list-style-type: none"> ・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。 <ul style="list-style-type: none"> ●重複除外を指定して取得する場合 DecoEmojiContract.CONTENT_DECODICLIST_URI .buildUpon() .appendQueryParameter (DecoEmojiContract.QUERY_PARAM_DISTINCT, "true") .build() ※レコード結果が完全一致する場合のみ重複の結果が除外されるため、レコードに一つでも異なる項目がある場合重複除外されない。 |
| 2 | projection | String[] | Opt | 取得カラム … 取得するカラム名 <ul style="list-style-type: none"> ・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。 ・「8.3.6取得データ詳細」参照 |
| 3 | selection | String | Opt | フィルタ … フィルタ(Where条件) <ul style="list-style-type: none"> ・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。 ・「8.3.6取得データ詳細」参照 |
| 4 | selectionArgs | String[] | Opt | クエリパラメータ … クエリパラメータ <ul style="list-style-type: none"> ・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。 ・「8.3.6取得データ詳細」参照 |
| 5 | sortOrder | String | Opt | ソート順 … ソート順 <ul style="list-style-type: none"> ・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 | | |

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(Uri.withAppendedPath(DecoEmojiContract.CONTENT_DECODICLIST_URI, "/1234"),
null, null, null, null);
```

8.3.7.4. デコメ絵文字辞書件数取得 IF

| 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。 ・「 8.3.6取得データ詳細 」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。 ・「 8.3.6取得データ詳細 」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 | | |

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);
```

8.3.7.5. デコメ絵文字カスタムタグ情報一覧取得 IF

| インタフェース名 | デコメ絵文字カスタムタグ情報一覧取得 | | | |
|------------|--|----------------------|----------|--|
| 取得モード | customtaglist | | | |
| 説明 | デコメ絵文字カスタムテーブルから、検索条件に沿ったデコメ絵文字カスタムタグ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 | <ul style="list-style-type: none"> ・詳細取得方式 DecoEmojiContract. CONTENT_CUSTOMTAGLIST_URI ●重複除外を指定して取得する場合 DecoEmojiContract. CONTENT_CUSTOMTAGLIST_URI .buildUpon() .appendQueryParameter (DecoEmojiContract. QUERY_PARAM_DISTINCT, "true") .build() ※レコード結果が完全一致する場合のみ重複の結果が除外されるため、レコードに一つでも異なる項目がある場合重複除外されない。 |
| 2 | projection | String[] | Opt | <ul style="list-style-type: none"> 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。 ・「8.3.6取得データ詳細」参照 |
| 3 | selection | String | Opt | <ul style="list-style-type: none"> 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。 ・「8.3.6取得データ詳細」参照 |
| 4 | selectionArgs | String[] | Opt | <ul style="list-style-type: none"> 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。 ・「8.3.6取得データ詳細」参照 |
| 5 | sortOrder | String | Opt | <ul style="list-style-type: none"> 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 | デコメ絵文字カスタムテーブルへのカーソル | | |

サンプルコード :

```
String column[] = {DecoEmojiContract. CustomTagColumns. CUSTOMTAG_NAME,
DecoEmojiContract. CustomTagColumns. CUSTOMTAG_ELEMENT};
String selection = DecoEmojiContract. CustomTagColumns. URI + " = ?";
String selectionArgs = {""};
selectionArgs [0] = uri;
String order = DecoEmojiContract. CustomTagColumns. CUSTOMTAG_ID + " ASC";
Cursor cur = resolver. query (DecoEmojiContract. CONTENT_CUSTOMTAGLIST_URI, column, selection, selectionArgs,
order);
```

8.3.8. 補助クラス (DecoEmojiContract クラス)

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

8.3.8.1. 「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 | | | | | |
| 概略説明 | | | | | |
| <ul style="list-style-type: none"> 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) 絵文字種別が複数ある場合には、“,”区切りの文字列に変換する。 0x0000 以下、もしくは 0x0010 以上の値が指定された場合には、1 を返却する。 <p>(例1) 引数のemojiTypeが0x0002 返却値: 2</p> <p>(例2) 引数のemojiTypeが0x000F 返却値: 1,2,3,4</p> <p>(例3) 引数のemojiTypeが0xFFFF 返却値: 1</p> | | | | | |

サンプルコード:

```
int emojiType = bundle.getInt( "emojiType" );
String emojiKind = DecoEmojiContract.makeStringEmojiKind(emojiType);
String where = DecoEmojiContract.DecoEmojiInfoColumns.USE_FLAG + " = 1 AND " +
DecoEmojiContract.DecoEmojiInfoColumns.KIND + " IN (" + emojiKind + ")";
Cursor cursor =
mContext.getContentResolver().query(DecoEmojiContract.CONTENT_DECOINFOLIST_URI, null, where,
null, order);
```

8.3.8.2. 「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 | | | | | |
| <ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 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
}

```

8.3.9. 定数

8.3.9.1. 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 |

8.3.9.2. クエリパラメータ定数定義

| No | 定数名 | 型 | 定数内容 |
|----|--|--------|------------|
| 1 | DecoEmojiContract.QUERY_PARAM_DISTINCT | String | 検索結果重複除外設定 |

8.3.9.3. 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 |
| 19 | DecoEmojiContract.DecoEmojiInfoColumns.FILE_LASTMODIFIED | String | (A).ファイル最終更新日時項目名 |

8.3.9.4. 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.DECOEM OJI_ID | String | (B).デコメ絵文字 ID 項目名 |
| 2 | DecoEmojiContract.DecoEmojiDicColumns.URI | String | (B).デコメ絵文字 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 |
| 8 | DecoEmojiContract.DecoEmojiDicColumns.FILE_LAS TMODIFIED | String | (B).ファイル最終更新日時項目名 |

8.3.10. データ取得時の推奨事項について

最新のデコメコンテンツの情報を取得する場合、アプリ側で取得前に「8.1.1ファイル差分チェック実施要求」を利用することを推奨します。

8.3.11. 例外

- (2) query の ContentURI が正しくない場合
例外 (IllegalStateException) を throw します。
- (3) DB 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 します。
例外については、Android の SQLite の仕様に準拠します。
- (4) 絵文字マネージャが DB の構築中、再構築中の場合
本ケースは Query で例外は発生せず、戻り値の cursor が null となります。この場合は、アプリ (IME) は DB にアクセスはできないので、処理は初期化処理終了が Broadcast で通知されるまで待つ必要があります。

8.4. 絵文字マネージャー サービス(aidl)インタフェース

絵文字マネージャーの DB データ取得以外のサービス機能利用インタフェースを、サービス呼び出し (Android Interface Definition Language) インタフェースとして定義します。

8.4.1. 絵文字マネージャー サービス (aidl) インタフェースの接続と切断

絵文字マネージャー (DecoEmojiManager) は、サービスアプリとして動作します。サービスに接続する場合は下記のように接続 (bind) する必要があります。

bind が成功した後、サービスインタフェースの aidl API を用いることができます。

サンプルコード：

```
// サービス接続
Intent intent = new Intent();
intent.setClassName("jp.co.omronsoft.android.decoemojimanager_docomo",
    "jp.co.omronsoft.android.decoemojimanager_docomo.DecoEmojiManager");
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);
```

8.4.2. サービス(aidl)インタフェースの API

絵文字マネージャ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

8.4.2.1. 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 | | | | | |
| <ul style="list-style-type: none"> 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。 | | | | | |

8.4.2.2. 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 | | | | | |
| <ul style="list-style-type: none"> ・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。 <ul style="list-style-type: none"> 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) | | | | | |

8.4.2.3. 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「9.1 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 | | | | | |
| <ul style="list-style-type: none"> 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。 | | | | | |

8.4.3. 定数

8.4.3.1. 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_URI | エラー:「uri」の値が不正 | -19 |
| 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 |

9. Data Format

9.1. 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。

9.1.1. 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) |

9.1.2. 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) |

9.1.3. デコメ絵文字 pop ファイルの判定方法

デコメ絵文字ファイルが、以下の 5 つの条件を全て満たす場合にデコメ絵文字 pop と判定する。

- (1)ファイル形式が「GIF89」であること
GIF Header のシグネチャ/バージョンを参照し、「GIF89」である。
- (2)Application Identifier Application が「HTMLEMJI」であること
Application Extension Block 内の Application Identifier を参照し、「HTMLEMJI」である。
- (3) Authentication Code が「1.0」であること
Application Extension Block 内の Application Authentication Code を参照し、「1.0」である。
- (4)素材データが存在すること
Application Extension Block 内の Application Data の Block Size の値が 1Byte 以上ある。
- (5)製作者名、絵文字 ID が設定されていること
Comment Extension Block 内の Comment Data に製作者、絵文字 ID が設定されている。
また、製作者、絵文字 ID には半角英数字、ハイフン「-」、ドット「.」、アンダーバー「_」のみが使用されている。

| 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 |
| Comment Extension Block | Extension Introducer | 1 byte | 0x21 |
| | Comment Label | 1 byte | 0xFE |
| | Block Size | 1 byte | Comment Data のデータサイズ (例：255 bytes の場合は 0xFF) |
| | Comment Data | n byte | 0x43 0x72 0x65 0x61 0x74 0x6F 0x72 0x3D XXXXXX 0x0D 0x0A 0x4E 0x61 0x6D 0x65 0x3D YYYYY 0x0D 0x0A 0x45 0x6D 0x6F 0x6A 0x69 0x49 0x44 0x3D ZZZZZZ (XXXXXX は製作者、会社名を指す。YYYYY は絵文字名を指す。ZZZZZZ は絵文字 ID を指す。) |
| | Block Terminator | 1 byte | 0x00 |

9.1.4. 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（タイプにデコメ絵文字を定義した場合）

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

付加情報の格納例（タイプにデコメピクチャを定義した場合）

```
<emoji>
  <category>感情</category>
  <type>picture</ type >
</emoji>
```

付加情報の格納例（カスタムタグに<URL>、<MEDIA>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="MEDIA">INTERNAL_SDCARD/aaa.ogg </tags>
</emoji>
```

9.1.5. 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、文字種は任意) 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。(64 文字 (256Bytes) まで、文字種は任意) 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。) ※Windows 上でフォルダ名に利用できない文字、Android 上に搭載されている SQLite でエスケープ処理が必要な文字(半角、全角共に)は除く 例) ¥/:*?"<> '_% |
| Type | タイプ | 省略可 (0-1) | 画像ファイルが「デコレーション絵文字」、「デコメピクチャ」のいずれであるかを格納する。「デコレーション絵文字」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。 |
| 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。 カスタムタグは複数個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。 カスタムタグ名は最大 24 文字、文字種は任意である。 要素内容の最大文字数は 1024Bytes とする。 また、カスタムタグ名および要素内容が空文字のカスタムタグは記述不可とする。 |

9.1.6. 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 >
  ] >
```

9.1.7. 付加情報文字列の上限

上限は 10240bytes である。

9.1.8. 付加情報文字列の文字コード

文字コードは UTF-8 である。

9.1.9. 付加情報文字列の分割

付加情報の 1 つのタグ文字列は、255 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。

9.1.10. 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。

9.2. DB エンティティ情報

絵文字マネージャで利用する DB のエンティティ情報は以下の通り。

9.2.1. デコメ絵文字情報テーブル[decoemoji_info]

| 論理名称 | 物理名称 | 主キー | 必須 | 型 | 初期値 | DB 変更契機 | | | | 備考 |
|-------------|--------------|-----|----|---------|----------------------------------|------------|------------|---|---------|--------------------------|
| | | | | | 絵文字マネージャ起動時に格納されているコンテンツの DB 登録値 | 再構築 | 初期化 | コンテンツ追加 | コンテンツ削除 | |
| デコメ絵文字 ID | decoemoji_id | ○ | ○ | INTEGER | 0 より順番に ID が割り当てられる | 初期値 ※ 1 | 初期値 ※ 1 | 最大値+1 ただし、9999 まで値が利用されている場合には、利用フラグが 0 の id を利用する | 値を保持 | 上限 : 9999 |
| デコメ絵文字 URI | uri | | ○ | TEXT | デコメ絵文字コンテンツ URI | 初期値 | 初期値 | 初期値 | 値を保持 | |
| デコメ絵文字 PATH | path | | ○ | TEXT | デコメ絵文字コンテンツ絶対パス | 初期値 | 初期値 | 初期値 | 値を保持 | |
| カテゴリ ID | category_id | | ○ | INTEGER | category_info テーブルのカテゴリ ID | 初期値 ※ 1 | 初期値 ※ 1 | 初期値 | 値を保持 | |
| ディレクトリ ID | directory_id | | ○ | INTEGER | category_info テーブルのディレクトリ ID | 初期値 ※ 1 | 初期値 ※ 1 | 初期値 | 値を保持 | |
| 更新タイムスタンプ | timestamp | | ○ | INTEGER | DB 登録日時 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 利用フラグ | use_flag | | ○ | INTEGER | 1 | 初期値 | 初期値 | 初期値 | 0 | 有効レコード : 1 無効レコード : 0 |

| | | | | | | | | | | |
|------------------|-------------------|--|---|---------|---|------|-----|-------|------|--|
| ファイルサイズ | file_size | | | INTEGER | コンテンツのファイルサイズ | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 拡張タグ名 | tages | | | TEXT | 拡張タグ名 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 幅 | height | | ○ | INTEGER | コンテンツの横幅 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 高さ | width | | ○ | INTEGER | コンテンツの縦幅 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 絵文字種別 | kind | | ○ | INTEGER | デコレーション絵文字 : 1 デコレーション絵文字 (正) : 2 デコレーション絵文字 (長) : 3 デコメピクチャ : 4 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 最終利用表示順 | last_use_cnt | | ○ | INTEGER | 0 | 値を保持 | 初期値 | 最大値+1 | 値を保持 | |
| 利用履歴表示順 | history_cnt | | ○ | INTEGER | 0 | 値を保持 | 初期値 | 初期値 | 値を保持 | |
| デコメ絵文字 pop 識別フラグ | decome_pop_flag | | ○ | INTEGER | デコメ絵文字 pop:1 デコメ絵文字 pop 以外 : 0 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| ファイル最終更新日時 | file_lastmodified | | ○ | INTEGER | 実ファイルの更新日時 | 初期値 | 初期値 | 初期値 | 値を保持 | |

※1 初期値は、自動割当されるため、同じ値にならない場合がある。

9.2.2. デコメ絵文字辞書テーブル[decoemoji_dic]

| 論理名称 | 物理名称 | 主キー | 必須 | 型 | 初期値 | DB 変更契機 | | | | 備考 |
|-----------|----------------|-----|----|---------|----------------------------------|---------|-------|---------|---------|----|
| | | | | | 絵文字マネージャ起動時に格納されているコンテンツの DB 登録値 | 再構築 | 初期化 | コンテンツ追加 | コンテンツ削除 | |
| デコメ絵文字 ID | decoemoji_id | | ○ | INTEGER | decoemoji_info テーブルのデコメ絵文字 ID | 初期値※1 | 初期値※1 | 初期値※1 | 値を保持 | |
| 読み | decoemoji_name | | ○ | TEXT | コンテンツの読み | 初期値 | 初期値 | 初期値 | 値を保持 | |
| 品詞 | decoemoji_part | | ○ | INTEGER | 読みの品詞情報 | 初期値 | 初期値 | 初期値 | 値を保持 | ※2 |
| 表記 | decoemoji_note | | | TEXT | コンテンツの表記 | 初期値 | 初期値 | 初期値 | 値を保持 | |

※1 初期値は、自動割当されるため、同じ値にならない場合がある。

※2 品詞指定なし : 0

その他品詞については、「9.1.10 日本語向け品詞一覧」を参照。

9.2.3. カテゴリ情報テーブル[category_info]

| 論理名称 | 物理名称 | 主キー | 必須 | 型 | 初期値 | DB 変更契機 | | | | 備考 |
|----------------|-------------------|-----|----|---------|---|------------|------------|---------|---------|----|
| | | | | | 絵文字マネージャ起動時に格納されているコンテンツの DB 登録値 | 再構築 | 初期化 | コンテンツ追加 | コンテンツ削除 | |
| カテゴリ ID | category_id | ○ | ○ | INTEGER | 0 より順番に割り当てられる | 初期値 ※ 1 | 初期値 ※ 1 | 初期値※ 1 | 値を保持 | |
| カテゴリ名 (日本語) | category_name_jpn | | ○ | TEXT | サブフォルダ毎に割り当てられる日本語カテゴリ名 通常はフォルダ名がそのままカテゴリ名となるが、プリセットデータに指定されている場合は、設定されているカテゴリ名となる | 初期値 | 初期値 | 初期値 | 値を保持 | |
| カテゴリ名 (英語) | category_name_eng | | ○ | TEXT | サブフォルダ毎に割り当てられる英語カテゴリ名 通常はフォルダ名がそのままカテゴリ名となるが、プリセットデータに指定されている場合は、設定されているカテゴリ名となる | 初期値 | 初期値 | 初期値 | 値を保持 | |
| カテゴリプリセット ID | category_reset_id | | ○ | INTEGER | プリセットデータに指定されているカテゴリの Index が割り当てられる (0 オリジン) プリセットデータに指定されていないカテゴリは一律プリセットカテゴリ数 +1 の ID となる | 初期値 | 初期値 | 初期値 | 値を保持 | |

※ 1 初期値は、自動割当されるため、同じ値にならない場合がある。

9.2.4. ディレクトリ情報テーブル[`directory_info`]

| 論理名称 | 物理名称 | 主キー | 必須 | 型 | 初期値 | DB 変更契機 | | | | 備考 |
|-----------|-----------------------------|-----|----|---------|---|------------|------------|---------|---------|----|
| | | | | | 絵文字マネージャ起動時に格納されているコンテンツの DB 登録値 | 再構築 | 初期化 | コンテンツ追加 | コンテンツ削除 | |
| ディレクトリ ID | <code>directory_id</code> | ○ | ○ | INTEGER | 0 より順番に割り当てられる | 初期値 ※ 1 | 初期値 ※ 1 | 初期値※ 1 | 値を保持 | |
| カテゴリ ID | <code>category_id</code> | | ○ | INTEGER | <code>category_info</code> テーブルの <code>category_id</code> | 初期値 ※ 1 | 初期値 ※ 1 | 初期値※ 1 | 値を保持 | |
| ディレクトリ名 | <code>directory_name</code> | | ○ | TEXT | ディレクトリ名 | 初期値 | 初期値 | 初期値 | 値を保持 | |

※ 1 初期値は、自動割当されるため、同じ値にならない場合がある。

9.2.5. カスタムタグ情報テーブル[`customtag_info`]

| 論理名称 | 物理名称 | 主キー | 必須 | 型 | 初期値 | DB 変更契機 | | | | 備考 |
|-----------|--------------------------------|-----|----|---------|--|------------|------------|---------|---------|----|
| | | | | | 絵文字マネージャ起動時に格納されているコンテンツの DB 登録値 | 再構築 | 初期化 | コンテンツ追加 | コンテンツ削除 | |
| デコメ絵文字 ID | <code>decoemoji_id</code> | | ○ | INTEGER | 0 より順番に割り当てられる | 初期値 ※ 1 | 初期値 ※ 1 | 初期値※ 1 | 値を保持 | |
| カスタムタグ ID | <code>customtag_id</code> | | ○ | INTEGER | デコ絵文字 ID 毎にタグの格納順で 0 より順番に割り当てられる (0 オリジン) | 初期値 | 初期値 | 初期値 | 値を保持 | |
| カスタムタグ名 | <code>customtag_name</code> | | ○ | TEXT | カスタムタグ名 | 初期値 | 初期値 | 初期値 | 値を保持 | |
| カスタムタグ要素 | <code>customtag_element</code> | | ○ | TEXT | カスタムタグ要素 | 初期値 | 初期値 | 初期値 | 値を保持 | |

※ 1 初期値は、自動割当されるため、同じ値にならない場合がある。

10. 商標について

- Android™ は Google Inc の商標です。
- デコメ、デコメピクチャ、デコメ絵文字 pop は株式会社 NTT ドコモの登録商標です。

| 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 |
| V1.4.0 | 2013/9/30 | 絵文字パッケージ V1.4 対応 |
| V1.6.0 | 2014/8/19 | 絵文字パッケージ V1.6 対応(Android V4.4 KitKat 対応)。 キャリア絵文字の Unicode 化、OS によるカラーフォント対応に伴い、絵文字パッケージからキャリア絵文字対応機能を削除し、デコメ絵文字対応機能に特化した。 |
| V1.9.0 | 2016/10/11 | 絵文字パッケージ V1.9 対応(Android V7.0 Nougat 対応)。 |
| V2.0.0 | 2016/10/12 | 絵文字パッケージ V1.9 対応(Android V7.0 Nougat 対応)。 文言やサンプルコードの修正。 |
| V3.0.0 | 2020/12/04 | 絵文字パッケージ V1.13.1 対応(Android V11.0 対応)。 |