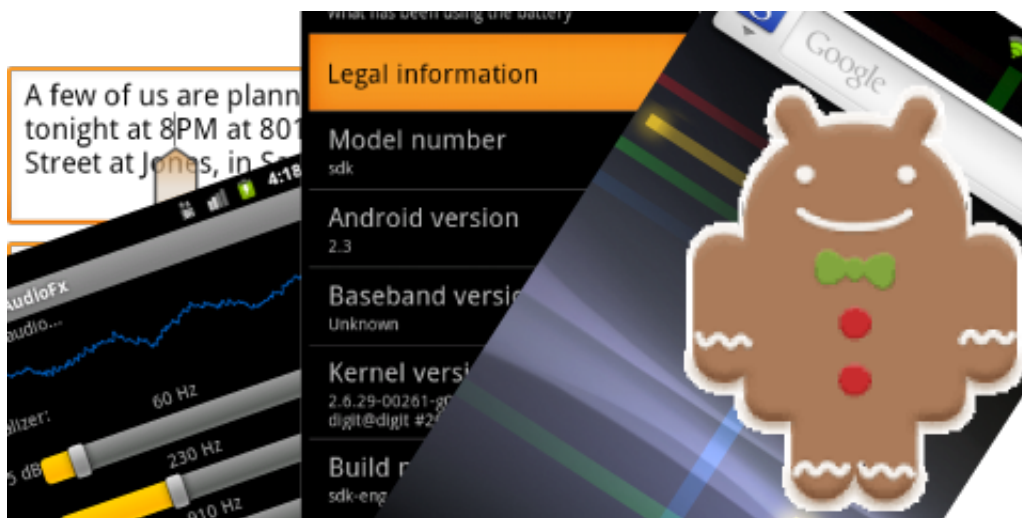


Android SDK 2.3(GingerBread)リリースまとめ(API編)



ついに待ちに待ったAndroid SDK 2.3のGingerBreadがリリースされたので、早速その新機能をまとめてみました。

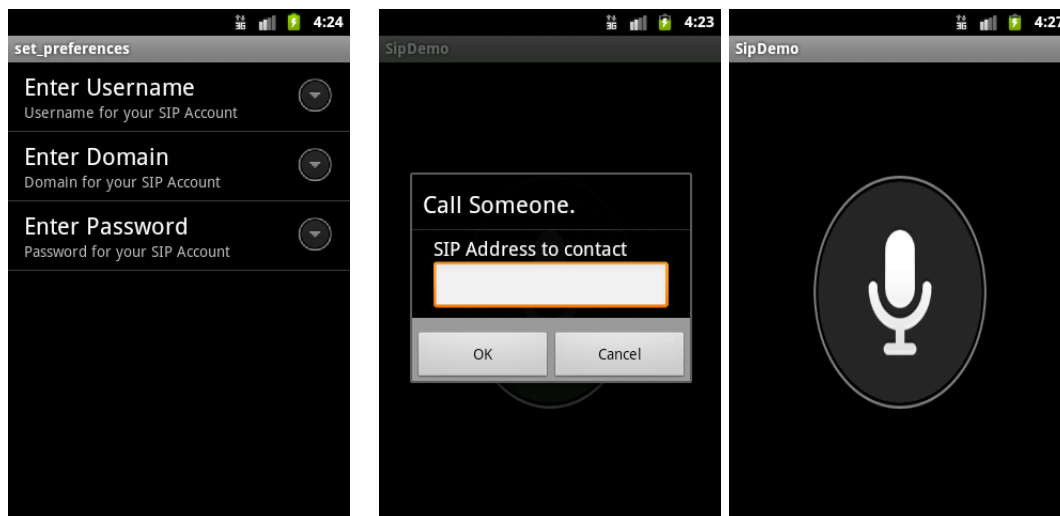
今回は見た目の変更が多いという噂でしたが、そうでもなかったようです。しかし、今までにないくらい新機能が aumentato したように思います。

まずは2.3で追加された新APIを見ていきましょう。

Android SDK 2.3からのAPI

SIPベースのVoIP

フレームワークにSIPプロトコルスタックが搭載され、インターネット電話を扱えるようになりました。SIPのサンプルアプリケーションは [ココ](#) にあります。以下は実際にSipDemoを動かしてみたところ。



NFC

NFC(Near Field Communications)のAPIが追加され、ハードウェアが搭載されていれば、

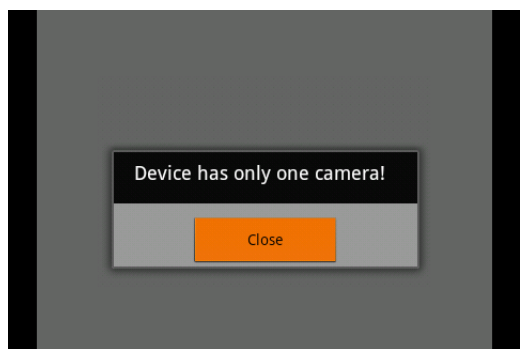
FeliCaのように近距離通信を行うことができるようになりました。NFCのサンプルコードは[ココ](#)です。現時点ではまだNFCのハードウェアが搭載されたAndroid 2.3の端末がないため、試せません。エミュレータで実行すると、当然のようにExceptionが発生してしまいました。

ジャイロスコープとその他のセンサー

ジャイロスコープ、回転速度、直線加速、重力、バロメーターのセンサーが増え、センサーAPIから使用できるようになりました。センサーAPIの詳細は[ココ](#)です。

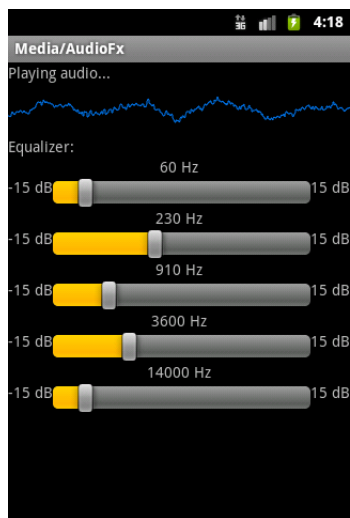
複数のカメラのサポート

アプリケーションから複数のカメラが利用できるようになりました。フロントカメラを利用するサンプルコードは[ココ](#)です。APIDemoのCameraPreviewを試してみましたが、「カメラが一個しか搭載されてないよ！」と怒られてしまいました。



オーディオ効果のミキシング

オーディオにバスブーストやリバーブなどの様々なエフェクトを掛けることのできるAPIが追加されました。オーディオエフェクトのサンプルコードは[ココ](#)です。以下は実際にAPIDemoを動かしてみたところ。



その他のメディア関連の新機能

JPEG画像のEXIF情報の中のaltitude(高度)の値が取得できるようになりました。また、ビデ

オ撮影中のオリエンテーション情報を [setOrientationHint\(\)](#) メソッドで [MediaRecorder](#) に通知することができるようになりました。

ダウンロードマネージャ

[DownloadManager](#) が追加され、長期間動作する HTTP ダウンロードをアプリケーションから要求することができるようになりました。

制限モード

[StrictMode](#) と呼ばれる制限モードが追加されました。ANR を発生させないために、UI スレッドで時間のかかる処理を使った場合に検知してくれるモードです。検知した場合は Handler、AysncTask、IntentService などを使って回避するようにコードを書き換えることを推奨します。

UI フレームワーク

overscroll のサポート

限界までスクロールしても、余分にスクロールさせることができるようになりました。余分にスクロールさせる量などは制御することができます。

タッチフィルタリング

マルウェアなどから購入ボタンやパーミッション許可を与えるボタンなどを押させることがないように、タッチフィルタリングを使ってセキュリティを強化することができますようになりました。サンプルコードは [ココ](#) です。

イベント管理の改善

[InputEvent](#) が新しく追加されました。

モーションイベントの改善

[MotionEvent](#) にポインタ ID が追加されました。それによって、アプリケーションから個々の指の動きを捕捉することができるようになりました。

テキスト選択コントロール

[setComposingRegion](#) メソッドにより、アプリケーションから編集集中のテキスト領域を設定できるようになりました。TextView に新しい属性も追加されました。

アクティビティの制御

[ActivityInfo](#) クラスにアクティビティの向きを管理する定数が追加されました。画面の回転に合わせてすべての向きに画面表示を合わせることが可能になったようです。

ノティファケーションのテキストとアイコンスタイル

ノティファケーションスタイルを管理する新しい属性が追加されました。

WebView

[setUseWebViewBackgroundForOverscrollBackground\(\)](#)メソッドを使ってoverscrollの背景を指定できるようになりました。

超大型スクリーン

タブレットデバイスなどに見られる超大型スクリーンに対応しました。超大型スクリーンの対応方法は[ココ](#)を参照してください。

グラフィック

OpenGL ES 2.0の残りのglDrawElements()、glVertexAttribPointer()メソッドに対応しました。また、YV12の画像フォーマットに対応しました。

コンテンツプロバイダ

- [AlarmClock](#) プロバイダが追加されました。
- [MediaStore](#) が [PLAY FROM SEARCH](#) の新しいインテントアクションに対応しました。検索されると、自動的に音楽が再生されるインテントです。
- [MediaStore](#) が新しい [MEDIA IGNORE FILENAME](#) フラグによって無視するファイル名を指定できるようになりました。
- [Settings](#) プロバイダに [APPLICATION DETAILS SETTINGS](#) と [MANAGE ALL APPLICATIONS SETTINGS](#) のアクションが増えました。前者はアプリケーションの詳細設定画面、後者はアプリケーション管理画面となります。
- [ContactsContract](#) プロバイダにSIPアドレスを格納するための [ContactsContract.CommonDataKinds.SipAddress](#) が追加されました。

Location

- [LocationManager](#) はシステムに管理された [WorkSource](#) クラスにしたがって、wake locks と wifi locks を要求するアプリケーションを追跡し、バッテリー使用量のパラメータに追加するようになりました。
- [LocationManager](#) に周期的または1回だけ位置情報を受け取る新しいメソッドが追加されました。

ストレージ

OBB (Opaque Binary Blob)をサポートする [StorageManager](#) が新しく追加されました。2011年の初頭までにAndroid 2.3の開発ツールでOBBファイルを管理することができるようになる予定です。

パッケージマネージャ

[PackageInfo](#) クラスに [firstInstallTime](#) と [lastUpdateTime](#) が追加されました。前者はパッケージのインストール日時で、後者は更新日時です。また、新しい [getProviderInfo\(\)](#) メソッドで独自のコンテンツプロバイダクラスの情報も取得できるようになりました。

Telephony

[TelephonyManager](#) にCDMA EVDO Rev Bのネットワークタイプのため [NETWORK_TYPE_EVDO_B](#) の定数が追加されました。また、新しい [getPsc\(\)](#) メソッドを使用してUMTSネットワーク上のスクランブルコードを取得できるようになりました。

アクティビティ、ウィンドウのライフサイクルのネイティブアクセス

- 新しい [NativeActivity](#) クラスを使ってネイティブコードにライフサイクルのコールバックを直接実装できるようになりました。
- [InputQueue](#) クラスのコールバックインターフェースでネイティブコードからイベントキューを管理できるようになりました。
- [SurfaceHolder.Callback2](#) インターフェースでネイティブコードから [SurfaceHolder](#) を管理できるようになりました。
- Windowクラスの新しいメソッドの [takeInputQueue](#) と [takeSurface\(\)](#) メソッドでネイティブコードからイベントとサーフェスを管理できるようになりました。
- NativeActivityについては [ココ](#) に詳細があります。

Dalvikランタイム

- dalvik.systemから、いくつかの非推奨クラスを削除
- 新しいコレクションの追加。 [ArrayDeque](#), [NavigableMap](#), [ConcurrentSkipListMap](#), [LinkedBlockingDeque](#)
- 新しいArraysユーティリティの追加。 [binarySearch\(\)](#), [copyOf\(\)](#), [copyOfRange\(\)](#), 他。
- [HttpURLConnection](#) のための [CookieManager](#) の追加。

- もっと下回りをカバーしたネットワークAPIの追加。 [InterfaceAddress](#), [NetworkInterface](#), [IDN](#)
- [File](#)クラスの読み書きアクセスメソッドの追加。
- [String.isEmpty\(\)](#)クラスのサポート。
- [Normalizer](#)と [Normalizer.Form](#)クラスのサポート。
- [javax.net.ssl](#)のサーバソケットの改善。

マニフェストの新しい要素と属性

- [supports-screens](#)要素への [xlargeScreens](#)属性の追加。超大型スクリーンをサポートするための物。
- <activity>要素への [android:screenOrientation](#)属性の値の追加。
 - [reverseLandscape](#)
 - アクティビティが通常のランドスケープモードの上下反対の画面を持つ。
 - [reversePortrait](#)
 - アクティビティが通常のポートレートモードの上下反対の画面を持つ。
 - [sensorLandscape](#)
 - アクティビティがランドスケープモード画面を持つが、センサーの検知により、持っている向きに合わせて画面の向きが入れ替わる。
 - [sensorPortrait](#)
 - アクティビティがポートレートモード画面を持つが、センサーの検知により、持っている向きに合わせて画面の向きが入れ替わる。
 - [fullSensor](#)
 - アクティビティがユーザーが動かした端末の向きに合わせて画面の向きが入れ替わる。

新しいパーミッション

- `com.android.permission.SET_ALARM`

- アプリケーションから [SET_ALARM](#) のインテントを broadcast できるようになる。
- android.permission.USE_SIP
 - アプリケーションから、[SIP API](#) を使ってインターネット電話を使えるようにする。
- android.permission.NFC
 - アプリケーションから、[NFC API](#) を使えるようにする。

新機能の定数

- [android.hardware.audio.low_latency](#)
 - アプリケーションは低待ち時間のオーディオパイプラインを利用して、音声の入出力を細かく扱います。
- [android.hardware.camera.front](#)
 - アプリケーションはフロントカメラを使います。
- [android.hardware.nfc](#)
 - アプリケーションはNFC機能を使います。
- [android.hardware.sensor.barometer](#)
 - アプリケーションは端末のバロメーターを使います。
- [android.hardware.sensor.gyroscope](#)
 - アプリケーションはジャイロスコープセンサーを使います。
- [android.software.sip](#)
 - アプリケーションはSIP APIを使っています。
- [android.software.sip.voip](#)
 - アプリケーションはSIPベースのVOIPサービスを使います。
- [android.hardware.touchscreen.multitouch.jazzhand](#)
 - アプリケーションは進歩したマルチポイントのマルチタッチ性能を使って、5点まで独立して追跡することができます。
- 詳細は [uses-feature](#) をご覧ください。

APIの新機能をひと列挙してみました。かなりの変更量でしたね。

API Level 8からの差分はココにまとまっています。

API以外にも変わっている部分があります。

引き続き、ハイライト編でお伝えします。