

# Rolto SDK for Android仕様書

---

Version 1.0

## 更新履歴

Version	日付	改訂内容
1.0	2015/3	初版作成

## 目次

はじめに(概要) .....	5
仕様 .....	5
仕様 .....	5
組み込み方法 .....	6
ライブラリーとヘッダーの追加 .....	6
技術概要 .....	6
クラス構成 .....	6
API一覧 .....	6
APIリファレンス .....	7
RoltoPrintクラスリファレンス .....	7
getModelIdentifier .....	7
getPrinterName .....	7
performPrintImage .....	8
performPrintImagePaths .....	9
performPrintImageUri .....	10
cancelPrint .....	12
requestPrinterStatus .....	12
RoltoPrint.PrintResultCallbackクラスリファレンス .....	13
onPrintComplete .....	13
RoltoPrint.PrinterStatusCallbackクラスリファレンス .....	14
onReceivePrinterStatus .....	14
RoltoPrintDiscoverクラスリファレンス .....	15
searchPrinters .....	15
RoltoPrintDiscover.OnFindPrinterCallbackクラスリファレンス .....	16
onFindPrinter .....	16
RoltoPrintDiscover.SearchCompleteCallbackクラスリファレンス .....	16
onPrinterSearchComplete .....	16
RoltoPrint Constant .....	17
PrintResult .....	17
PrinterStatus .....	18
PrintParamKey .....	19

---

サンプルアプリケーション .....	20
実装方法 .....	20
画面と機能 .....	20
印刷結果例 .....	21
サンプルの設定について .....	21

## 1.はじめに(概要)

本仕様書はAndroid上からロルト PT10に印刷するための機能を提供するRolto SDKの説明、リファレンスを記述します。

「キングジム」、「KING JIM」、「ロルト」、「Rolto」は、株式会社キングジムの登録商標または商標です。

その他記載の会社名および商品名は、各社の登録商標または商標です。

使用者は、エンドユーザーが認識できるよう、開発アプリケーション上に「Powered by KING JIM Rolto SDK」を記載するほか、同部分に“<http://www.kingjim.co.jp/support/rolto/sdk>へのリンクを埋め込まなければなりません。

### フォルダー構成

本ライブラリーのフォルダー構成は以下のようになっています。

Rolto SDK		
libs	RoltoLib.jar	
	armeabi	libRoltoCore.so
	armeabi-v7a	libRoltoCore.so
	Documents	
	Rolto SDK for Android仕様書.pdf	
Sample	RoltoPrintSample.zip	
	README.txt	

## 2.仕様

### 2.1. 仕様

項目	説明
開発環境	JDK6 eclipse 4.2 以降 Android SDK API 14 以降
対応デバイス	Android 4.0 以降 すべての Android 4.0 以降のデバイスでの印刷を保証するものではありません。
開発言語	Java
対応プリンター（ロルト）	PT10
対応インターフェイス	無線LAN:IEEE802.11b/g/n
印刷幅	約48mm(203dpi / 384dpt)
印刷長	最大約4,000mm(203dpi / 32,000dot)
対応紙	「ロルト」専用ロール紙50mm幅

### 3. 組み込み方法

#### 3.1. ライブラリーとヘッダーの追加

本SDKのLibsフォルダを、作成したプロジェクトフォルダにコピーします。

実装方法につきましては、サンプルプロジェクトをご参照ください。

### 4. 技術概要

#### 4.1. クラス構成

Rolto SDKは下記のクラスで構成されます。

RoltoPrint: 本体管理 (印刷、ステータス、本体制御)

RoltoPrintDiscover: 検索

### 5. API一覧

Class	API	機能
<a href="#">RoltoPrint</a>	<a href="#">getModelIdentifier</a>	ロルトの識別情報
	<a href="#">getPrinterName</a>	ロルトに設定されている名前を返します
	<a href="#">performPrintImage</a>	イメージの印刷を実行します
	<a href="#">performPrintImagePaths</a>	パスで指定したイメージの印刷を実行します
	<a href="#">performPrintImageUri</a>	Uriで指定したイメージの印刷を実行します
	<a href="#">cancelPrint</a>	印刷の中止をします
	<a href="#">requestPrinterStatus</a>	ロルトのステータスを取得します
	<a href="#">PrintResultCallback.onPrintComplete</a>	プリントが終了した場合に呼び出されます
	<a href="#">PrinterStatusCallback.onReceivePrinterStatus</a>	ステータスを取得した場合に呼び出されます
<a href="#">RoltoPrintDiscover</a>	<a href="#">searchPrinters</a>	ロルトを検索します
	<a href="#">OnFindPrinterCallback.onFindPrinter</a>	検索結果を返します
	<a href="#">SearchCompleteCallback.onPrinterSearchComplete</a>	ロルト検索が終了した場合に呼び出されます

## 6.APIリファレンス

### RoltoPrintクラスリファレンス

RoltoPrintクラスはプリンターデータ生成及びデータ送信を行うクラスです。

#### getModelIdentifier

ロルトの識別情報を返します。

```
String getModelIdentifier()
```

リターン

ロルトの識別情報。

#### getPrinterName

ロルトに設定されている名前を返します。

```
String getPrinterName()
```

リターン

ロルトに設定されている名前を返します。

## performPrintImage

イメージの印刷を実行します。

```
void performPrintImage(  
    Bitmap bitmap,  
    Map<PrintParamKey, Object> params,  
    PrintResultCallback callback)
```

### パラメーター

bitmap

印刷するイメージ。

params

印刷パラメーターを指定します。

[PrintParamKey](#)を参照。

callback

印刷終了のコールバック。

## performPrintImagePath

パスのイメージを印刷します。

```
void performPrintImagePath(  
    List<String> imagePath,  
    Map<PrintParamKey, Object> params,  
    PrintResultCallback callback)
```

### パラメーター

imagePaths

イメージのパスの配列。

params

印刷パラメーターを指定します。

[PrintParamKey](#)を参照。

callback

印刷終了のコールバック。

## performPrintImageUri

Uriのイメージを印刷します。

```
void performPrintImageUri(  
    Context context,  
    List<Uri> imageUri,  
    Map<PrintParamKey, Object> params,  
    PrintResultCallback callback)
```

### パラメーター

context

コンテキスト。

imageUri

イメージのUriの配列。

params

印刷パラメーターを指定します。

[PrintParamKey](#)を参照。

callback

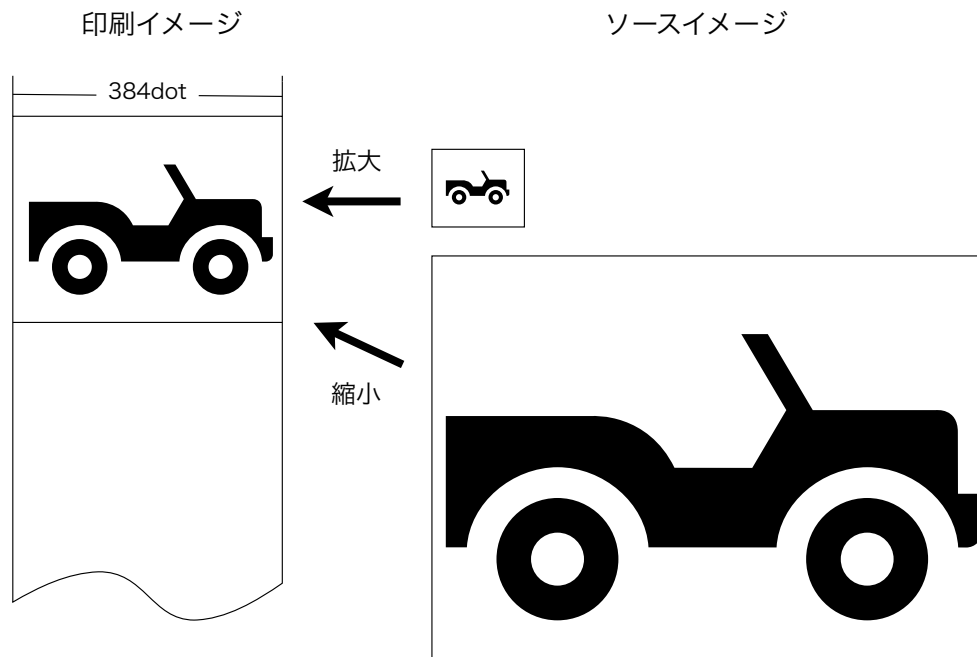
印刷終了のコールバック。

## イメージの描画について

イメージはアスペクトフィットにて描画します。

縦横比を保持したまま、幅が384dotになるように、拡大または縮小し印刷します。

拡大縮小させたくない場合は、幅384dotのイメージを用意してください。



## cancelPrint

印刷を中止します。

```
void cancelPrint()
```

## requestPrinterStatus

ステータスから本体の情報を取得します。

```
void requestPrinterStatus(final PrinterStatusCallback callback)
```

### パラメーター

callback

ステータス取得用コールバック。

[PrinterStatus](#)を返します。

---

## RoltoPrint.PrintResultCallbackクラスリファレンス

### onPrintComplete

プリントが終了した場合に呼び出されます。

```
void onPrintComplete(PrintResult result)
```

#### パラメーター

**result**

印刷終了時の結果を示す[PrintResult](#)を返します。

## RoltoPrint.PrinterStatusCallbackクラスリファレンス

### onReceivePrinterStatus

ステータスを取得した場合に呼び出されます。

```
void onReceivePrinterStatus(  
    PrinterStatus status,  
    boolean isDirectMode)
```

#### パラメーター

status

ステータスを示す[PrinterStatus](#)を返します。

isDirectMode

ダイレクトモードONの場合にTRUE。

## RoltoPrintDiscoverクラスリファレンス

RoltoPrintDiscoverクラスはロルト検索を行うクラスです。

### searchPrinters

ロルトを検索します。

```
void searchPrinters(  
    final OnFindPrinterCallback onFindCallback,  
    final SearchCompleteCallback completeCallback,  
    final int duration)
```

#### パラメーター

onFindCallback

ロルトが見つかった時のコールバック。

completeCallback

ロルト検索が終了した時のコールバック。

duration

検索の秒数を定義します。

## RoltoPrintDiscover.OnFindPrinterCallbackクラスリファレンス

### onFindPrinter

検索されたロルトのRoltoPrintクラスが返ります。

```
void onFindPrinter(RoltoPrint print)
```

#### パラメーター

print

[RoltoPrint](#)クラスを返します。

## RoltoPrintDiscover.SearchCompleteCallbackクラスリファレンス

### onPrinterSearchComplete

ロルト検索が終了した場合に呼び出されます。

```
void onPrinterSearchComplete()
```

## 7. RoltoPrint Constant

これ以降はConstantの説明になります。

### PrintResult

印刷結果。

```
enum PrintResult {  
    Success,  
    UserCancel,  
    CommunicationError,  
    PaperOutError,  
    OverheatError,  
    CoverOpenError,  
    PrinterBusyError,  
    UnexpectedError,  
}
```

Success

印刷成功。

UserCancel

ユーザーキャンセルによる印刷中止。

CommunicationError

通信エラー。

PaperOutError

用紙なしエラー。

OverheatError

オーバーヒートエラー。

CoverOpenError

カバーオープンエラー。

PrinterBusyError

使用中。

UnexpectedError

原因不明のエラー。



## PrinterStatus

ロルトのステータス。

```
enum PrinterStatus {  
    Idle,  
    Printing,  
    WritingFirmware,  
    CoverOpen,  
    PaperOut,  
    Overheated,  
}
```

Idle

印刷可能。

Printing

印刷中。

WritingFirmware

ファームウェア書き込み中。

CoverOpen

カバーオープン。

PaperOut

用紙なし。

Overheated

オーバーヒート中。



## PrintParamKey

[performPrintImage](#)と[performPrintImagePaths](#)と[performPrintImageUri](#)で設定する辞書のキーです。

```
enum PrintParamKey {  
    UsesErrorDiffusion,  
    BinarizationOffset,  
    BlockSize  
}
```

UsesErrorDiffusion

Boolean

YES : 誤差拡散

NO : 適応的二値化

※誤差拡散を指定した場合はBinarizationOffsetとBlockSizeは無視します。

BinarizationOffset

Double

適応的二値化の閾値

範囲:0.0 - 1.0

BlockSize

Double

適応的二値化のブロックサイズ

範囲:0.0 - 1.0

## 8. サンプルアプリケーション

### 8.1. 実装方法

サンプルアプリケーションでは次の実装方法を確認出来ます。

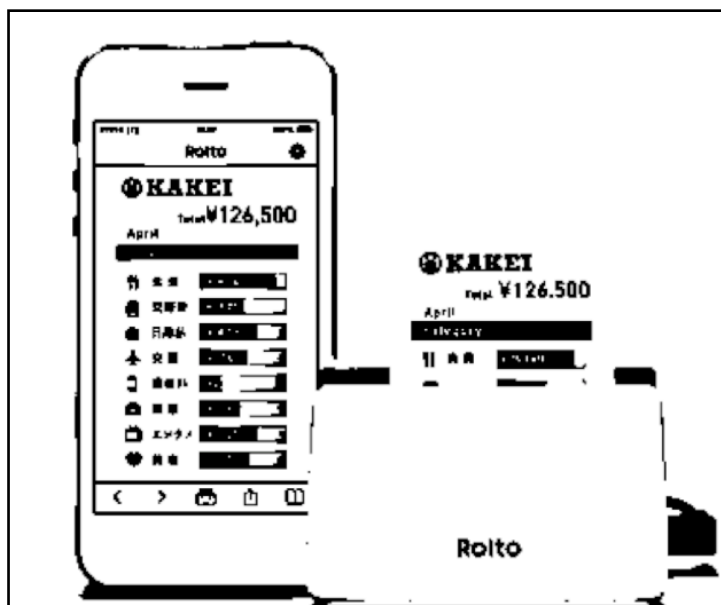
- ・ ロルト検索
- ・ 印刷
- ・ 印刷設定

### 8.2. 画面と機能



番号	内容
1	イメージ選択画面を表示し、選択したイメージを表示します
2	ロルト選択画面を表示します
3	ロルトのステータスを取得します
4	<a href="#">performPrintImageUri</a> を実行します
5	<a href="#">performPrintImage</a> を実行します

### 8.3. 印刷結果例



### 8.4. サンプルの設定について

サンプルは、下記の設定になっています。

#### <複数画像OFF時>

UsesErrorDiffusion: false

BinarizationOffset: 0.05

BlockSize: 0.79

[performPrintImage](#)を実行します

#### <複数画像ON時>

UsesErrorDiffusion: true

[performPrintImageUris](#)を実行します