

# TEPRA-Print SDK for Android 仕様書

Version 1.4

**更新履歴**

Version	日付	改訂内容
1.0	2016/03/04	初版作成
1.1	2020/02/19	<p>3.1.仕様</p> <p>開発環境の更新</p> <p>アーキテクチャーに armeabi-v7a/arm64-v8a/x86/x86_64 を追加</p> <p>対応プリンター（テプラ）に SR-R7900P/SR-MK1 を追加</p> <p>5.5.ビットマップ印刷</p> <p>ビットマップイメージの印刷シーケンスを追加</p> <p>6. API 一覧</p> <p>ラベルイメージデーターを取得する API を追加 (getLabelImages)</p> <p>テープ検出状態を取得する API を追加 (getTapeSwitchFromStatus)</p> <p>テープオプションを取得する API を追加 (getTapeOptionFromStatus)</p> <p>電圧状態を取得する API を追加 (getBatteryLevelFromStatus)</p> <p>バッテリー種類の設定を行う API を追加 (setBatteryType)</p> <p>バッテリー種類を取得する API を追加 (getBatteryType)</p> <p>小巻機接続状態を取得する API を追加 (isTapeWinder)</p> <p>EX ロングテープ情報を取得する API を追加 (isBigroll)</p> <p>EX ロングテープのテープ長を取得する API を追加 (bigrollTapeLengthFromStatus)</p> <p>EX ロングテープのテープ使用長を取得する API を追加 (bigrollTapeUsageLengthFromStatus)</p> <p>7. API リファレンス</p> <p>ラベルイメージデーターを取得する API を追加 (labelImages)</p> <p>ビットマップを印刷する API を追加 (doPrint)</p> <p>テープ検出状態を取得する API を追加 (tapeSwitchFromStatus)</p> <p>テープオプションを取得する API を追加 (tapeOptionFromStatus)</p> <p>電圧状態を取得する API を追加 (getBatteryLevelFromStatus)</p> <p>バッテリー種類の設定を行う API を追加 (setBatteryType)</p> <p>バッテリー種類を取得する API を追加 (getBatteryType)</p> <p>小巻機接続状態を取得する API を追加 (isTapeWinder)</p> <p>EX ロングテープ情報を取得する API を追加 (isBigroll)</p> <p>EX ロングテープのテープ長を取得する API を追加 (bigrollTapeLengthFromStatus)</p> <p>EX ロングテープのテープ使用長を取得する API を追加 (bigrollTapeUsageLengthFromStatus)</p> <p>テープ幅に 50mm テープを追加 (TepraPrintTapeWidth)</p> <p>テープ種類の内容を更新 (TepraPrintTapeKind)</p>

		<p>エラー状態の内容を更新 (TepraPrintStatusError)</p> <p>印刷設定を優先するキーを追加 (PrintParameter)</p> <p>印刷速度に中速を追加(PrintParameter)</p> <p>ハーフカット連続を指定するキーを追加 (PrintParameter)</p> <p>テープ検出状態辞書のキーを追加 (TapeSwitch)</p> <p>テープオプション辞書のキーを追加(TapeOption)</p> <p>ビットマップ印刷時の余白指定に対応 (doPrint)</p> <p>ビットマップ印刷時の複数ページ指定に対応 (doPrint)</p> <p>8. サンプルアプリケーション</p> <p>TepraPrintSimpleSample を追加</p> <p>9. その他</p> <p>9.2.ProGuard settings を追加</p>
1.2	2021/02/18	<p>6. API 一覧</p> <p>オブジェクト情報を設定する API を追加 (setObjectType)</p> <p>7. API リファレンス</p> <p>オブジェクト情報を設定する API を追加 (setObjectType)</p> <p>オブジェクト種類を追加 (TepraPrintObjectType)</p> <p>8. サンプルアプリケーション</p> <p>開発環境の更新</p>
1.3	2021/09/10	<p>3.1.仕様</p> <p>開発環境の更新</p> <p>対応プリンター（テプラ）に SR-R2500P を追加</p> <p>6. API 一覧</p> <p>テープの印字領域を取得する API を追加 (getPrintableSizeFromTape)</p> <p>7. API リファレンス</p> <p>テープの印字領域を取得する API を追加 (getPrintableSizeFromTape)</p> <p>8. サンプルアプリケーション</p> <p>開発環境の更新</p>
1.4	2023/04/25	<p>2. フォルダー構成</p> <p>jniLibs/armabi を削除</p> <p>3.1.仕様</p> <p>開発環境の更新</p> <p>対応 OS の更新</p> <p>対応デバイスの更新</p> <p>対応アーキテクチャーの更新</p> <p>4.2.AndroidManifest.xml 設定</p> <p>権限を更新</p> <p>8. サンプルアプリケーション</p> <p>開発環境の更新</p>

## 目次

<b>1. はじめに（概要）</b>	<b>7</b>
<b>2. フォルダー構成</b>	<b>7</b>
<b>3. 仕様</b>	<b>8</b>
3.1. 仕様	8
3.2. 注意事項・制限事項	8
<b>4. 組み込み方法</b>	<b>9</b>
4.1. ライブラリーの追加	9
4.2. AndroidManifest.xml 設定	9
<b>5. 技術概要</b>	<b>9</b>
5.1. クラス構成	9
5.2. テプラ検索	10
5.3. ステータス取得	11
5.4. 印刷方法	13
5.5. ビットマップ印刷	14
5.6. フォームデータ・コンテンツデータ	15
<b>6. API 一覧</b>	<b>16</b>
<b>7. API リファレンス</b>	<b>19</b>
TepraPrint クラスリファレンス	19
TepraPrint メソッド	19
TepraPrint	19
getVersion	19
setPrinterInformation	19
setObjectType	20
setCallback	20
getLabellImages	20
getPrintableSizeFromTape	21
doPrint	22
getProgressOfPrint	23
getPageNumberOfPrinting	23
cancelPrint	23
fetchPrinterStatus	23
doTapeFeed	23
resumeOfPrint	24
getTapeWidthFromStatus	24
getTapeKindFromStatus	24
getTapeSwitchFromStatus	25
getTapeOptionFromStatus	25
getDeviceErrorFromStatus	25

getBatteryLevelFromStatus .....	26
setBatteryType .....	26
getBatteryType .....	26
getKindOfTape .....	26
getModelName .....	27
getResolution .....	27
isSupportHalfCut .....	27
isPrintSpeedSupport .....	27
isTapeWinder .....	28
isBigroll .....	28
getBigrollTapeLengthFromStatus .....	28
getBigrollTapeUsageLengthFromStatus .....	29
<b>TepraPrint Constant .....</b>	<b>30</b>
TepraPrintTapeOperation .....	30
TepraPrintTapeWidth .....	30
TepraPrintTapeKind .....	33
TepraPrintPrintingPhase .....	39
TepraPrintConnectionStatus .....	40
TepraPrintStatusError .....	41
TepraPrintTapeCut .....	45
TepraPrintPrintSpeed .....	45
TepraPrintObjectType .....	46
TepraPrintBatteryLevel .....	47
TepraPrintBatteryType .....	47
TepraPrintParameterKey .....	48
TepraPrintStatusKey .....	50
TepraPrintTapeSwitchKey .....	50
TepraPrintTapeOptionKey .....	51
<b>TepraPrintCallback インターフェイスリファレンス .....</b>	<b>52</b>
TepraPrintCallback メソッド .....	52
onChangePrintOperationPhase .....	52
onSuspendPrintOperation .....	52
onAbortPrintOperation .....	53
onChangeTapeFeedOperationPhase .....	53
onAbortTapeFeedOperation .....	54
<b>TepraPrintDiscoverPrinter クラスリファレンス .....</b>	<b>55</b>
TepraPrintDiscoverPrinter メソッド .....	55
TepraPrintDiscoverPrinter .....	55
setCallback .....	56
startDiscover .....	56

stopDiscover .....	56
<b>TepraPrintDiscoverPrinter Constant.....</b>	<b>57</b>
TepraPrintDiscoverConnectionType .....	57
<b>TepraPrintDiscoverPrinterCallback インターフェイスリファレンス .....</b>	<b>58</b>
TepraPrintDiscoverPrinterCallback メソッド .....	58
onFindPrinter .....	58
onRemovePrinter .....	58
<b>TepraPrintDataProvider インターフェイスリファレンス .....</b>	<b>59</b>
TepraPrintDataProvider メソッド .....	59
startOfPrint .....	59
endOfPrint .....	59
startPage .....	59
endPage .....	59
getNumberOfPages.....	59
getFormdataForPage.....	60
getBitmapContentData .....	60
getStringContentData .....	61
<b>8. サンプルアプリケーション .....</b>	<b>62</b>
8.1. 概要 .....	62
8.2. 機能 - TepraPrintSample .....	63
8.3. 機能 - TepraPrintSimpleSample .....	68
<b>9. その他 .....</b>	<b>69</b>
9.1. Android 6.0 以降での動作 .....	69
9.2. ProGuard settings .....	70

## 1. はじめに（概要）

本仕様書は Android 上から KING JIM テプラに印刷するための機能を提供する TEPRA-Print SDK の説明、リファレンスを記述します。

「キングジム」、「KING JIM」、「テプラ」、「TEPRA」、「テプラ」PRO、P テープマーク、R テープマーク、EX ロングマークは、株式会社キングジムの商標または登録商標です。

その他記載されている会社名および製品名は、各社の商標または登録商標です。

## 2. フォルダ構成

本ライブラリーのフォルダ構成は以下のようになっています。

TEPRA-Print_SDK_Android	
libs	
	TepraPrint.jar
jniLibs	
armeabi-v7a	
	libTepraPrint.so
arm64-v8a	
	libTepraPrint.so
x86	
	libTepraPrint.so
x86_64	
	libTepraPrint.so
Documents	
	TEPRA-Print SDK for Android Specifications.pdf
	TEPRA-Print SDK Library Form Data Reference.pdf
	TEPRA-Print SDK Appendix.pdf
Samples	
	Conceptual Diagram Form Data.pdf
	TepraPrintSample.zip
	TepraPrintSimpleSample.zip
README_Android.txt	

### 3. 仕様

#### 3.1. 仕様

項目	説明
開発環境	Android Studio Dolphin 以降 Android SDK API 23 以降
対応 OS	Android 6.0 以降
開発言語	Java
対応デバイス	対応 OS を搭載した端末 すべての Android 6.0 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。
対応アーキテクチャー	armeabi-v7a, arm64-v8a, x86, x86_64
対応プリンター（テプラ）	SR5900P SR5500P SR-R7900P SR-MK1 SR-R2500P
対応インターフェイス	Wi-Fi Bluetooth

#### 3.2. 注意事項・制限事項

USB 接続にて印刷中は、Bluetooth 接続から印刷はできません。

## 4. 組み込み方法

### 4.1. ライブラリーの追加

本 SDK の libs フォルダーにある jar ファイルを作成したプロジェクトの app/libs フォルダーにコピーしてください。

本 SDK の jniLibs フォルダーを作成したプロジェクトの app/src/main フォルダーにコピーしてください。

### 4.2. AndroidManifest.xml 設定

以下のパラメーターを AndroidManifest.xml に設定してください。

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
    android:maxSdkVersion="30" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
    android:maxSdkVersion="32" />
<uses-permission android:name="android.permission.BLUETOOTH" android:maxSdkVersion="30" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"
    android:maxSdkVersion="30"/>
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"
    android:usesPermissionFlags="neverForLocation" tools:targetApi="s" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.NEARBY_WIFI_DEVICES"
    android:usesPermissionFlags="neverForLocation" />
```

## 5. 技術概要

### 5.1. クラス構成

TEPRA-Print SDK は下記のクラスで構成されます。

[TepraPrint](#) : 本体管理（印刷、ステータス、本体制御）

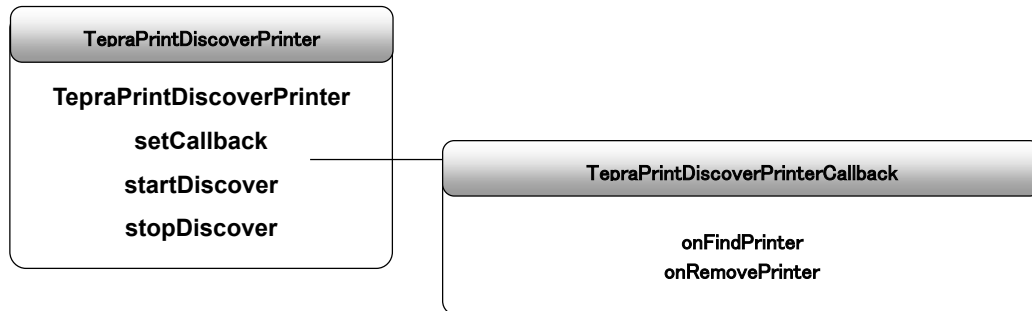
[TepraPrintDiscoverPrinter](#) : 検索

[TepraPrintDataProvider](#) : 描画用コンテンツ生成（アプリケーション実装用）

## 5.2. テプラ検索

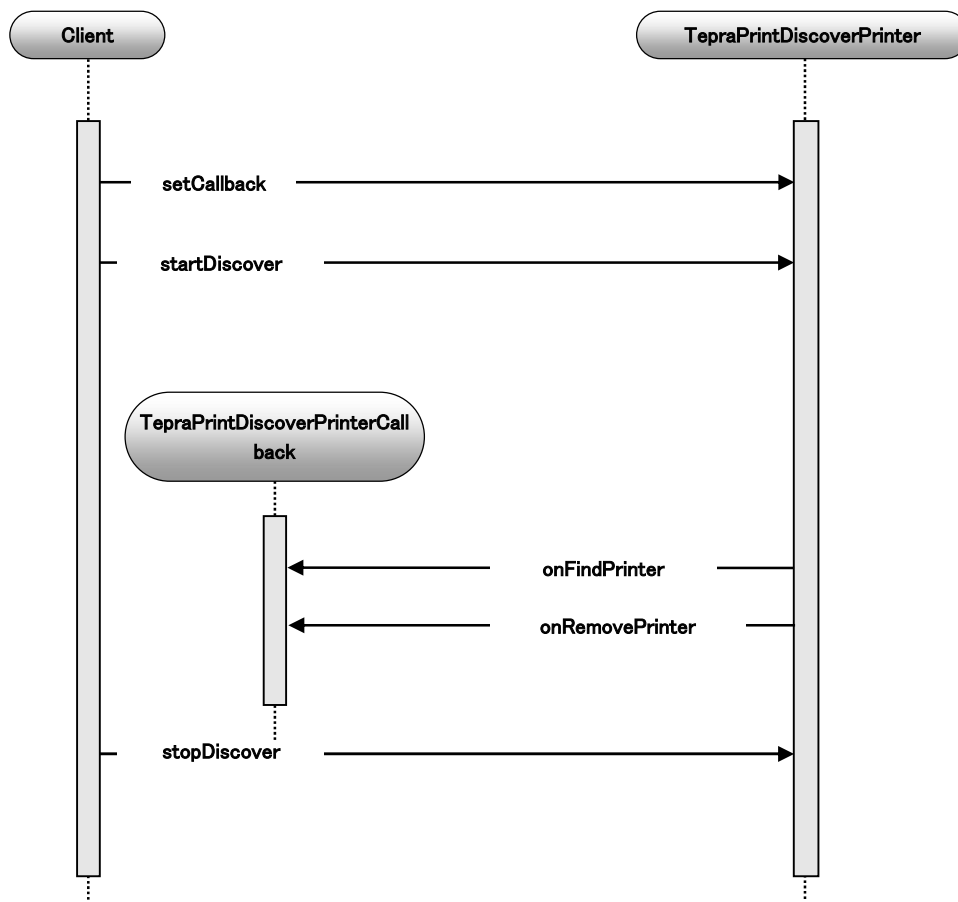
[TepraPrintDiscoverPrinter](#) により、特定のテプラのみ検索します。

Bonjour および Bluetooth でペアリングされたテプラが検索されます。



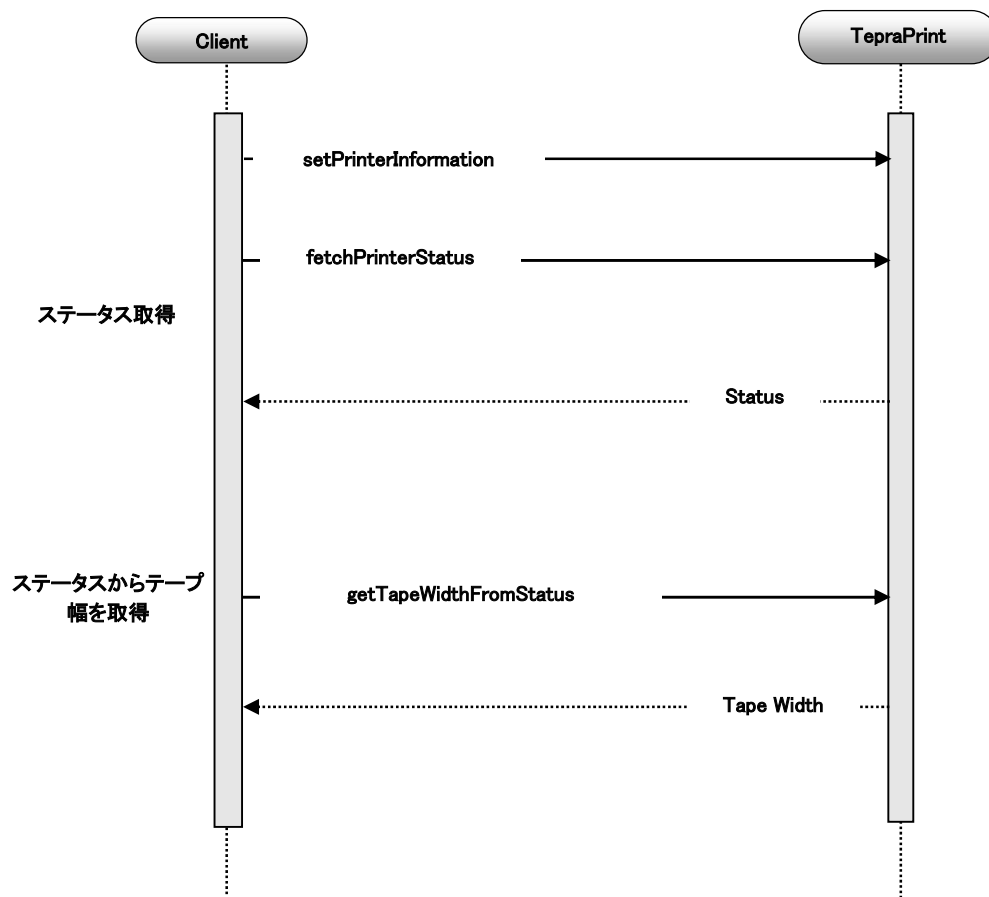
[TepraPrintDiscoverPrinter](#) クラスの [startDiscover](#) で検索が開始され、[stopDiscover](#) で検索を中止します。テプラが検索された場合、コールバックで検索されたテプラ情報が通知されます。テプラが Bonjour 上から削除された場合は、[onRemovePrinter](#) で削除されたことが通知されます。

ただし Bluetooth の場合は、Class of Device が 0x040680 に設定されている機器が [onFindPrinter](#) で通知され、[onRemovePrinter](#) は通知されません。



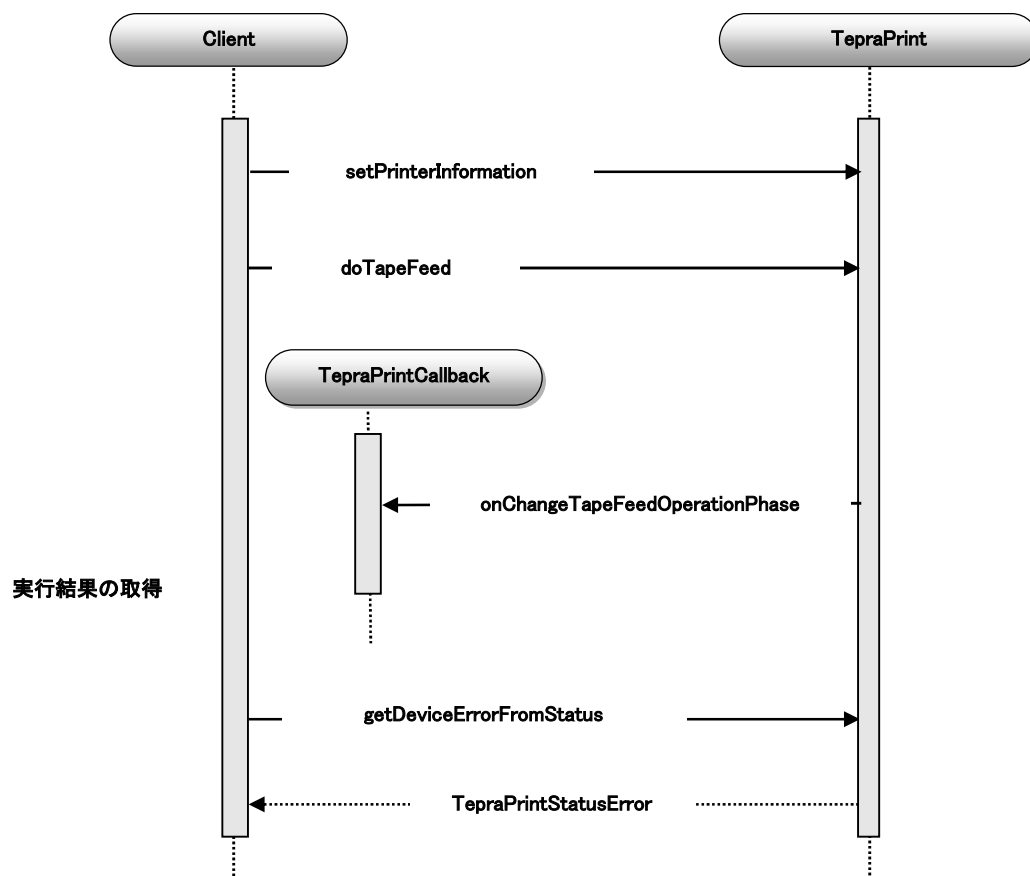
### 5.3. ステータス取得

[TepraPrint](#) クラスを生成し、[fetchPrinterStatus](#) でステータスを取得します。ステータス取得で返された情報（辞書）は情報取得用のメソッドで各情報が取得できます。



テープカット／テープ送りは [doTapeFeed](#) で行います。結果はステータスが返ります。

実行結果は [getDeviceErrorFromStatus](#) で行います。

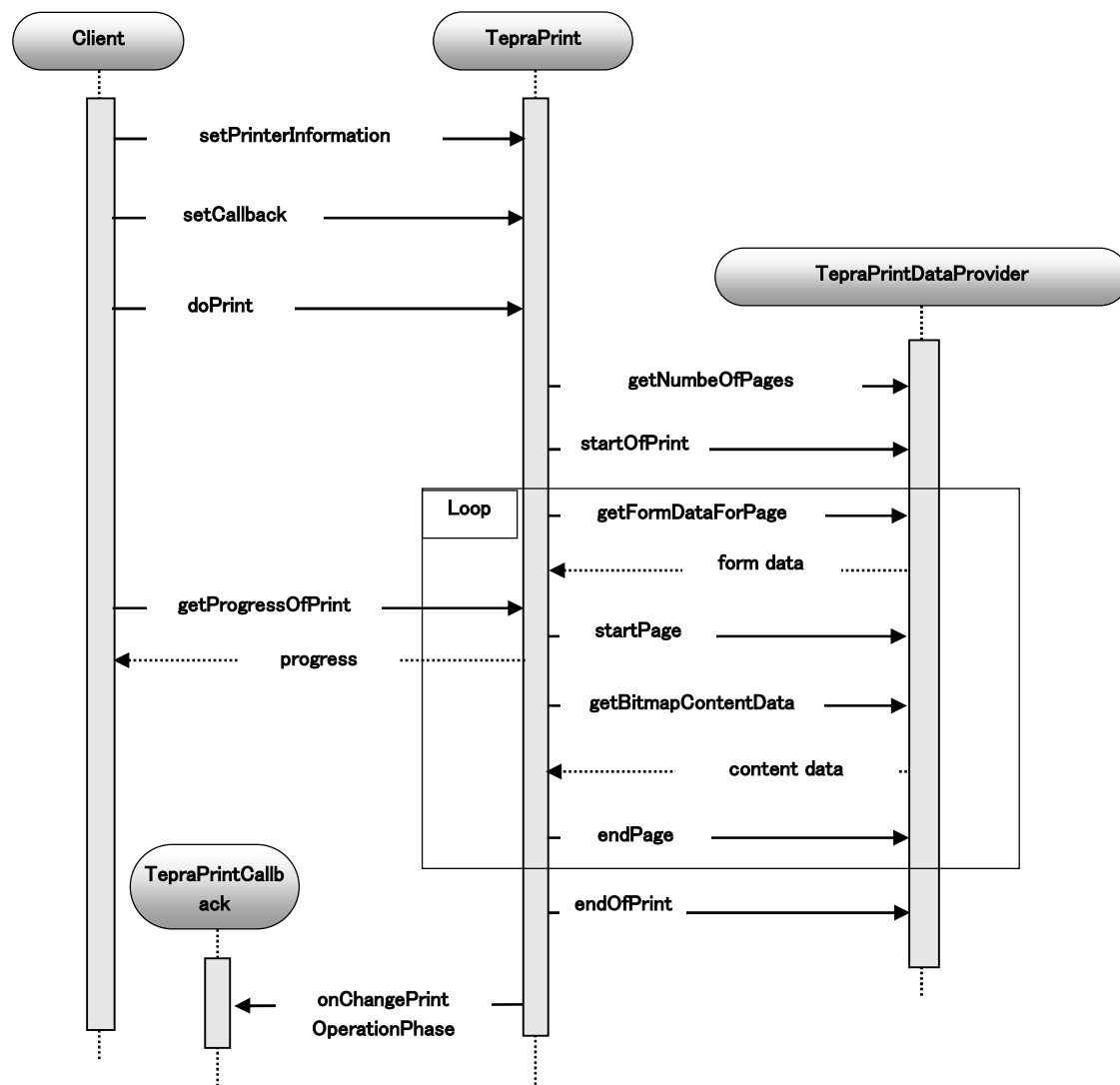


#### 5.4. 印刷方法

[TepraPrintDataProvider](#) を実装し各メソッドをオーバーライドして実現します。アプリケーションが [TepraPrint](#) クラスの [doPrint](#) を呼び出すとオーバーライドしたメソッドが呼び出されます。

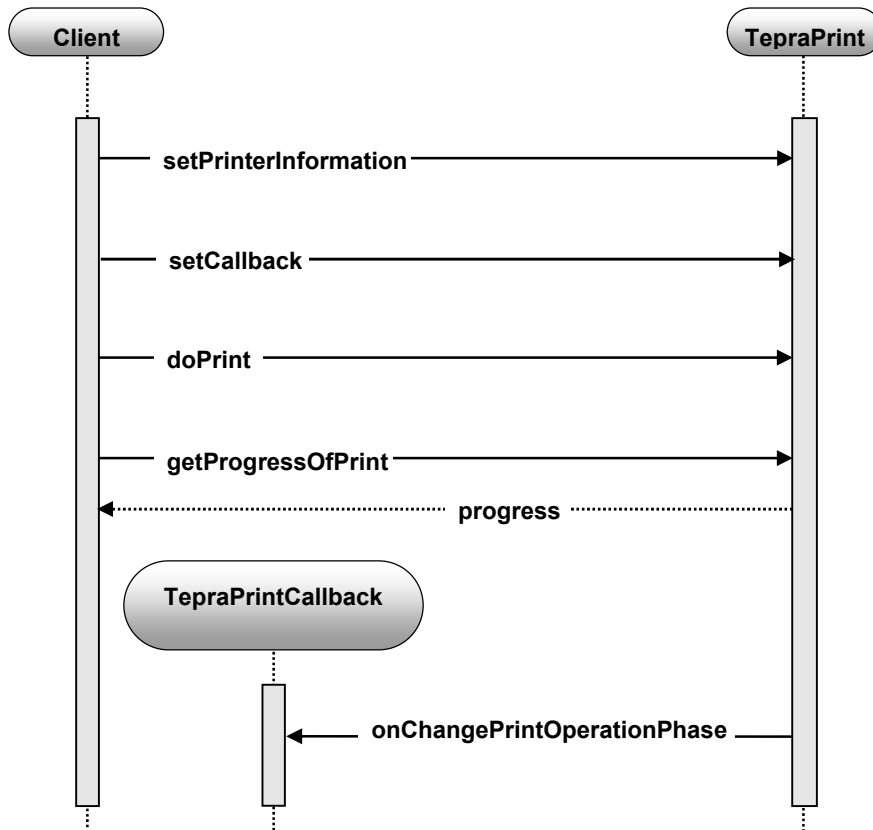
印刷ループ（部数印刷を含む）はアプリケーションで実装する必要はなく、各オーバーライドメソッドで必要な手続きを行います。

[startOfPrint](#) の後、アプリケーションは指定されているテープ（ページ）のフォームデータを [getFormDataForPage](#) でライブラリーに渡します。[startPage](#) の後、[getStringContentData](#) または [getBitmapContentData](#) で指定されたコンテンツをライブラリーに渡します。アプリケーションは [doPrint](#) 以降、[onChangePrintOperationPhase](#) もしくは [onSuspendPrintOperation](#) か [onAbortPrintOperation](#) が呼び出されるまで、[TepraPrintDataProvider](#) 実装クラスを保持する必要があります。



### 5.5. ビットマップ印刷

ビットマップイメージの印刷シーケンスは、以下の通りです。



## 5.6. フォームデータ・コンテンツデータ

### フォームデータ設定（オーバーライド）

ライブラリーからコールされたアプリケーション側オーバーライドメソッド ([getFormDataForPage](#))は該当するページのフォームデータを渡します。

フォームデータの詳細は別紙「TEPRA-Print SDK Library Form Data Reference.pdf」を参照してください。

## 6.API 一覧

Class	API	function
<a href="#">TepraPrint</a>	<a href="#">TepraPrint</a>	TepraPrint クラスのコンストラクター
	<a href="#">getVersion</a>	SDK のバージョン情報を取得
	<a href="#">setPrinterInformation</a>	プリンター情報をセット
	<a href="#">setObjectType</a>	オブジェクト情報をセット (SR-R7900P のみ)
	<a href="#">setCallback</a>	コールバックをセット
	<a href="#">getLabelImages</a>	ラベルイメージデータを取得
	<a href="#">getPrintableSizeFromTape</a>	テープの印字領域を取得
	<a href="#">doPrint</a>	印刷
	<a href="#">getProgressOfPrint</a>	データ送信進捗 (0.0~1.0)
	<a href="#">getPageNumberOfPrinting</a>	印刷部数を加味したデータ送信中のページ番号
	<a href="#">cancelPrint</a>	印刷中止
	<a href="#">fetchPrinterStatus</a>	プリンターステータス
	<a href="#">doTapeFeed</a>	テープ送り、テープカットを実行
	<a href="#">resumeOfPrint</a>	一時停止中のジョブを再開
	<a href="#">getTapeWidthFromStatus</a>	ステータスからテープ幅を取得
	<a href="#">getTapeKindFromStatus</a>	ステータスからテープ種類を取得
	<a href="#">getTapeSwitchFromStatus</a>	ステータスからテープ検出状態を取得 ( SR-R7900P/SR-MK1/SR-R2500P のみ)
	<a href="#">getTapeOptionFromStatus</a>	ステータスからテープオプションを取得 (SR-R7900P のみ)
	<a href="#">getDeviceErrorFromStatus</a>	ステータスからテプラのエラー情報を取得
	<a href="#">getBatteryLevelFromStatus</a>	ステータスから電圧状態を取得 (SR-MK1/SR-R2500P のみ)
	<a href="#">setBatteryType</a>	バッテリー種類をセット (SR-MK1/SR-R2500P のみ)
	<a href="#">getBatteryType</a>	バッテリー種類を取得 (SR-MK1/SR-R2500P のみ)

	<a href="#">getKindOfTape</a>	対応しているテープ幅のリストを取得
	<a href="#">getModelName</a>	モデル名を取得
	<a href="#">getResolution</a>	解像度を取得
	<a href="#">isSupportHalfCut</a>	ハーフカット対応状況を取得
	<a href="#">isPrintSpeedSupport</a>	低速印刷対応状況を取得
	<a href="#">isTapeWinder</a>	小巻機接続状態を取得 (SR-R7900P のみ)
	<a href="#">isBigroll</a>	EX ロングテープ情報を取得 (SR-R7900P のみ)
	<a href="#">getBigrollTapeLengthFromStatus</a>	EX ロングテープのテープ長を取得 (SR-R7900P のみ)
	<a href="#">getBigrollTapeUsageLengthFromStatus</a>	EX ロングテープのテープ使用長を取得 (SR-R7900P のみ)
<a href="#">TepraPrintCallback</a>	<a href="#">onChangePrintOperationPhase</a>	印刷状態が遷移した場合に呼出
	<a href="#">onSuspendPrintOperation</a>	エラーにより印刷が一時停止された場合に呼出
	<a href="#">onAbortPrintOperation</a>	エラーにより印刷が中止された場合に呼出
	<a href="#">onChangeTapeFeedOperationPhase</a>	テープ送りテープカットの状態が遷移した場合に呼出
	<a href="#">onAbortTapeFeedOperation</a>	エラーによりテープ送り、テープカットが中止された場合に呼出
<a href="#">TepraPrintDiscoverPrinter</a>	<a href="#">TepraPrintDiscoverPrinter</a>	TepraPrintDiscoverPrinter クラスのコンストラクター
	<a href="#">setCallback</a>	コールバックをセット
	<a href="#">startDiscover</a>	検索開始
	<a href="#">stopDiscover</a>	検索終了
<a href="#">TepraPrintDiscoverPrinterCallback</a>	<a href="#">onFindPrinter</a>	プリンターが検索された
	<a href="#">onRemovePrinter</a>	プリンターが削除された
<a href="#">TepraPrintDataProvider</a>	<a href="#">startOfPrint</a>	ジョブ開始
	<a href="#">endOfPrint</a>	ジョブ終了
	<a href="#">startPage</a>	ページ開始
	<a href="#">endPage</a>	ページ終了
	<a href="#">getNumberOfPages</a>	印刷するページ数を取得
	<a href="#">getFormDataForPage</a>	フォームデータを取得

	<a href="#"><u>getBitmapContentData</u></a>	コンテンツデータ（Bitmap） を取得
	<a href="#"><u>getStringContentData</u></a>	コンテンツデータ（String）を 取得

## 7. API リファレンス

### TepraPrint クラスリファレンス

TepraPrint 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。

### TepraPrint メソッド

#### TepraPrint

TepraPrint クラスのコンストラクターです。

```
- TepraPrint(Context context)
```

#### パラメーター

context

コンテキストオブジェクト。

#### getVersion

SDK のバージョン情報を返します。

```
- static String getVersion()
```

#### リターン

SDK のバージョン情報。

#### setPrinterInformation

プリンター情報をセットします。

```
- void setPrinterInformation(Map<String, String> printerInformation)
```

#### パラメーター

printerInformation

プリンター情報。

[TepraPrintDiscoverPrinterCallback](#) の [onFindPrinter](#) で返された値をセットします。

### setObjectType

オブジェクト情報をセットします。（SR-R7900P のみ）

```
- void setObjectType(Set<TepraPrintObjectType> type)
```

#### パラメーター

type

[TepraPrintObjectType](#) オブジェクト。

### setCallback

コールバックをセットします。

```
- void setCallback(TepraPrintCallback callback)
```

#### パラメーター

callback

コールバックオブジェクト。

### getLabelImages

ラベルイメージデータターを返します。

```
- List<Bitmap> getLabelImages(TepraPrintDataProvider dataProvider,  
Map<String, Object> printParameter)
```

#### パラメーター

dataProvider

[TepraPrintDataProvider](#) オブジェクト。

printParameter

印刷設定パラメーター。[TepraPrintParameterKey](#) を参照。

#### リターン

ラベルイメージデータター。

### getPrintableSizeFromTape

テープの印字領域を返します。

```
- int getPrintableSizeFromTape(int width)
```

#### パラメーター

width

テープの幅。[TepraPrintTapeWidth](#)を参照。

#### リターン

印字領域。

## doPrint

印刷を開始します。

```
- void doPrint(TepraPrintDataProvider dataProvider, Map<String, Object>
printParameter)
```

### パラメーター

dataProvider

[TepraPrintDataProvider](#) オブジェクト。

printParameter

印刷設定パラメーター。 [TepraPrintParameterKey](#) を参照。

ビットマップ印刷を開始します。

```
- void doPrint(Bitmap bitmap, Map<String, Object> printParameter)
```

### パラメーター

bitmap

Bitmap オブジェクト。

printParameter

印刷設定パラメーター。 [TepraPrintParameterKey](#) を参照。

ビットマップ配列の印刷を開始します。

```
- void doPrint(ArrayList<Bitmap> bitmapArray, Map<String, Object>
printParameter, int margin)
```

### パラメーター

bitmapArray

Bitmap オブジェクトの配列。

printParameter

印刷設定パラメーター。 [TepraPrintParameterKey](#) を参照。

margin

余白（単位：mm）。

### **getProgressOfPrint**

データ送信の進捗を返します。

```
- float getProgressOfPrint()
```

#### **リターン**

データ送信の進捗 (0.0~1.0)。

### **getPageNumberOfPrinting**

印刷部数を加味したデータ送信中のページ番号を返します。

```
- int getPageNumberOfPrinting()
```

#### **リターン**

データ送信中のページ番号。

### **cancelPrint**

印刷を中止します。

```
- void cancelPrint()
```

### **fetchPrinterStatus**

ステータスを取得します。

```
- Map<String, Integer> fetchPrinterStatus()
```

#### **リターン**

ステータス。[TepraPrintStatusKey](#) を参照。

### **doTapeFeed**

テープ送り、テープカットを実行します。

```
- void doTapeFeed(int mode)
```

#### **パラメーター**

mode

テープ送り・テープカット。[TepraPrintTapeOperation](#) を参照。

### **resumeOfPrint**

一時停止中のジョブを再開します。

```
- void resumeOfPrint()
```

### **getTapeWidthFromStatus**

ステータスからテープ幅を取得します。

```
- int getTapeWidthFromStatus(Map<String, Integer> status)
```

#### **パラメーター**

status

ステータス。

#### **リターン**

[TepraPrintTapeWidth](#) : テープ幅。

### **getTapeKindFromStatus**

ステータスからテープ種類を取得します。

```
- int getTapeKindFromStatus(Map<String, Integer> status)
```

#### **パラメーター**

status

ステータス。

#### **リターン**

[TepraPrintTapeKind](#) : テープ種類。

### getTapeSwitchFromStatus

ステータスからテープ検出状態を取得します。（SR-R7900P/SR-MK1/SR-R2500P のみ）

```
- Map<String, Integer> getTapeSwitchFromStatus (Map<String, Integer>
status)
```

#### パラメーター

status

ステータス。

#### リターン

テープ検出状態。[TepraPrintTapeSwitchKey](#) を参照。

### getTapeOptionFromStatus

ステータスからテープオプションを取得します。（SR-R7900P のみ）

```
- Map<String, Integer> getTapeOptionFromStatus (Map<String, Integer>
status)
```

#### パラメーター

status

ステータス。

#### リターン

テープオプション。[TepraPrintTapeOptionKey](#) を参照。

### getDeviceErrorFromStatus

ステータスからテプラのエラー情報を取得します。

```
- int getDeviceErrorStatus (Map<String, Integer> status)
```

#### パラメーター

status

ステータス。

#### リターン

[TepraPrintStatusError](#) : テプラのエラー情報。

### getBatteryLevelFromStatus

ステータスから電圧状態を取得します。（SR-MK1/SR-R2500P のみ）

```
- int getBatteryLevelFromStatus (Map<String, Integer> status)
```

#### パラメーター

status

ステータス。

#### リターン

[TepraPrintBatteryLevel](#) : 電圧状態。

### setBatteryType

バッテリー種類をセットします。（SR-MK1/SR-R2500P のみ）

```
- void setBatteryType (int batteryType)
```

#### パラメーター

batteryType

バッテリー種類。[TepraPrintBatteryType](#) を参照。

### getBatteryType

バッテリー種類を取得します。（SR-MK1/SR-R2500P のみ）

```
- int getBatteryType ()
```

#### リターン

[TepraPrintBatteryType](#) : バッテリー種類。

### getKindOfTape

対応しているテープ幅のリストを返します。

```
- List<Integer> getKindOfTape ()
```

#### リターン

対応しているテープ幅([TepraPrintTapeWidth](#))一覧。Integer の List で返ります。

### **getModelName**

モデル名を返します。

```
- String getModelName()
```

#### **リターン**

モデル名。

### **getResolution**

解像度を返します。

```
- int getResolution()
```

#### **リターン**

解像度（180/270/300/360）。

### **isSupportHalfCut**

ハーフカット対応状況を取得します。

```
- boolean isSupportHalfCut()
```

#### **リターン**

ハーフカット機能をサポートしている場合は true。

### **isPrintSpeedSupport**

低速印刷対応状況を取得します。

```
- boolean isPrintSpeedSupport()
```

#### **リターン**

低速印刷をサポートしている場合は true。

### isTapeWinder

小巻機接続状態を取得します。(SR-R7900P のみ)

```
- boolean isTapeWinder(Map<String, Integer> status)
```

#### パラメーター

status

ステータス。

#### リターン

小巻機が接続状態の場合は true。

### isBigroll

EX ロングテープ情報を取得します。(SR-R7900P のみ)

```
- boolean isBigroll(Map<String, Integer> status)
```

#### パラメーター

status

ステータス。

#### リターン

EX ロングテープ情報が有効な場合は true。

### getBigrollTapeLengthFromStatus

ステータスから EX ロングテープのテープ長を取得します。(SR-R7900P のみ)

```
- int getBigrollTapeLengthFromStatus(Map<String, Integer> status)
```

#### パラメーター

status

ステータス。

#### リターン

EX ロングテープのテープ長 (mm) 。

### **getBigrollTapeUsageLengthFromStatus**

ステータスから EX ロングテープのテープ使用長を取得します。（SR-R7900P のみ）

```
- int getBigrollTapeUsageLengthFromStatus (Map<String, Integer> status)
```

#### **パラメーター**

status

ステータス。

#### **リターン**

EX ロングテープのテープ使用長（mm）。

## TepraPrint Constant

これ以降は Constant の説明になります。

## TepraPrintTapeOperation

[doTapeFeed](#) のパラメーターです。

```
class TepraPrintTapeOperation {  
    static final int Feed = 0;  
    static final int FeedAndCut = -1;  
}
```

### Feed

テープ送り。

### FeedAndCut

テープ送りカット。

## TepraPrintTapeWidth

テープの幅。

対応テープ幅以外のテープ幅を指定した場合は、18mm になります。

```
class TepraPrintTapeWidth {  
    static final int None = 0;  
    static final int Normal_4mm = 1;  
    static final int Normal_6mm = 2;  
    static final int Normal_9mm = 3;  
    static final int Normal_12mm = 4;  
    static final int Normal_18mm = 5;  
    static final int Normal_24mm = 6;  
    static final int Normal_36mm = 7;  
    static final int Cable_24mm = 8;  
    static final int Cable_36mm = 9;  
    static final int Normal_50mm = 10;  
    static final int Normal_100mm = 11;  
    static final int New_50mm = 12;  
    static final int Unknown = -1;  
}
```

**None**

未装着。

**Normal\_4mm**

4mm。

**Normal\_6mm**

6mm。

**Normal\_9mm**

9mm。

**Normal\_12mm**

12mm。

**Normal\_18mm**

18mm。

**Normal\_24mm**

24mm。

**Normal\_36mm**

36mm。

**Cable\_24mm**

24mm ケーブル。

**Cable\_36mm**

36mm ケーブル。

**Normal\_50mm**

50mm（広幅）。

**Normal\_100mm**

100mm（広幅）。

**New\_50mm**

50mm。

**Unknown**

不明。

## TepraPrintTapeKind

テープの種類。

```
class TepraPrintTapeKind {
    static final int Normal = 0;
    static final int Transfer = 1;
    static final int Cable = 16;
    static final int Index = 17;
    static final int Braille = 64;
    static final int Olefin = 80;
    static final int ThermalPaper = 81;
    static final int Tube = 82;
    static final int PET = 83;
    static final int DieCut1 = 84;
    static final int DieCut2 = 85;
    static final int DieCut3 = 86;
    static final int DieCut4 = 87;
    static final int DieCut5 = 88;
    static final int WideReserved1 = 89;
    static final int WideReserved2 = 90;
    static final int WideReserved3 = 91;
    static final int WideReserved4 = 92;
    static final int WideReserved5 = 93;
    static final int WideReserved6 = 94;
    static final int WideReserved7 = 95;
    static final int DieCutCircle = 96;
    static final int DieCutEllipse = 97;
    static final int DieCutRoundedCorners = 98;
    static final int DieCutReserved1 = 99;
    static final int DieCutReserved2 = 100;
    static final int DieCutReserved3 = 101;
    static final int DieCutReserved4 = 102;
    static final int DieCutReserved5 = 103;
    static final int DieCutReserved6 = 104;
    static final int DieCutReserved7 = 105;
    static final int DieCutReserved8 = 106;
    static final int DieCutReserved9 = 107;
    static final int DieCutReserved10 = 108;
    static final int DieCutReserved11 = 109;
```

```
static final int DieCutReserved12 = 110;
static final int DieCutReserved13 = 111;
static final int HST = 112;
static final int Vinyl = 128;
static final int Cleaning = 144;
static final int EquipmentManagement = 145;
static final int Ribbon = 146;
static final int Magnet = 147;
static final int LuminousLight = 148;
static final int HighQualityPaper = 149;
static final int Iron = 150;
static final int BigrollDieCut = 151;
static final int BrPET = 201;
static final int BrVinyl = 202;
static final int BrHST = 203;
static final int BrSelfLam = 204;
static final int BrDieCutSelfLam1 = 210;
static final int BrDieCutSelfLam2 = 211;
static final int BrDieCutSelfLam3 = 212;
static final int BrDieCutSelfLam4 = 213;
static final int Unknown = -1;
}
```

**Normal**

標準テープ。

**Transfer**

転写テープ。

**Cable**

ケーブルテープ（セルフラミ）。

**Index**

インデックステープ。

**Braille**

点字テープ。

**Olefin**

オレフィン（広幅）。

**ThermalPaper**

感熱紙（広幅）。

**Tube**

チューブ（広幅）。

**PET**

PET（広幅）。

**DieCut1**

ダイカット 1 4"×6"（広幅）。

**DieCut2**

ダイカット 2 4"×3"（広幅）。

**DieCut3**

ダイカット 3 4"×2"（広幅）。

**DieCut4**

ダイカット 4 2.5"×3"（広幅）。

**DieCut5**

ダイカット 5 1"×2"（広幅）。

**WideReserved1**

広幅予約（広幅）。

**WideReserved2**

広幅予約（広幅）。

**WideReserved3**

広幅予約（広幅）。

**WideReserved4**

広幅予約（広幅）。

**WideReserved5**

広幅予約（広幅）。

**WideReserved6**

広幅予約（広幅）。

**WideReserved7**

広幅予約（広幅）。

**DieCutCircle**

ダイカット丸（36mm）。

**DieCutEllipse**

ダイカット楕円（36mm）。

**DieCutRoundedCorners**

ダイカット角丸（36mm）。

**DieCutReserved1**

ダイカット予約（36mm）。

**DieCutReserved2**

ダイカット予約（36mm）。

**DieCutReserved3**

ダイカット予約（36mm）。

**DieCutReserved4**

ダイカット予約（36mm）。

**DieCutReserved5**

ダイカット予約（36mm）。

**DieCutReserved6**

ダイカット予約（36mm）。

**DieCutReserved7**

ダイカット予約（36mm）。

**DieCutReserved8**

ダイカット予約（36mm）。

**DieCutReserved9**

ダイカット予約（36mm）。

**DieCutReserved10**

ダイカット予約（36mm）。

**DieCutReserved11**

ダイカット予約（36mm）。

**DieCutReserved12**

ダイカット予約（36mm）。

**DieCutReserved13**

ダイカット予約（36mm）。

**HST**

HST（熱収縮チューブ）。

**Vinyl**

ビニール。

**Cleaning**

クリーニングテープ。

**EquipmentManagement**

備品管理。

**Ribbon**

リボン。

**Magnet**

マグネット。

**LuminousLight**

蓄光。

**HighQualityPaper**

上質紙。

**Iron**

アイロン。

**BigrollDieCut**

EX ロングテープダイカット。

**BrPET**

PET（EX ロングテープ）。

**BrVinyl**

ビニール（EX ロングテープ）。

**BrHST**

HST（EX ロングテープ）。

**BrSelfLam**

セルフラミ（EX ロングテープ）。

**BrDieCutSelfLam1**

ダイカット セルフラミ 12.7×19.1（EX ロングテープ）。

**BrDieCutSelfLam2**

ダイカット セルフラミ 12.7×38.1（EX ロングテープ）。

**BrDieCutSelfLam3**

ダイカット セルフラミ 25.4×38.1（EX ロングテープ）。

**BrDieCutSelfLam4**

ダイカット セルフラミ 38.1×38.1（EX ロングテープ）。

**Unknown**

その他。

### **TepraPrintPrintingPhase**

印刷、テープ送り、テープカットの状態。

```
class TepraPrintPrintingPhase {  
    static final int Prepare = 1;  
    static final int Processing = 2;  
    static final int WaitingForPrint = 3;  
    static final int Complete = 4;  
}
```

#### **Prepare**

準備中。

#### **Processing**

処理中。

#### **WaitingForPrint**

印刷待ち。

#### **Complete**

終了。

## **TepraPrintConnectionStatus**

印刷、テープ送り、テープカットのエラー状態

```
class TepraPrintConnectionStatus {  
    static final int NoError = 0;  
    static final int ConnectionFailed = -1;  
    static final int Disconnected = -2;  
    static final int DeviceBusy = -3;  
    static final int OutOfMemory = -4;  
    static final int DeviceError = -5;  
    static final int CommunicationError = -6;  
    static final int OtherUsing = -1000;  
}
```

### **NoError**

エラーなし。

### **ConnectionFailed**

接続に失敗した。

### **Disconnected**

接続が切断された。

### **DeviceBusy**

プリンターが使用中。

### **OutOfMemory**

メモリ不足。

### **DeviceError**

テプラエラー。

### **CommunicationError**

通信エラー。

### **OtherUsing**

他人が使用中。

## TepraPrintStatusError

テブラのエラー状態

```
class TepraPrintStatusError {
    static final int NoError = 0x00;
    static final int CutterError = 0x01;
    static final int NoTapeCartridge = 0x06;
    static final int HeadOverheated = 0x15;
    static final int PrinterCancel = 0x20;
    static final int CoverOpen = 0x21;
    static final int LowVoltage = 0x22;
    static final int PowerOffCancel = 0x23;
    static final int TapeEjectError = 0x24;
    static final int TapeFeedError = 0x30;
    static final int InkRibbonSlack = 0x40;
    static final int InkRibbonShort = 0x41;
    static final int TapeEnd = 0x42;
    static final int CutLabelError = 0x43;
    static final int TemperatureError = 0x44;
    static final int InsufficientParameters = 0x45;
    static final int HalfCutterBladeNotSet = 0x50;
    static final int FullCutterBladeNotSet = 0x51;
    static final int HalfCutterBladeOff = 0x52;
    static final int FullCutterBladeOff = 0x53;
    static final int WinderCoverOpen = 0x54;
    static final int VinylTapeTemperatureError = 0x55;
    static final int WinderError = 0x56;
    static final int HalfCutAllCut = 0x57;
    static final int BigrollRecognitionAbnormality = 0x58;
    static final int BigrollNonCompliant = 0x59;
    static final int StopPrintingByAutoPowerOff = 0x5c;
    static final int StopPrintingByPowerSupplyChange = 0x5d;
    static final int WinderSet = 0x5e;
    static final int WinderNotSet = 0x5f;
    static final int WinderHalfCutAllCut = 0x60;
    static final int ConnectionFailed = 0xffffffff0;
    static final int OtherUsing = 0xfffffffffa;
    static final int FirmwareUpdating = 0xfffffffffb;
    static final int DeviceUsing = 0xfffffffffc;
}
```

```
static final int UnknownError = 0xffffffff;  
}
```

**NoError**

エラーなし。

**CutterError**

カッター異常。

**NoTapeCartridge**

テープ未装着。

**HeadOverheated**

ヘッド過熱。

**PrinterCancel**

テプラによる印刷中止。

**CoverOpen**

蓋開き。

**LowVoltage**

電圧低下。

**PowerOffCancel**

電源 OFF による印刷中止。

**TapeEjectError**

テープ排紙異常。

**TapeFeedError**

テープ送り異常。

**InkRibbonSlack**

インクリボンたるみ異常。

**InkRibbonShort**

インクリボン残量少。

**TapeEnd**

テープ／ラベル終了。

**CutLabelError**

カットラベル位置合わせ異常。

**TemperatureError**

環境温度過熱異常。

**InsufficientParameters**

印刷パラメーター不足。

**HalfCutterBladeNotSet**

ハーフカッターの刃未セット。

**FullCutterBladeNotSet**

フルカッターの刃未セット。

**HalfCutterBladeOff**

ハーフカッターの刃外れ。

**FullCutterBladeOff**

フルカッターの刃外れ。

**WinderCoverOpen**

小巻蓋開き（小巻未セット）。

**VinylTapeTemperatureError**

ビニールテープ環境温度異常。

**WinderError**

小巻巻き取り異常。

**HalfCutAllCut**

ハーフカット全切れ。

**BigrollRecognitionAbnormality**

EX ロングテープ認識異常。

**BigrollNonCompliant**

EX ロングテープ非対応。

**StopPrintingByAutoPowerOff**

オートパワーオフによる印刷停止。

**StopPrintingByPowerSupplyChange**

電源変化による印刷停止。

**WinderSet**

小巻機セット。

**WinderNotSet**

小巻機未セット。

**WinderHalfCutAllCut**

小巻ハーフカット全切れ。

**ConnectionFailed**

接続エラー。

**OtherUsing**

他人が使用中。

**FirmwareUpdating**

ファームウェアアップデート。

**DeviceUsing**

デバイスエラー。

**UnknownError**

その他のエラー。

## **TepraPrintTapeCut**

テープカットの種類。

```
class TepraPrintTapeCut {  
    static final int EachLabel = 0;  
    static final int AfterJob = 1;  
    static final int CutNotCut = 2;  
}
```

### **EachLabel**

ラベル毎にテープカットする。

### **AfterJob**

印刷 JOB 毎にテープカットする。

### **CutNotCut**

テープカットしない。

CutNotCut を設定した場合、TepraPrintParameterKey.HalfCut の設定は false（ハーフカットしない）になります。

## **TepraPrintPrintSpeed**

印刷速度。

```
class TepraPrintPrintSpeed {  
    PrintSpeedHigh = 0,  
    PrintSpeedLow = 1,  
    PrintSpeedMiddle = 2  
};
```

### **PrintSpeedHigh**

高速印刷する。

### **PrintSpeedLow**

低速印刷する。

### **PrintSpeedMiddle (SR-R7900P のみ)**

中速印刷する。

## **TepraPrintObjectType**

印刷データに含まれるオブジェクト種類。（SR-R7900P のみ）

```
enum TepraPrintObjectType {  
    None, Barcode1D, Barcode2D  
}
```

### **None**

オブジェクトなし。

### **Barcode1D**

1次元バーコード。

### **Barcode2D**

2次元バーコード。

## **TepraPrintBatteryLevel**

電圧状態。

```
class TepraPrintBatteryLevel {  
    static final int NoLowVoltage = 0;  
    static final int WarningLowVoltage = 1;  
    static final int ErrorLowVoltage = 2;  
}
```

### **NoLowVoltage**

良好。

### **WarningLowVoltage**

電圧低下警告。

### **ErrorLowVoltage**

電圧低下エラー。

## **TepraPrintBatteryType**

バッテリー種類。

```
class TepraPrintBatteryType {  
    static final int AlkalineBattery = 0;  
    static final int NickelHydrogenBattery = 1;  
}
```

### **AlkalineBattery**

アルカリ電池。

### **NickelHydrogenBattery**

ニッケル水素電池。

## TepraPrintParameterKey

[doPrint](#) で設定する辞書のキーです。

```
class TepraPrintParameterKey {  
    static final String Copies;  
    static final String TapeCut;  
    static final String HalfCut;  
    static final String PrintSpeed;  
    static final String Density;  
    static final String TapeWidth;  
    static final String PriorityPrintSetting;  
    static final String HalfCutContinuous;  
}
```

### Copies

印刷部数：1～99。

### TapeCut

テープカットの種類。 [TepraPrintTapeCut](#) を参照。

### HalfCut

true：ハーフカットする。

false：ハーフカットしない。

### PrintSpeed

印刷速度。 [TepraPrintPrintSpeed](#) を参照。

### Density

印刷濃度：-5～+5。

### TapeWidth

テープの幅。 [TepraPrintTapeWidth](#) を参照。

### PriorityPrintSetting（SR-R7900P/SR-MK1/SR-R2500P のみ）

true：指定した印刷設定を優先する。

false：テープカートリッジの印刷設定に従う。

**HalfCutContinuous (SR-R7900P のみ)**

true : ハーフカット連続で切り離れたラベルを作る。

false : ハーフカット連続でつながったラベルを作る。

### **TepraPrintStatusKey**

ステータス辞書のキーです。

```
class TepraPrintStatusKey {  
    static final String TapeKind;  
    static final String TapeWidth;  
    static final String DeviceError;  
}
```

#### **TapeKind**

テープの種類。

#### **TapeWidth**

テープの幅。

#### **DeviceError**

デバイスの状態。

### **TepraPrintTapeSwitchKey**

テープ検出状態辞書のキーです。（SR-R7900P/SR-MK1/SR-R2500P のみ）

```
class TepraPrintTapeSwitchKey {  
    static final String Magnet;  
    static final String Bigroll;  
}
```

#### **Magnet**

マグネット（0: 非マグネット / 1: マグネット）。

#### **Bigroll**

EX ロングテープ（0: 非 EX ロングテープ / 1: EX ロングテープ）。

## TepraPrintTapeOptionKey

テープオプション辞書のキーです。（SR-R7900P のみ）

```
class TepraPrintTapeOptionKey {  
    static final String FullCut;  
    static final String HalCut;  
    static final String Density;  
    static final String PrintSpeed;  
    static final String Mirror;  
    static final String Winder;  
}
```

### FullCut

フルカット（0: 許可 / 1: 禁止）。

### HalfCut

ハーフカット（0: 許可 / 1: 禁止）。

### Density

印刷濃度（-3～+3）。

### PrintSpeed

印刷速度（0: 高速 / 1: 低速 / 2: 中速）。

### Mirror

鏡文字（0: 正像 / 1: 鏡像）。

### Winder

巻き戻し（0: 許可 / 1: 禁止）。

## TepraPrintCallback インターフェイスリファレンス

### TepraPrintCallback メソッド

#### onChangePrintOperationPhase

印刷状態が遷移した場合に呼び出されます。

```
- void onChangePrintOperationPhase(TepraPrint tepraPrint, int phase)
```

#### パラメーター

tepraPrint

TepraPrint オブジェクト。

phase

ジョブステータス。[TepraPrintPrintingPhase](#) を参照。

#### onSuspendPrintOperation

エラーにより印刷が一時停止された場合に呼び出されます。

```
- void onSuspendPrintOperation(TepraPrint tepraPrint, int errorStatus, int deviceStatus)
```

#### パラメーター

tepraPrint

TepraPrint オブジェクト

errorStatus

ジョブのエラー情報。[TepraPrintConnectionStatus](#) を参照。

deviceStatus

テプラのエラー情報。[TepraPrintStatusError](#) を参照。

### onAbortPrintOperation

エラーにより印刷が中止された場合に呼び出されます。

```
- void onAbortPrintOperation(TepraPrint tepraPrint, int errorStatus, int deviceStatus)
```

#### パラメーター

tepraPrint

TepraPrint オブジェクト

errorStatus

ジョブのエラー情報。[TepraPrintConnectionStatus](#) を参照。

deviceStatus

テプラのエラー情報。[TepraPrintStatusError](#) を参照。

### onChangeTapeFeedOperationPhase

テープ送り、テープカットの状態が遷移した場合に呼び出されます。

```
- void onChangeTapeFeedOperationPhase(TepraPrint tepraPrint, int phase)
```

#### パラメーター

tepraPrint

TepraPrint オブジェクト

phase

ジョブステータス。[TepraPrintPrintingPhase](#) を参照。

### onAbortTapeFeedOperation

エラーによりテープ送り、テープカットが中止された場合に呼び出されます。

```
- void onAbortTapeFeedOperation(TepraPrint tepraPrint, int errorStatus,  
int deviceStatus)
```

#### パラメーター

tepraPrint

TepraPrint オブジェクト

errorStatus

ジョブのエラー情報。[TepraPrintConnectionStatus](#) を参照。

deviceStatus

テプラのエラー情報。[TepraPrintStatusError](#) を参照。

## TepraPrintDiscoverPrinter クラスリファレンス

### TepraPrintDiscoverPrinter メソッド

#### TepraPrintDiscoverPrinter

TepraPrintDiscoverPrinter クラスのコンストラクターです。

```
- TepraPrintDiscoverPrinter(List<String> types)
```

##### パラメーター

types

検索するタイプの一覧。

記述例 : "\_pdl-datastream.\_tcp.local."

```
- TepraPrintDiscoverPrinter(List<String> types, List<String> models,  
    EnumSet<TepraPrintDiscoverConnectionType> connectionType)
```

##### パラメーター

types

検索するタイプの一覧。

記述例 : "\_pdl-datastream.\_tcp.local."

models

検索するモデルの配列。

記述例 : "(KING JIM TEPRA PRO SR5900P)", "SR5500P", "(KING JIM TEPRA PRO SR-R7900P)",  
"SR-MK1", "SR-R2500P"

connectionType

接続方法。 [TepraPrintDiscoverConnectionType](#) を参照。

### **setCallback**

コールバックをセットします。

```
- void setCallback(TepraPrintDiscoverPrinterCallback callback)
```

#### **パラメーター**

callback

コールバックオブジェクト。

### **startDiscover**

検索を開始します。

```
- void startDiscover(Context context)
```

#### **パラメーター**

context

コンテキストオブジェクト。

### **stopDiscover**

検索を終了します。

```
- void stopDiscover()
```

## TepraPrintDiscoverPrinter Constant

これ以降は Constant の説明になります。

### TepraPrintDiscoverConnectionType

[TepraPrintDiscoverPrinter](#) のパラメーターです。

```
enum TepraPrintDiscoverConnectionType {  
    ConnectionTypeAll(0),  
    ConnectionTypeNetwork(1 << 0),  
    ConnectionTypeBluetooth(1 << 1);  
}
```

#### ConnectionTypeAll

すべての接続。

#### ConnectionTypeNetwork

ネットワーク。

#### ConnectionTypeBluetooth

Bluetooth。

## TepraPrintDiscoverPrinterCallback インターフェイスリファレンス

### TepraPrintDiscoverPrinterCallback メソッド

#### onFindPrinter

プリンターが検索された場合に呼び出されます。

```
- void onFindPrinter(TepraPrintDiscoverPrinter discoverPrinter,  
Map<String, String> printer)
```

#### パラメーター

discoverPrinter

TepraPrintDiscoverPrinter オブジェクト。

printer

見つかったプリンターの配列を返しますので、[setPrinterInformation](#) の引数として渡してください。

#### onRemovePrinter

プリンターが削除された場合に呼び出されます。

```
- void onRemovePrinter(TepraPrintDiscoverPrinter discoverPrinter,  
Map<String, String> printer)
```

#### パラメーター

discoverPrinter

TepraPrintDiscoverPrinter オブジェクト。

printer

削除されたプリンターの配列。

## TepraPrintDataProvider インターフェイスリファレンス

### TepraPrintDataProvider メソッド

#### startOfPrint

ジョブ開始の処理を行います。

```
- void startOfPrint()
```

#### endOfPrint

ジョブ終了の処理を行います。

```
- void endOfPrint()
```

#### startPage

ページ開始の処理を行います。

```
- void startPage()
```

#### endPage

ページ終了の処理を行います。

```
- void endPage()
```

#### getNumberOfPages

印刷するページ数を返します。

```
- int getNumberOfPages()
```

#### リターン

総ページ数。

## **getFormdataForPage**

フォームデータを返します。

```
- InputStream getFormdataForPage(int pageIndex)
```

### **パラメーター**

pageIndex

印刷するページ番号（1 から）。

### **リターン**

該当するページのフォームデータ。

フォームデータの詳細は、

別紙の「TEPRA-Print SDK Library Form Data Reference.pdf」を参照してください。

## **getBitmapContentData**

コンテンツデータを返します。

```
- Bitmap getBitmapContentData(String contentName, int pageIndex)
```

### **パラメーター**

contentName

コンテンツ名。

pageIndex

印刷するページ番号。

### **リターン**

該当するコンテンツ(Bitmap)。

### **備考**

ページ内の同一コンテンツが複数回取得される場合があります。

### getStringContentData

コンテンツデータを返します。

```
- String getStringContentData(String contentName, int pageIndex)
```

#### パラメーター

contentName

コンテンツ名。

pageIndex

印刷するページ番号。

#### リターン

該当するコンテンツ(String)。

#### 備考

ページ内の同一コンテンツが複数回取得される場合があります。

## 8. サンプルアプリケーション

### 8.1. 概要

TEPRA-Print SDK for Android が提供するサンプルアプリケーションは、Androidアプリケーションソフト開発者向けの実装サンプルです。

Javaソースファイルを含むAndroid Studio用Androidアプリケーションプロジェクトを提供します。

---

#### Note:

提供するサンプルプログラムは、API Level 23 で作成されています。

---

アプリケーション	開発環境	開発言語	対応 OS バージョン
TepraPrintSample	Android Studio Dolphin	Java	Android 6.0 以降
TepraPrintSimpleSample	Android Studio Dolphin	Java	Android 6.0 以降

#### 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。

- ・ テプラ検索
- ・ 印刷
- ・ 印刷設定

#### フォームデータ

サンプルアプリケーションではフォームデータとコンテンツデータをそれぞれ対のplist形式にしています。

例) フォームデータ : simple.plist

simple.plistに対応するコンテンツデータ : simple\_CONTENTS.plist

simple.plistのTepraContentIDで指定される、コンテンツが  
simple\_CONTENTS.plistに格納されていますので、[TepraPrintDataProvider](#)では  
指定されたコンテンツの内容をsimple\_CONTENTS.plistから取得し、ライブラリーに返しています。  
サンプルアプリケーションでは、フォームとコンテンツを対の形式にしてありますが、  
コンテンツ自体はplistになっている必要はありませんので、お使いになるシステムに合わせて  
コンテンツを取得し、ライブラリーに渡すようにしてください。

## 8.2. 機能 - TepraPrintSample

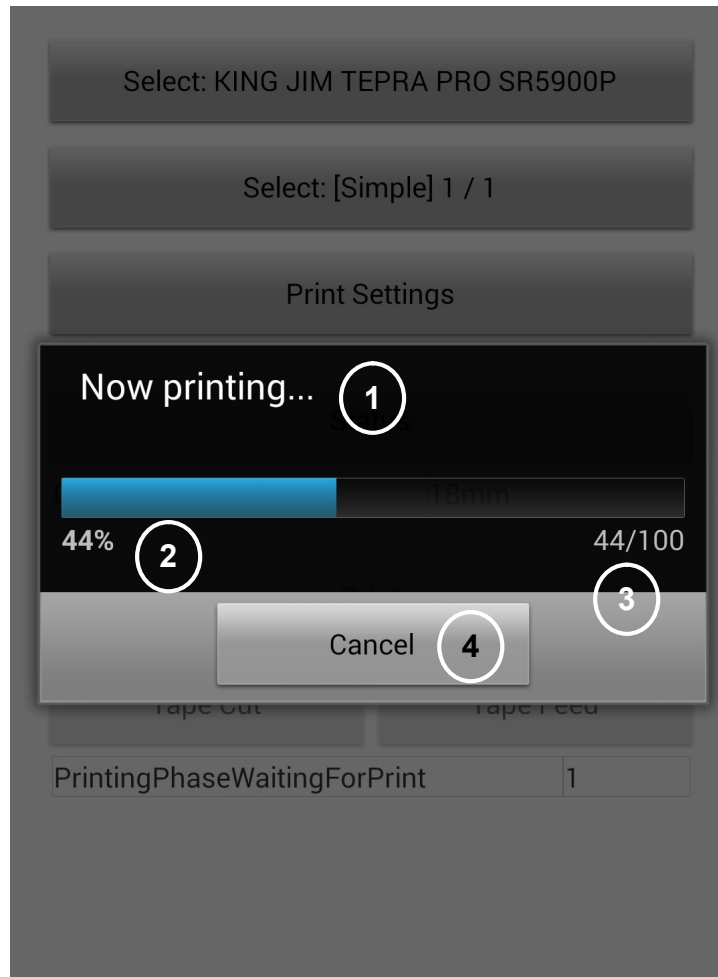
UI : メイン画面

1 Select: KING JIM TEPRA PRO SR5900P		
2 Select: [Simple] 1 / 1		
3 Print Settings		
4 Status		
0	5	18mm 6
Print 7		
8 Tape Cut		Tape Feed 9
PrintingPhaseComplete	10	1 11

## 説明：メイン画面

番号	内容
1	選択されているテブラを表示します。 タップするとテブラ検索画面を表示します。
2	選択されているフォームデーターを表示します。 タップするとフォームデーター選択画面を表示します。
3	プリントセッティング画面を表示します。
4	テブラのステータスを取得します。
5	デバイスエラーを表示します。
6	テブラにセットされているテープ幅を表示します。
7	印刷をします。
8	テープカットをします。
9	テープ送りをします。
10	印刷、テープ送り、テープカットの状態を表示します。
11	印刷中のページ数を表示します。

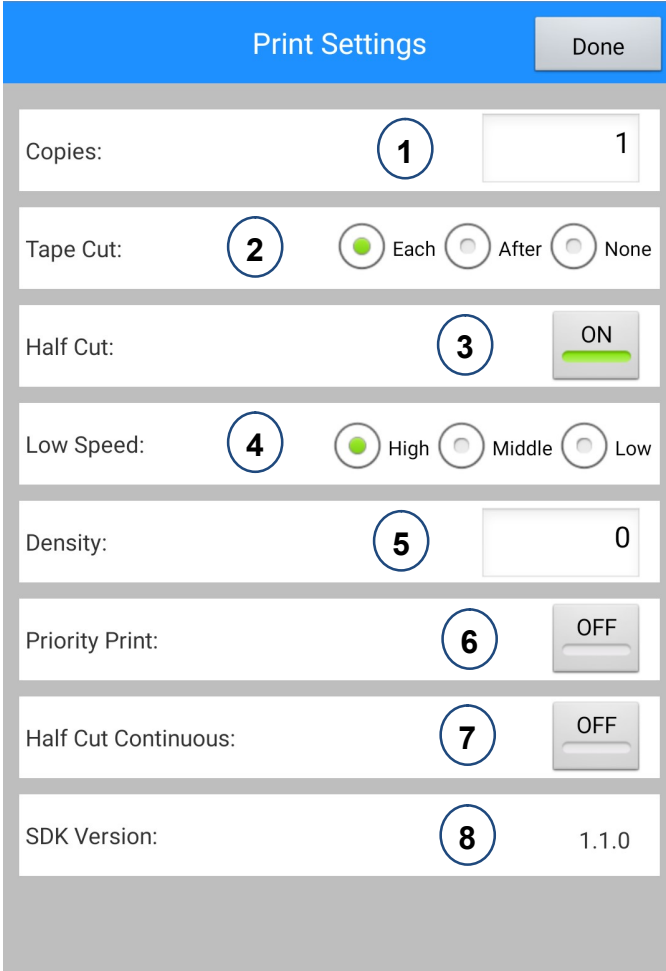
UI : ステータス画面



説明 : ステータス画面

番号	内容
1	印刷、テープ送り、テープカットの状態を表示します。
2	印刷の進捗を表示します。
3	印刷の進捗を表示します。
4	印刷を中止します。

## UI : プリントセッティング画面



The image shows a 'Print Settings' screen with a blue header bar containing the title 'Print Settings' and a 'Done' button. Below the header, there are eight settings sections, each with a circled number from 1 to 8:

- 1** Copies: A numeric input field showing '1'.
- 2** Tape Cut: Three radio buttons labeled 'Each' (selected), 'After', and 'None'.
- 3** Half Cut: A toggle switch labeled 'ON'.
- 4** Low Speed: Three radio buttons labeled 'High' (selected), 'Middle', and 'Low'.
- 5** Density: A numeric input field showing '0'.
- 6** Priority Print: A toggle switch labeled 'OFF'.
- 7** Half Cut Continuous: A toggle switch labeled 'OFF'.
- 8** SDK Version: A text field showing '1.1.0'.

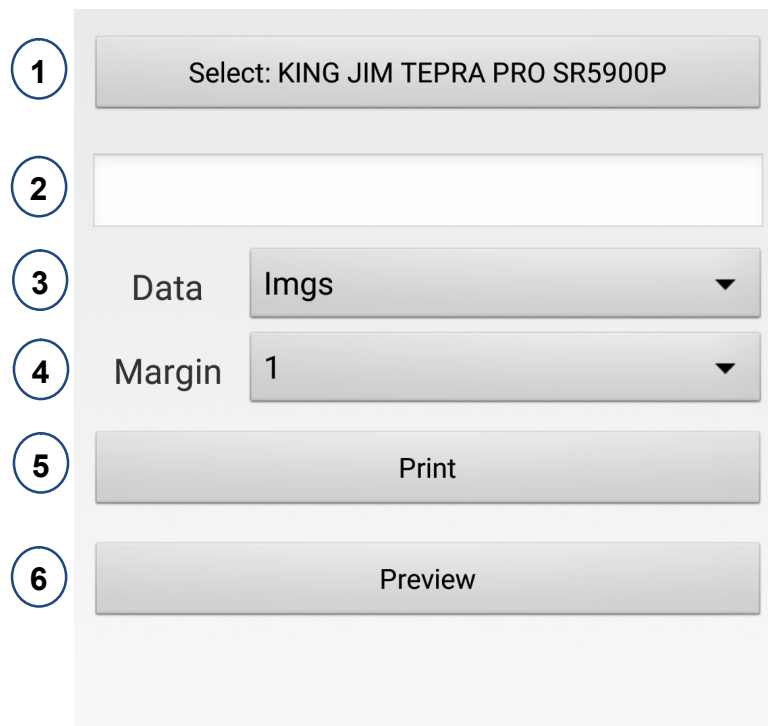
## 説明 : プリントセッティング画面

番号	内容	範囲
1	印刷部数を指定します。	1～99
2	テープカットの指定をします。 ハーフカット“する”の場合はハーフカットの指定になります。	Each : ラベル毎にテープカットします After : 印刷 JOB 毎にテープカットします None: テープカットしない
3	ハーフカットの指定をします。	ON: ハーフカットする / OFF: しない
4	印刷速度の指定をします。	High: 高速印刷 / Low: 低速印刷  SR-R7900P のみ High: 高速印刷 / Middle: 中速印刷 / Low: 低速印刷
5	印字濃度の指定をします。	-5～5

6	印刷設定優先の指定をします。	ON: アプリの印刷設定を優先／OFF: しない
7	ハーフカット連続の指定をします。	ON: 切り離す／OFF: つなげる
8	SDK のバージョン情報を表示します。	文字列

### 8.3. 機能 - TepraPrintSimpleSample

UI : メイン画面



The screenshot shows the main interface of the TepraPrintSimpleSample application. It features a light gray background with several interactive elements:

- 1**: A button labeled "Select: KING JIM TEPRA PRO SR5900P" at the top.
- 2**: A large, empty white rectangular input field below the selection button.
- 3**: A label "Data" followed by a dropdown menu currently showing "Imgs".
- 4**: A label "Margin" followed by a dropdown menu currently showing "1".
- 5**: A wide button labeled "Print" located below the configuration options.
- 6**: A wide button labeled "Preview" located at the bottom of the main content area.

説明 : メイン画面

番号	内容
1	選択されているテプラを表示します。 タップするとテプラ検索画面を表示します。
2	テキスト/QR コードの入力文字列を表示します。
3	テキスト/QR コード/画像 1/画像 2/画像（複数）の入力オブジェクトを切り替えます。
4	印刷前余白長の指定をします。  画像 1/画像 2/画像（複数）選択時のみ 1～21mm
5	印刷をします。
6	印刷プレビューを表示します。

## 9. その他

### 9.1. Android 6.0 以降での動作

Android 6.0 以降では、アプリケーションのアクセス権が個別に ON/OFF できるようになりました。アプリケーションのインストール後にもアクセス権の ON/OFF が任意で切り替えられるようになったため、必要な権限が許可されているかを確認する必要があります。

API レベル 23 からは、Bluetooth のデバイススキャンをするには位置情報（ACCESS\_FINE\_LOCATION または ACCESS\_COARSE\_LOCATION）の権限が必要です。

端末の Android バージョン	アプリの targetSDKVersion	アプリへの権限付与 タイミング	ユーザーによる 権限制御
≥8.0	≥26	アプリ実行時 (付与は Permission 単位)	あり
	<26	アプリ実行時 (付与は Group 単位)	あり
	<23	インストール時	あり
≥6.0	≥23	アプリ実行時 (付与は Group 単位)	あり
	<23	インストール時	あり
≤5.1	≥23	インストール時	なし
	<23	インストール時	なし

#### 権限が付与されているかのチェック：

適切なアクセスレベルを指定して、ContextCompat.checkSelfPermission() メソッドを呼び出します。

```
ContextCompat.checkSelfPermission(Context context, String permission);
```

権限が既に付与されている場合、このメソッドは PackageManager.PERMISSION\_GRANTED という値を返します。

#### 権限の要求：

適切なアクセスレベルの権限をユーザーに要求するには、ActivityCompat.requestPermissions() メソッドを使用します。

```
ActivityCompat.requestPermissions(Activity activity, String[] permissions, int requestCode);
```

このメソッドを呼び出すと、権限を付与するようにアプリケーションに促すダイアログボックスが表示され、選択が行われたときに onRequestPermissionsResult() メソッドが呼び出されます。

## 9.2. ProGuard settings

Android Studio で minifyEnabled を true に設定する場合は、proguard ファイルに以下を追加してください。

```
-keep class jp.co.kingjim.tepraprint.sdk.** { *; }
```

proguard ファイル (proguard-rules.pro) は build.gradle ファイルに以下のように設定します。

```
buildTypes {  
    release {  
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
    }  
}
```