

Epson ePOS SDK for Android

マイグレーションガイド

マイグレーションの概要

ePOS-Print SDK からのマイグレーション

ePOS-Device SDK からのマイグレーション

付録

ご注意

- 本書の内容の一部または全部を無断で転載、複写、複製、改ざんすることは固くお断りします。
- 本書の内容については、予告なしに変更することがあります。最新の情報はお問い合わせください。
- 本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がありましたらご連絡ください。
- 運用した結果の影響については、上項に関わらず責任を負いかねますのでご了承ください。
- 本製品がお客様により不適切に使用されたり、本書の内容に従わずに取り扱われたり、またはエプソンおよびエプソン指定の者以外の第三者により修理・変更されたことなどに起因して生じた損害などにつきまして、責任を負いかねますのでご了承ください。
- エプソン純正品およびエプソン品質認定品以外のオプションまたは消耗品を装着してトラブルが発生した場合には、責任を負いかねますのでご了承ください。

商標について

EPSON および EXCEED YOUR VISION はセイコーエプソン株式会社の登録商標です。

Android™ は、Google Inc. の商標です。

Bluetooth® のワードマークおよびロゴは、Bluetooth SIG, Inc. が所有する登録商標であり、セイコーエプソン株式会社はこれらのマークをライセンスに基づいて使用しています。その他の商標およびトレードネームは、それぞれの所有者に帰属します。

その他の製品名および会社名は、各社の商標または登録商標です。

© Seiko Epson Corporation 2015 - 2017. All rights reserved.

使用制限

本製品を航空機・列車・船舶・自動車などの運行に直接関わる装置・防災防犯装置・各種安全装置など機能・精度などにおいて高い信頼性・安全性が必要とされる用途に使用される場合は、これらのシステム全体の信頼性および安全維持のためにフェールセーフ設計や冗長設計の措置を講じるなど、システム全体の安全設計にご配慮いただいた上で弊社製品をご使用いただくようお願いいたします。

本製品は、航空宇宙機器、幹線通信機器、原子力制御機器、医療機器など、きわめて高い信頼性・安全性が必要とされる用途への使用を意図しておりませんので、これらの用途には本製品の適合性をお客様において十分ご確認の上、ご判断ください。

もくじ

■ 使用制限	3
■ もくじ	4

マイグレーションの概要..... 6

■ マイグレーションの種類	6
マイグレーションの種類	6
新製品・新機能への対応方針	7

ePOS-Print SDK からのマイグレーション..... 8

■ ePOS-Print SDK 互換 API を使用するマイグレーション	8
マイグレーションの手順	8
SDK の置き換え	8
ePOS-Print SDK 互換 API を使用したアプリケーション開発	8
■ Epson ePOS SDK の API を使用するマイグレーション	9
マイグレーションの手順	9
SDK の置き換え	10
パッケージ名の変更	10
クラスの変更	11
プリンターとの通信の接続と切断	12
印刷	13
ステータスの取得	15
プリンターの検索	17
ステータスの監視	19
API の名称変更	21
API のパラメーター変更	23
リスナーインターフェイスの変更	24

ePOS-Device SDK からのマイグレーション.....26

■ ePOS-Device SDK 互換 API を使用するマイグレーション	26
マイグレーションの手順	26
SDK の置き換え	26
ePOS-Device SDK 互換 API を使用したアプリケーション開発	26
■ Epson ePOS SDK の API を使用するマイグレーション	27
マイグレーションの手順	27
SDK の置き換え	28
パッケージ名の変更	28
クラスの変更	29
デバイスとの通信の接続と切断	30
再接続通知	33
印刷	35
強制送信	37
ステータスの取得	39

API の名称変更	41
API のパラメーター変更	45
リスナーインターフェイスの変更	46

付録50

■ ePOS-Print SDK 互換 API	50
プリンターごとのサポート API 一覧	50
TM-m10	52
TM-m30	54
TM-P80	56
TM-T88VI	58

マイグレーションの概要

本書は、以下の開発ツールを用いて開発したアプリケーションを Epson ePOS SDK for Android（以降、Epson ePOS SDK）で動作するように修正する方法を説明したマニュアルです。

- ePOS-Print SDK for Android（以降、ePOS-Print SDK）
- ePOS-Device SDK for Android（以降、ePOS-Device SDK）

ePOS-Print SDK、ePOS-Device SDK は、今後、新製品対応、新機能対応は行われません。本書を参考にして Epson ePOS SDK に移行（マイグレーション）してください。

マイグレーションの種類

Epson ePOS SDK へのマイグレーションには 2 種類の方法があります。

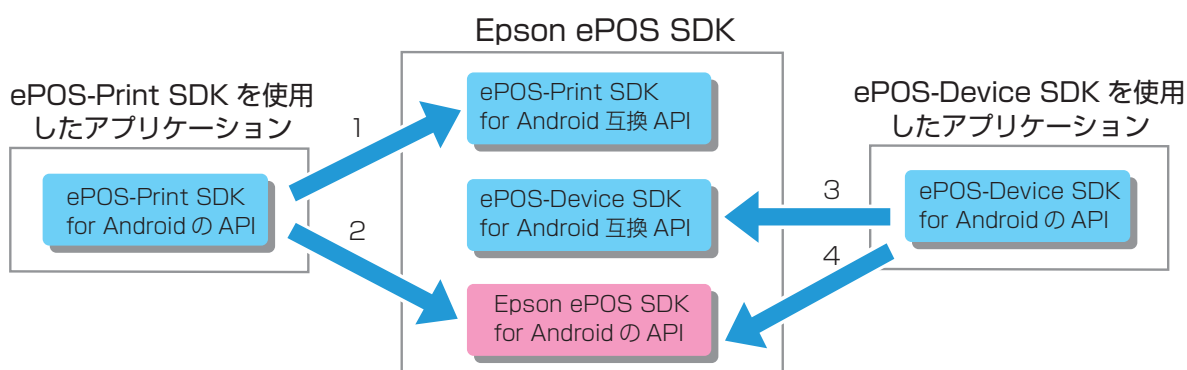
- Epson ePOS SDK の互換 API を使用したマイグレーション

Epson ePOS SDK には、ePOS-Print SDK 互換 API と ePOS-Device SDK 互換 API が含まれています。既存のアプリケーションのプログラムは修正せずに、構成ファイルを変更してビルドすることで、Epson ePOS SDK 対応のアプリケーションに移行できます。また、印刷などの基本的な機能を使用する場合は、TM プリンターの新製品にも対応できます。

- Epson ePOS SDK の API を使用したマイグレーション

既存のアプリケーションを、Epson ePOS SDK の API を使ったプログラムに修正することで、Epson ePOS SDK 対応のアプリケーションに移行できます。プログラムの修正量は多くなりますが、TM プリンターや周辺機器の新製品、新機能に対応できます。

マイグレーションの種類



- 1: ePOS-Print SDK から Epson ePOS SDK の ePOS-Print SDK 互換 API にマイグレーション
- 2: ePOS-Print SDK から Epson ePOS SDK の API にマイグレーション
- 3: ePOS-Device SDK から Epson ePOS SDK の ePOS-Device SDK 互換 API にマイグレーション
- 4: ePOS-Device SDK から Epson ePOS SDK の API にマイグレーション

新製品・新機能への対応方針

新製品・新機能	ePOS-Print SDK ePOS-Device SDK	Epson ePOS SDK	
			ePOS-Print SDK 互換 API ePOS-Device SDK 互換 API
エプソン製 TM プリンター、周辺機器	-	○	○
エプソン製 TM プリンター、周辺機器の新機能	-	○	- *
Android OS の新バージョン	-	○	○
タブレット端末、スマートフォンの新製品	-	○	○

○: 対応します -: 対応しません

* API の新規追加、変更が必要になる新機能には対応しません。

ePOS-Print SDK からのマイグレーション

ePOS-Print SDK を使用したアプリケーションを、Epson ePOS SDK 対応アプリケーションに移行する方法を説明します。

ePOS-Print SDK 互換 API を使用するマイグレーション

既存のアプリケーションのプログラムは修正せずに、構成ファイルを置き換えることで、Epson ePOS SDK 対応のアプリケーションに移行します。

マイグレーションの手順

変更手順は以下のとおりです。

手順		概要
1	SDK の置き換え	クラスファイルとライブラリーファイルの置き換え 「 SDK の置き換え 」参照
2	アプリケーションのビルド	SDK のファイルを置き換えたアプリケーションのプロジェクトをビルド

以上で ePOS-Print SDK 互換 API を使用したマイグレーションは完了です。

SDK の置き換え

アプリケーションプロジェクトに組み込まれている以下のファイルを、Epson ePOS SDK のファイルに置き換えます。

種類	ePOS-Print SDK	ePOS-Print SDK 互換 API
クラスファイル *	ePOS-Print.jar	ePOS2.jar
ライブラリー	libeposprint.so	libepos2.so

* ePOSEasySelect.jar のファイルは、Epson ePOS SDK のパッケージに含まれるファイルに置き換える必要はありません。

ePOS-Print SDK 互換 API を使用したアプリケーション開発

ePOS-Print SDK 互換 API を使用したアプリケーションを開発・保守する場合に必要な情報は、以下のマニュアルを参照してください。

□ ePOS-Print SDK 互換 API の仕様

「ePOS-Print SDK for Android ユーザーズマニュアル」

ePOS-Print SDK 互換 API の仕様は、ePOS-Print SDK API の仕様と同じです。

□ エプソン製 TM プリンターの新製品の機種情報・サポート API

本書「Epson ePOS SDK for Android マイグレーションガイド」の[付録](#)

Epson ePOS SDK の API を使用するマイグレーション

既存のアプリケーションのプログラムを修正して、Epson ePOS SDK 対応のアプリケーションに移行します。プログラムの修正量は多くなりますが、TM プリンターの新製品、新機能に対応していくことができます。

マイグレーションの手順

変更手順は以下のとおりです。

手順		概要
1	SDK の置き換え	クラスファイルとライブラリーファイルの置き換え 「 SDK の置き換え 」参照
2	パッケージ名の変更	ePOS-Print SDK のパッケージ名を Epson ePOS SDK のパッケージ名に変更 「 パッケージ名の変更 」参照
3	クラスの変更	ePOS-Print SDK のクラスを Epson ePOS SDK のクラスに変更 「 クラスの変更 」参照
4	API の変更	Epson ePOS SDK と ePOS-Print SDK で仕様の異なる API の変更やプログラムを修正 変更する内容は、以下のとおりです。 <input type="checkbox"/> 特定の機能を実現するためにプログラムを修正 以下の機能を修正してください。 <ul style="list-style-type: none"> • プリンターとの通信の接続と切断 「プリンターとの通信の接続と切断」参照 • 印刷 「印刷」参照 • ステータスの取得 「ステータスの取得」参照 • プリンターの検索 「プリンターの検索」参照 • ステータスの監視 「ステータスの監視」参照 <input type="checkbox"/> API の名称変更 API の名称を変更するもの (パラメーターの変更が必要な場合もあります) 「 API の名称変更 」参照 <input type="checkbox"/> API のパラメーター変更 API の名称は変わらないが、パラメーターの変更が必要なもの 「 API のパラメーター変更 」参照 <input type="checkbox"/> リスナーインターフェイスの変更 「 リスナーインターフェイスの変更 」参照
5	アプリケーションのビルド	修正したアプリケーションのプロジェクトをビルド

以上で Epson ePOS SDK の API を使用したマイグレーションは完了です。

SDK の置き換え

アプリケーションプロジェクトに組み込まれている以下のファイルを、Epson ePOS SDK のファイルに置き換えます。

種類	ePOS-Print SDK	Epson ePOS SDK
クラスファイル	ePOS-Print.jar	ePOS2.jar
	ePOSEasySelect.jar	ePOSEasySelect.jar *
ライブラリー	libeposprint.so	libepos2.so
	libeposeasyselect.so	libeposeasyselect.so *

* ファイル名は変わりません。Epson ePOS SDK のパッケージに含まれるファイルを使用してください。

パッケージ名の変更

アプリケーションのプロジェクトに設定されているパッケージ名を、Epson ePOS SDK のパッケージ名に変更します。

削除する ePOS-Print SDK パッケージ名

パッケージ名	説明
com.epson.epsonio	ePOS-Print SDK 互換 API を使用するマイグレーションに必要なパッケージです。 Epson ePOS SDK の API を使用するマイグレーションでは使用しませんので削除してください。

追加する Epson ePOS SDK パッケージ名

パッケージ名	説明
com.epson.epos2	Epson ePOS SDK の各クラスで使用する共用のパッケージ
com.epson.epos2.discovery	Epson ePOS SDK のプリンター検索用のパッケージ

変更するパッケージ名

種類	ePOS-Print SDK	Epson ePOS SDK
Print クラス Builder クラス	com.epson.eposprint	com.epson.epos2.printer

クラスの変更

ePOS-Device SDK から Epson ePOS SDK にマイグレーションする際に、以下のクラスの変更を行います。

アプリケーションのプロジェクトで使用している ePOS-Print SDK のクラスを、Epson ePOS SDK のクラスに変更します。

種類	ePOS-Print SDK	Epson ePOS SDK
印刷機能	Builder クラス	Printer クラス
	Print クラス	
プリンター検索	Finder クラス	Discovery クラス
例外処理	EposException クラス	Epos2Exception クラス
	EpsonIoException クラス	

アプリケーションのプロジェクトで使用している Print クラス内の一部の定義値を EasySelectDeviceType クラスに変更します。

ePOS-Print SDK	Epson ePOS SDK
Print.DEVTYPE_TCP	EasySelectDeviceType.TCP
Print.DEVTYPE_BLUETOOTH	EasySelectDeviceType.Bluetooth

プリンターとの通信の接続と切断

実行手順の違い

	ePOS-Print SDK	Epson ePOS SDK
接続		
切断		

プログラムの違い

❑ ePOS-Print SDK

```
Print printer = new Print();
try {
    printer.openPrinter(Print.DEVTYPE_TCP, "192.168.192.168", Print.TRUE,
        Print.PARAM_DEFAULT, Print.PARAM_DEFAULT);
    ... 処理 ...

    printer.closePrinter();
} catch (EposException e) {
    int errStatus = e.getErrorStatus();
}
```

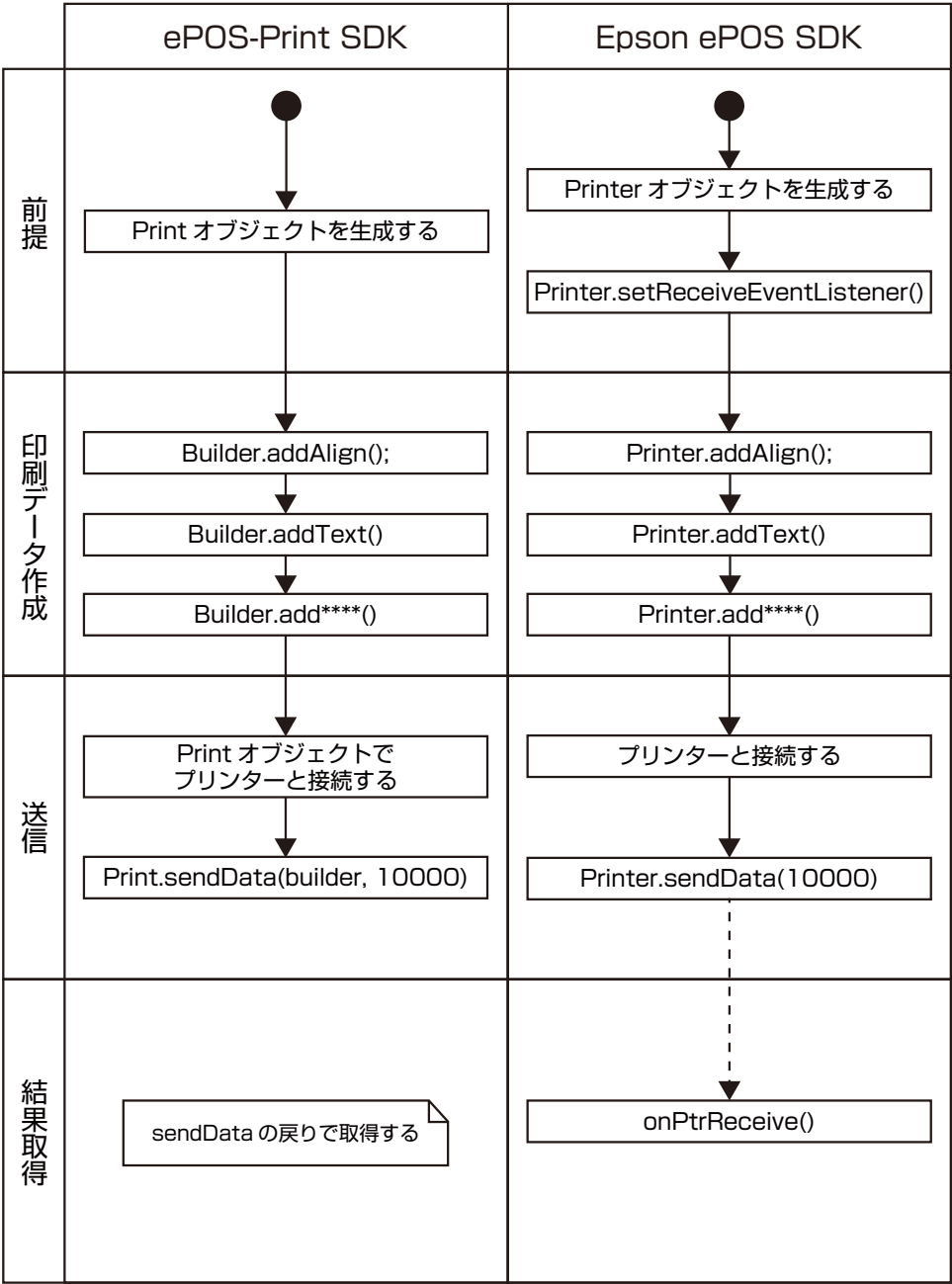
❑ Epson ePOS SDK

```
Printer printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK,
    getApplicationContext());
try {
    printer.connect("TCP:192.168.192.168", Printer.PARAM_DEFAULT);
    ... 処理 ...
    printer.disconnect();
} catch (Epos2Exception e) {
    int errStatus = e.getErrorStatus();
}
```

印刷

ePOS-Print SDK では、印刷データの送信処理の戻り値で印刷結果を取得していましたが、Epson ePOS SDK では、コールバックで印刷結果を取得します。

実行手順の違い



コールバック : ----▶

プログラムの違い

□ ePOS-Print SDK

```
Print printer = new Print();
int[] status = new int[1];
int[] battery = new int[1];
status[0] = 0;
battery[0] = 0;
try {
    Builder builder = new Builder("TM-T88V", Builder.MODEL_ANK);
    builder.addText("ABCDE");
    printer.openPrinter(Print.DEVTYPE_TCP, "192.168.192.168");
    printer.sendData(builder, 10000, status, battery);
    printer.closePrinter();
} catch (EposException e) {
    int errStatus = e.getErrorStatus();
    status[0] = e.getPrinterStatus();
    battery[0] = e.getBatteryStatus();
}
```

□ Epson ePOS SDK

```
import android.context.Context;
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

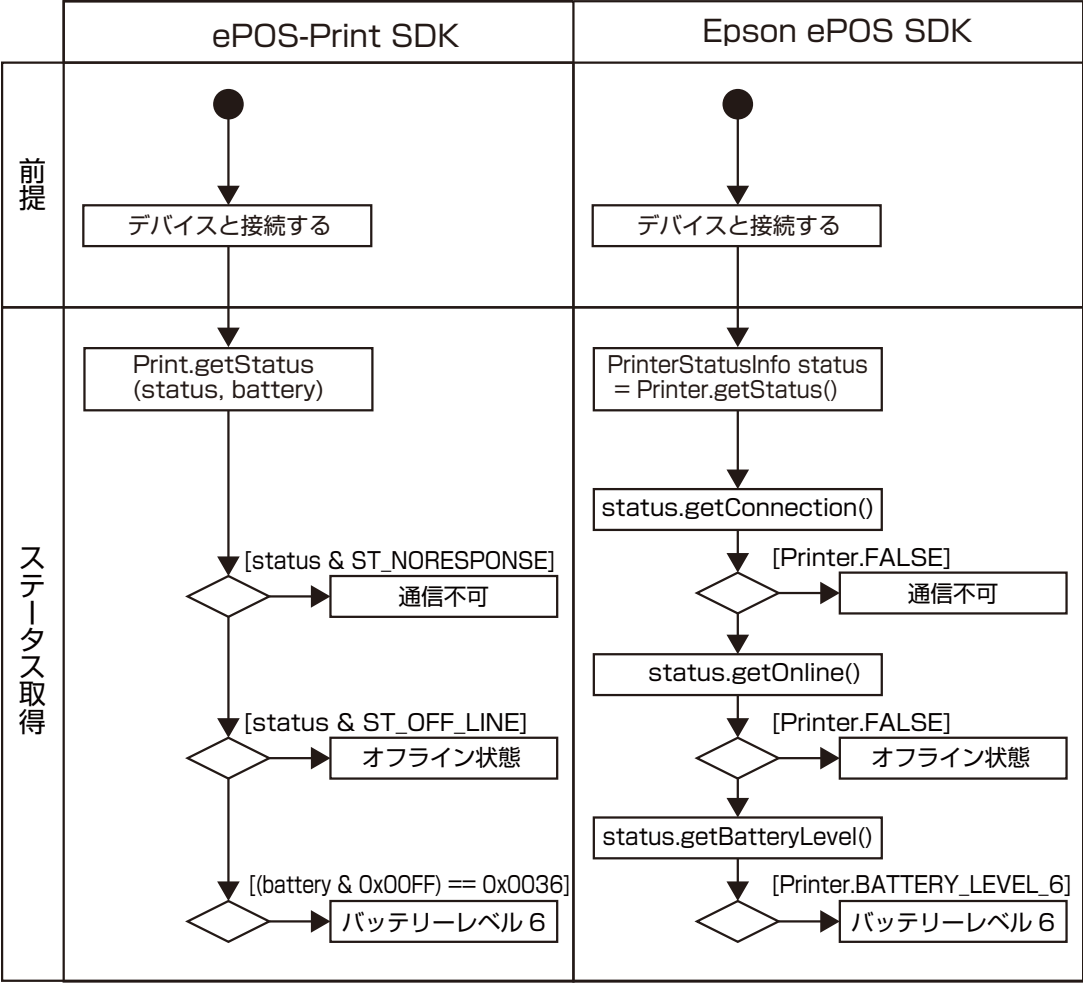
    private void openPrinter() {
        int errorStatus;
        try {
            printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK,
getApplicationContext());
            printer.setReceiveEventListener(this);
            . . . 接続 . . .
            printer.addText("ABCDE");
            printer.sendData(Printer.PARAM_DEFAULT);
        } catch (Epos2Exception e){
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onPtrReceive (Printer printerObj, int code, PrinterStatusInfo status, String
printJobId)
    {
        . . . 処理 . . .
    }
}
```

ステータスの取得

ePOS-Print SDK では、複数のプリンターステータスの組み合わせを戻り値で取得していましたが、Epson ePOS SDK では、PrinterStatusInfo 型のプロパティで各ステータスを取得します。

実行手順の違い



プログラムの違い

□ ePOS-Print SDK

```
Print printer = new Print();
int[] status = new int [1];
int[] battery = new int [1];
status[0] = 0;
battery[0] = 0;
try {
    printer.openPrinter(Print.DEVTYPE_TCP, "192.168.192.168", Print.TRUE,
        Print.PARAM_DEFAULT, Print.PARAM_DEFAULT);

    printer.getStatus(status, battery);
    ///Process///
    if(status[0] & Print.ST_NO_RESPONSE) {
        // no response
    }
    if(status[0] & Print.ST_OFF_LINE){
        // status offline
    }
    if((battery[0] & 0xFF) == 0x36){
        // battery level 6
    }
    printer.closePrinter();
} catch (EposException e) {
    int errStatus = e.getErrorStatus();
    status[0] = e.getPrinterStatus();
    battery[0] = e.getBatteryStatus();
}
```

□ Epson ePOS SDK

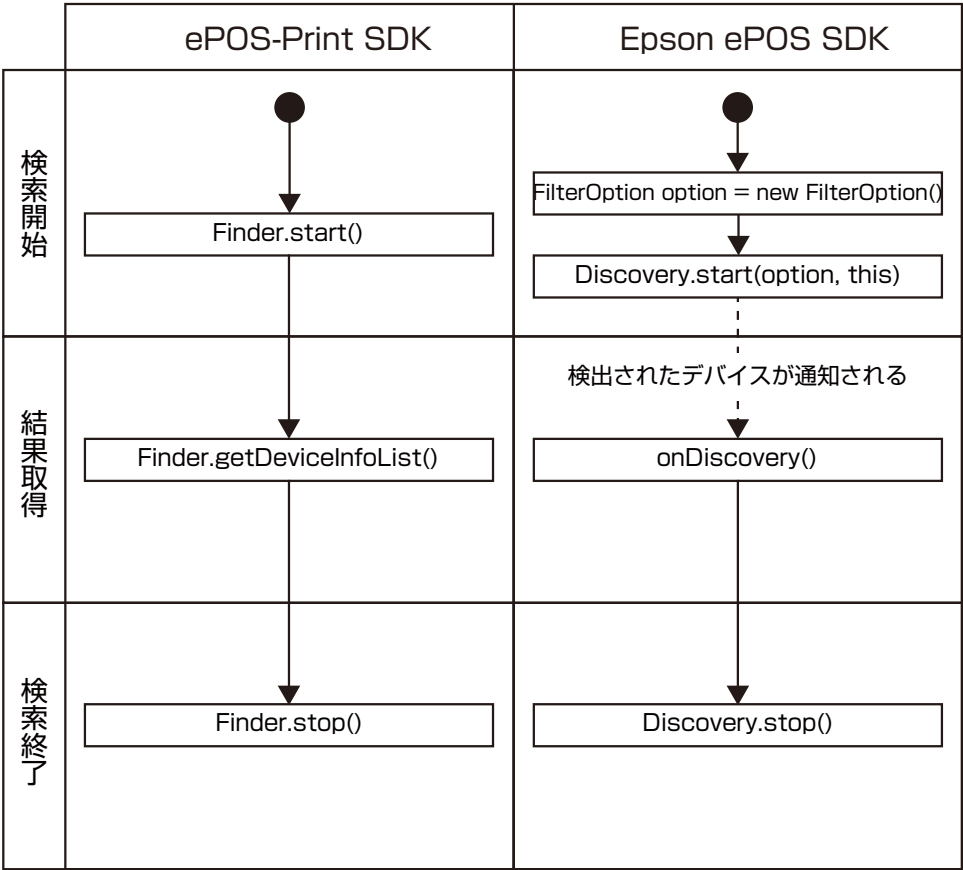
```
int errStatus;
Printer printer = NULL;
try {
    . . . 接続 . . .
    PrinterStatusInfo status = printer.getStatus();

    if(status.getConnection() != Printer.TRUE) {
        // no response
    }
    if(status.getOnline() != Printer.TRUE){
        // status offline
    }
    if(status.getBatteryLevel() == Printer.BATTERY_LEVEL_6){
        // battery level 6
    }
} catch (Epos2Exception e) {
    errStatus = e.getErrorStatus();
}
```


プリンターの検索

ePOS-Print SDK では、プリンターの検索結果を API で取得していましたが、Epson ePOS SDK では、フィルタリング設定しコールバックメソッドで取得します。

実行手順の違い



コールバック : ----▶

プログラムの違い

□ ePOS-Print SDK

```
import android.context.Context;

try{

    Finder.start(getBaseContext(), DevType.TCP, "255.255.255.0");

    DeviceInfo[] array = Finder.getDeviceInfoList(FilterOption.FILTER_NONE);
    String deviceName = NULL;
    String ipAddress = NULL;
    for(int index=0; index < array.length; index++) {
        deviceName = array[i].getDeviceName();
        ipAddress = array[i].getIpAddress();
    }

    Finder.stop();
} catch (EpsonIoException e) {
    int errStatus = e.getStatus();
}
```

□ Epson ePOS SDK

```
public class SampleActivity extends Activity implements DiscoveryListener {

    private void discovery() {
        int errorStatus;
        FilterOption option = new FilterOption();
        try{
            Discovery.start(getBaseContext(), option, this);

            . . . 検索中 . . .

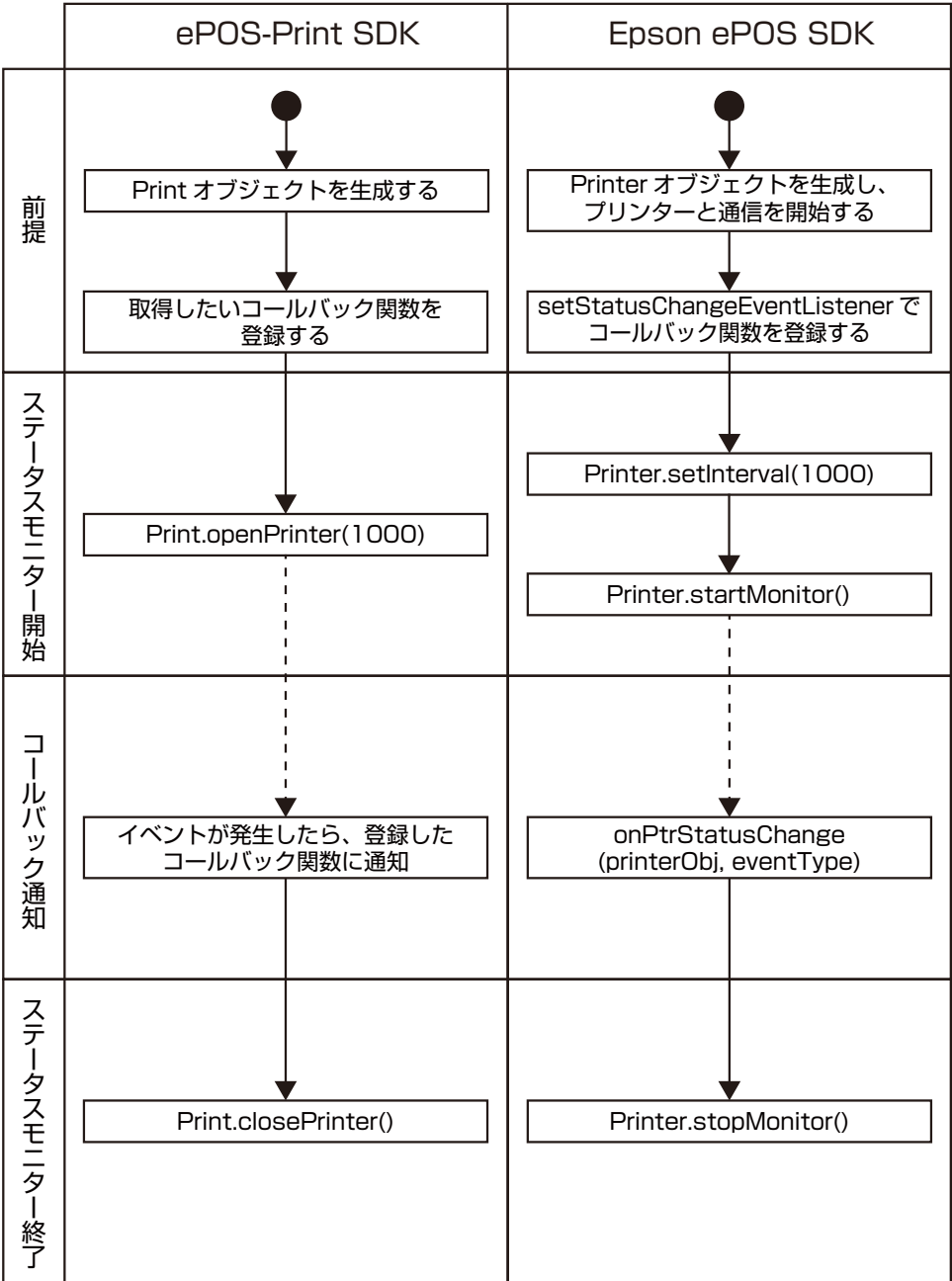
            Discovery.stop();
        } catch (Epos2Exception e) {
            int errStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onDiscovery(DeviceInfo deviceInfo)
    {
        String target = deviceInfo.getTarget();
    }
}
```

ステータスの監視

ePOS-Print SDK では、プリンターとの通信とステータスの監視を同時に開始していましたが、Epson ePOS SDK では、プリンターとの通信を開始した後にステータスの監視を開始します。

実行手順の違い



コールバック : ----▶

プログラムの違い

□ ePOS-Print SDK

```
public class SampleActivity extends Activity implements StatusChangeListener {
    @Override
    private void onStatusChangeEvent(String deviceName, int status) {
        . . . 処理 . . .
    }
    private void openPrinter() {
        Print printer = new Print();
        printer.setStatusChangeEventCallback(this);
        try {
            printer.openPrinter(Print.DEVTYPE_TCP, "192.168.192.168", Print.TRUE,
            Print.PARAM_DEFAULT, Print.PARAM_DEFAULT);
            . . . 処理 . . .

            printer.closePrinter();
        } catch (EposException e) {
            int errStatus = e.getErrorStatus();
        }
    }
}
```

□ Epson ePOS SDK

```
public class SampleActivity extends Activity implements StatusChangeListener {

    private Printer printer = NULL;

    private void openPrinter() {
        printer.setStatusChangeListener(this);
        . . . 接続 . . .
        printer.startMonitor();

        printer.stopMonitor();
    }

    @Override
    public void onPtrStatusChange(Printer printerObj, int eventType) {
        . . . 処理 . . .
    }
}
```

API の名称変更

ePOS-Print SDK から Epson ePOS SDK にマイグレーションする際に、名称を変更する必要がある API は下表のとおりです。API によっては複数の API が 1 つにまとめられたり、1 つの API が複数の API に分けられたりしたものが 있습니다。下表の API には、名称以外に仕様が変更になっている API もあります。変更内容は「ePOS-Print SDK for Android ユーザーズマニュアル」と「Epson ePOS SDK for Android ユーザーズマニュアル」で API を比較してください。

名称変更する API の一覧表

機能	ePOS-Print SDK	Epson ePOS SDK
クラスの初期化	Print	Printer
	Builder	
改行量設定を命令バッファに追加	addTextLineSpace	addLineSpace
文字倍角設定を命令バッファに追加	addTextDouble	addTextSize
文字印字位置設定を命令バッファに追加	addTextPosition	addHPosition
通信を開始	openPrinter	connect
ステータスの監視を開始		startMonitor
通信を終了	closePrinter	disconnect
ステータスの監視を終了		stopMonitor
プリンタステータスの通知先を登録	setStatusChangeEventCallback	setStatusChangeListener
オンラインイベントの通知先を登録	setOnlineEventCallback	
オフラインイベントの通知先を登録	setOfflineEventCallback	
無応答イベントの通知先を登録	setPowerOffEventCallback	
カバークローズイベントの通知先を登録	setCoverOkEventCallback	
カバーオープンイベントの通知先を登録	setCoverOpenEventCallback	
用紙ありイベントの通知先を登録	setPaperOkEventCallback	
用紙残量少イベントの通知先を登録	setPaperNearEndEventCallback	
用紙なしイベントの通知先を登録	setPaperEndEventCallback	
ドロアークローズイベントの通知先を登録	setDrawerClosedEventCallback	
ドロアークオープンイベントの通知先を登録	setDrawerOpenEventCallback	

機能	ePOS-Print SDK	Epson ePOS SDK
バッテリー残量なしイベントの通知先を登録	setBatteryLowEventCallback	setStatusChangeEventListe ner
バッテリー残量ありイベントの通知先を登録	setBatteryOkEventCallback	
バッテリースtatusの通知先を登録	setBatteryStatusChangeEventCallback	
プリンターの検索結果を取得	start	start
	getDeviceInfoList	

API のパラメーター変更

ePOS-Print SDK から Epson ePOS SDK にマイグレーションする際に、パラメーターの変更が必要な API は下表のとおりです。変更内容は「ePOS-Print SDK for Android ユーザーズマニュアル」と「Epson ePOS SDK for Android ユーザーズマニュアル」で API を比較してください。

パラメーターを変更する API の一覧表

API	パラメーターの変更内容
addTextAlign	Builder クラスから Printer クラスへの変更
addTextRotate	Builder クラスから Printer クラスへの変更
addTextLang	Builder クラスから Printer クラスへの変更
addTextFont	Builder クラスから Printer クラスへの変更
addTextSmooth	Builder クラスから Printer クラスへの変更
addTextSize	Builder クラスから Printer クラスへの変更
addTextStyle	Builder クラスから Printer クラスへの変更
addImage	Builder クラスから Printer クラスへの変更 compress の設定値が追加
addBarcode	Builder クラスから Printer クラスへの変更
addSymbol	Builder クラスから Printer クラスへの変更
addPageDirection	Builder クラスから Printer クラスへの変更
addPagePosition	Builder クラスから Printer クラスへの変更
addPageLine	Builder クラスから Printer クラスへの変更
addPageRectangle	Builder クラスから Printer クラスへの変更
addCut	Builder クラスから Printer クラスへの変更
addPulse	Builder クラスから Printer クラスへの変更
addSound	Builder クラスから Printer クラスへの変更
addLayout	Builder クラスから Printer クラスへの変更
sendData	timeout のみに変更
getStatus	パラメーターがなくなり、戻り値で値の取得
parseNFC	timeout パラメーターが追加
createQR	deviceType の設定値の名前が変更

リスナーインターフェイスの変更

ePOS-Print SDK から Epson ePOS SDK にマイグレーションする際に、変更・削除する必要のあるリスナーインターフェイスは下表のとおりです。

変更内容は「ePOS-Print SDK for Android ユーザーズマニュアル」と「Epson ePOS SDK for Android ユーザーズマニュアル」でリスナーの通知先メソッドを登録する API を比較してください。

変更するリスナーインターフェイス

機能	ePOS-Print SDK	Epson ePOS SDK
	イベントリスナー	
	通知先メソッド	
プリンタステータスの通知	public interface StatusChangeListener extends EventListener	public interface StatusChangeListener extends EventListener
	void onStatusChangeEvent(String deviceName, int status)	void onPtrStatusChange(Printer printerObj, int eventType)

削除する ePOS-Print SDK のリスナーインターフェイス

機能	イベントリスナー
	通知先メソッド
オンラインの通知	public interface OnlineEventListener extends EventListener
	void onOnlineEvent(String deviceName)
オフラインの通知	public interface OfflineEventListener extends EventListener
	void onOfflineEvent(String deviceName)
無応答の通知	public interface PowerOffEventListener extends EventListener
	void onPowerOffEvent(String deviceName)
カバークローズの通知	public interface CoverOkEventListener extends EventListener
	void onCoverOkEvent(String deviceName)
カバーオープンの通知	public interface CoverOpenEventListener extends EventListener
	void onCoverOpenEvent(String deviceName)
用紙ありの通知	public interface PaperOkEventListener extends EventListener
	void onPaperOkEvent(String deviceName)
用紙残量少の通知	public interface PaperNearEndEventListener extends EventListener
	void onPaperNearEndEvent(String deviceName)

機能	イベントリスナー
	通知先メソッド
用紙なしの通知	public interface PaperEndEventListener extends EventListener
	void onPaperEndEvent(String deviceName)
ドロアークローズの通知	public interface DrawerClosedEventListener extends EventListener
	void onDrawerClosedEvent(String deviceName)
ドロアークオープン通知	public interface DrawerOpenEventListener extends EventListener
	void onDrawerOpenEvent(String deviceName)
バッテリー残量なしの通知	public interface BatteryLowEventListener extends EventListener
	void onBatteryLowEvent(String deviceName)
バッテリー残量ありの通知	public interface BatteryOkEventListener extends EventListener
	void onBatteryOkEvent(String deviceName)
バッテリーステータスの通知	public interface BatteryStatusChangeListener extends EventListener
	void onBatteryStatusChangeEvent(String deviceName, int battery)

ePOS-Device SDK からのマイグレーション

ePOS-Device SDK を使用したアプリケーションを、Epson ePOS SDK 対応アプリケーションに移行する方法を説明します。

ePOS-Device SDK 互換 API を使用するマイグレーション

既存のアプリケーションのプログラムは修正せずに、構成ファイルを置き換えることで、Epson ePOS SDK 対応のアプリケーションに移行します。

マイグレーションの手順

変更手順は以下のとおりです。

手順		概要
1	SDK の置き換え	クラスファイルとライブラリーファイルの置き換え 「 SDK の置き換え 」参照
2	アプリケーションのビルド	SDK のファイルを置き換えたアプリケーションのプロジェクトをビルド

以上で ePOS-Device SDK 互換 API を使用したマイグレーションは完了です。

SDK の置き換え

アプリケーションプロジェクトに組み込まれている以下のファイルを、Epson ePOS SDK のファイルに置き換えます。

種類	ePOS-Device SDK	ePOS-Device SDK 互換 API
クラスファイル	ePOS-Device.jar	ePOS2.jar
ライブラリー	libeposdevice.so	libepos2.so

ePOS-Device SDK 互換 API を使用したアプリケーション開発

ePOS-Device SDK 互換 API を使用したアプリケーションを開発・保守する場合に必要な情報は、以下のマニュアルを参照してください。

ePOS-Device SDK 互換 API の仕様：「ePOS-Device SDK for Android ユーザーズマニュアル」

Epson ePOS SDK の API を使用するマイグレーション

既存のアプリケーションのプログラムを修正して、Epson ePOS SDK 対応のアプリケーションに移行します。プログラムの修正量は多くなりますが、TM プリンターや周辺機器の新製品、新機能に対応していくことができます。

マイグレーションの手順

変更手順は以下のとおりです。

手順		概要
1	SDK の置き換え	クラスファイルとライブラリーファイルの置き換え 「 SDK の置き換え 」参照
2	パッケージ名の変更	ePOS-Device SDK のパッケージ名を Epson ePOS SDK のパッケージ名に変更 「 パッケージ名の変更 」参照
3	クラスの変更	ePOS-Device SDK のクラスを Epson ePOS SDK のクラスに変更 「 クラスの変更 」参照
4	API の変更	Epson ePOS SDK と ePOS-Device SDK で仕様の異なる API の変更やプログラムを修正 変更する内容は、以下のとおりです。 <div> <div>□ 特定の機能を実現するためにプログラムを修正 以下の機能を修正してください。</div> <ul style="list-style-type: none"> • デバイスとの通信の接続と切断 「デバイスとの通信の接続と切断」参照 • 再接続の通知 「再接続通知」参照 • 印刷 「印刷」参照 • 強制送信 「強制送信」参照 • ステータスの取得 「ステータスの取得」参照 <div> <div>□ API の名称変更 API の名称を変更するもの (パラメーターの変更が必要な場合もあります) 「API の名称変更」参照</div> <div> <div>□ API のパラメーター変更 API の名称は変わらないが、パラメーターの変更が必要なもの 「API のパラメーター変更」参照</div> <div> <div>□ リスナーインターフェイスの変更 「リスナーインターフェイスの変更」参照</div> </div> </div> </div></div>
5	アプリケーションのビルド	修正したアプリケーションのプロジェクトをビルド

以上で Epson ePOS SDK の API を使用したマイグレーションは完了です。

SDK の置き換え

アプリケーションプロジェクトに組み込まれている以下のファイルを、Epson ePOS SDK のファイルに置き換えます。

種類	ePOS-Device SDK	Epson ePOS SDK
クラスファイル	ePOS-Device.jar	ePOS2.jar
ライブラリー	libeposdevice.so	libepos2.so

パッケージ名の変更

アプリケーションのプロジェクトに設定されているパッケージ名を、Epson ePOS SDK のパッケージ名に変更します。

変更するパッケージ名

種類	ePOS-Device SDK	Epson ePOS SDK
各クラスで使用する共用のパッケージ	com.epson.eposdevice	com.epson.epos2
Display クラス	com.epson.eposdevice.display	com.epson.epos2.linedisplay
Keyboard クラス	com.epson.eposdevice.keyboard	com.epson.epos2.keyboard
Printer クラス	com.epson.eposdevice.printer	com.epson.epos2.printer
Scanner クラス	com.epson.eposdevice.scanner	com.epson.epos2.barcodescanner
SimpleSerial クラス	com.epson.eposdevice.simpleserial	com.epson.epos2.simpleserial
CommBoxManagerクラス	com.epson.eposdevice.commbox	com.epson.epos2.commbox
CommBox クラス		

クラスの変更

アプリケーションのプロジェクトで使用している ePOS-Device SDK のクラスを、Epson ePOS SDK のクラスに変更します。

削除する ePOS-Device SDK のクラス名

クラス名	説明
Device クラス	Device クラスが持っている以下の機能は、各クラスの API に割り当てています。 <ul style="list-style-type: none"> • 通信経路の確立 • 通信経路の切り離し • 通信経路の確立状態を取得 • 管理者情報の取得 • 設置場所情報の取得 • 再接続処理開始イベントのコールバックメソッドを登録 • 再接続終了イベントのコールバックメソッドを登録 • ネットワーク切断イベントのコールバックメソッドを登録
CommBoxManager クラス	CommBox クラスに統合します。

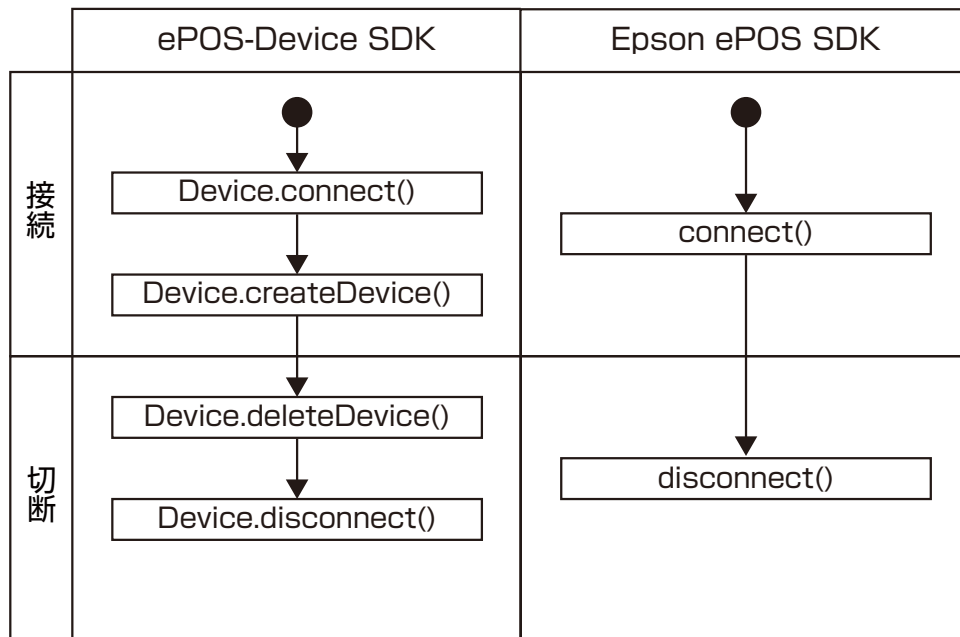
変更するクラス名

種類	ePOS-Device SDK	Epson ePOS SDK
デバイス制御	Display クラス	LineDisplay クラス
	Scanner クラス	BarcodeScanner クラス
例外	EposException クラス	Epos2Exception クラス

デバイスとの通信の接続と切断

ePOS-Device SDK では、ePOS-Device Service と接続した後に各デバイスとの通信を開始していましたが、Epson ePOS SDK では、デバイスごと通信を開始します。また ePOS-Device SDK では、デバイスと通信を終了した後に ePOS-Device Service との接続を終了していましたが、Epson ePOS SDK では、デバイスごとに通信を終了します。

実行手順の違い



プログラムの違い

□ ePOS-Device SDK

```
import android.context.Context;
public class SampleActivity extends Activity implements ConnectListener,
CreateDeviceListener, DeleteDeviceListener {

    private Device device = NULL;
    private Printer printer = NULL;

    private void openPrinter() {
        int errorStatus;
        try {
            device = new Device(getApplicationContext());
            device.connect("192.168.192.168", this);
        } catch (EposException e) {
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onConnect(String ipAddress, int code) {
        int errorStatus;
        if(code == EposCallbackCode.SUCCESS) {
            try {
                device.createDevice("local_printer", Device.DEV_TYPE_PRINTER, Device.FALSE,
Device.FALSE, this);
            } catch (EposException e) {
                errorStatus = e.getErrorStatus();
            }
        }
    }

    @Override
    public void onCreateDevice(String ipAddress, String deviceId,
        int deviceType, Object deviceObject, int code)
    {
        if(code == EposCallbackCode.SUCCESS) {
            if(deviceType == Device.DEV_TYPE_PRINTER) {
                printer = (Printer) deviceObject;
            }
        }
    }

    private void closePrinter() {
        int errorStatus;
        if(printer != NULL) {
            try {
                device.deleteDevice(printer, this);
            } catch (EposException e){
                errorStatus = e.getErrorStatus();
            }
        }
    }

    @Override
    public void onDeleteDevice(String ipAddress, String deviceId, int code)
    {
        int errorStatus;
        if(code == EposCallbackCode.SUCCESS) {
            try {
                device.disconnect();
            } catch (EposException e) {
                errorStatus = e.getErrorStatus();
            }
        }
    }
}
```

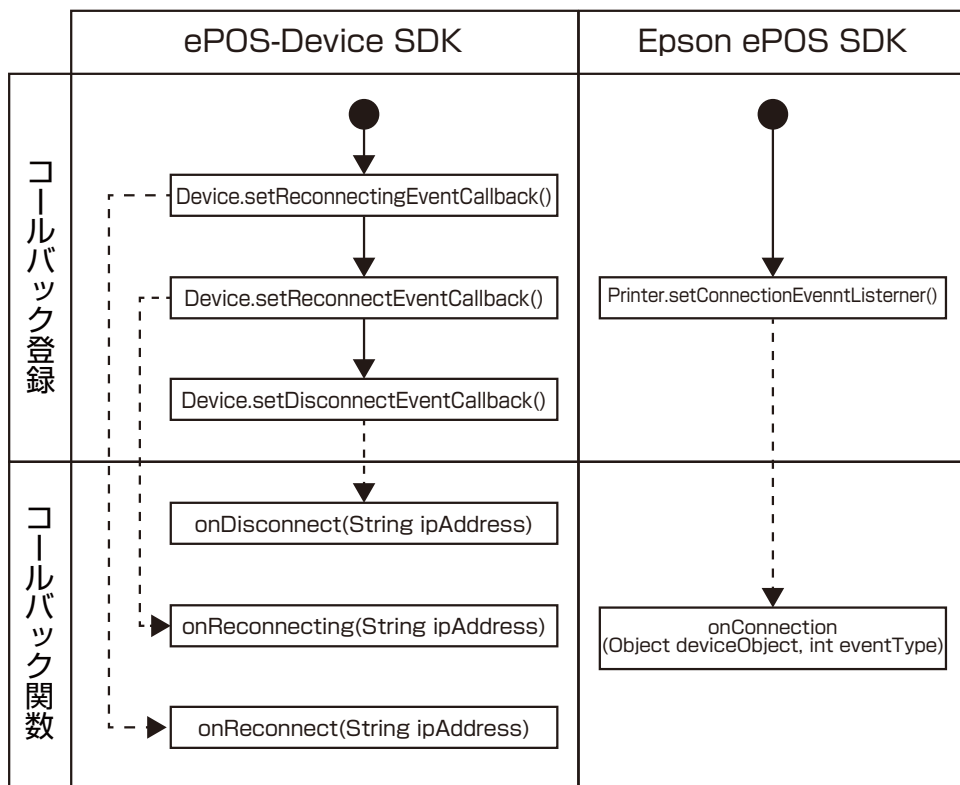
□ Epson ePOS SDK

```
Printer printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK,
    getApplicationContext());
try {
    printer.connect("TCP:192.168.192.168", Printer.PARAM_DEFAULT);
    . . . 処理 . . .
    printer.disconnect();
} catch (Epos2Exception e) {
    int errStatus = e.getErrorStatus();
}
```


再接続通知

ePOS-Device SDK では、通知の種類ごと API で登録していましたが、Epson ePOS SDK では、通知先登録 API が一つに統合され、通知先メソッドで通知の種類ごと処理します。

実行手順の違い



コールバック : ----▶

プログラムの違い

□ ePOS-Device SDK

```
import android.context.Context;
public class SampleActivity extends Activity implements ReconnectingListener,
ReconnectListener, DisconnectListener {

    private Device device = NULL;
    private Printer printer = NULL;

    private void openPrinter() {
        int errorStatus;
        try {
            device = new Device(getApplicationContext());
            device.setReconnectingEventCallback(this);
            device.setReconnectEventCallback(this);
            device.setDisconnectEventCallback(this);
            . . . 接続 . . .
        } catch (EposException e) {
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onReconnecting(String ipAddress) {
        . . . 再接続開始 . . .
    }
    @Override
    public void onReconnect(String ipAddress) {
        . . . 再接続完了 . . .
    }
    @Override
    public void onDisconnect(String ipAddress) {
        . . . 再接続失敗 . . .
    }
}
```

□ Epson ePOS SDK

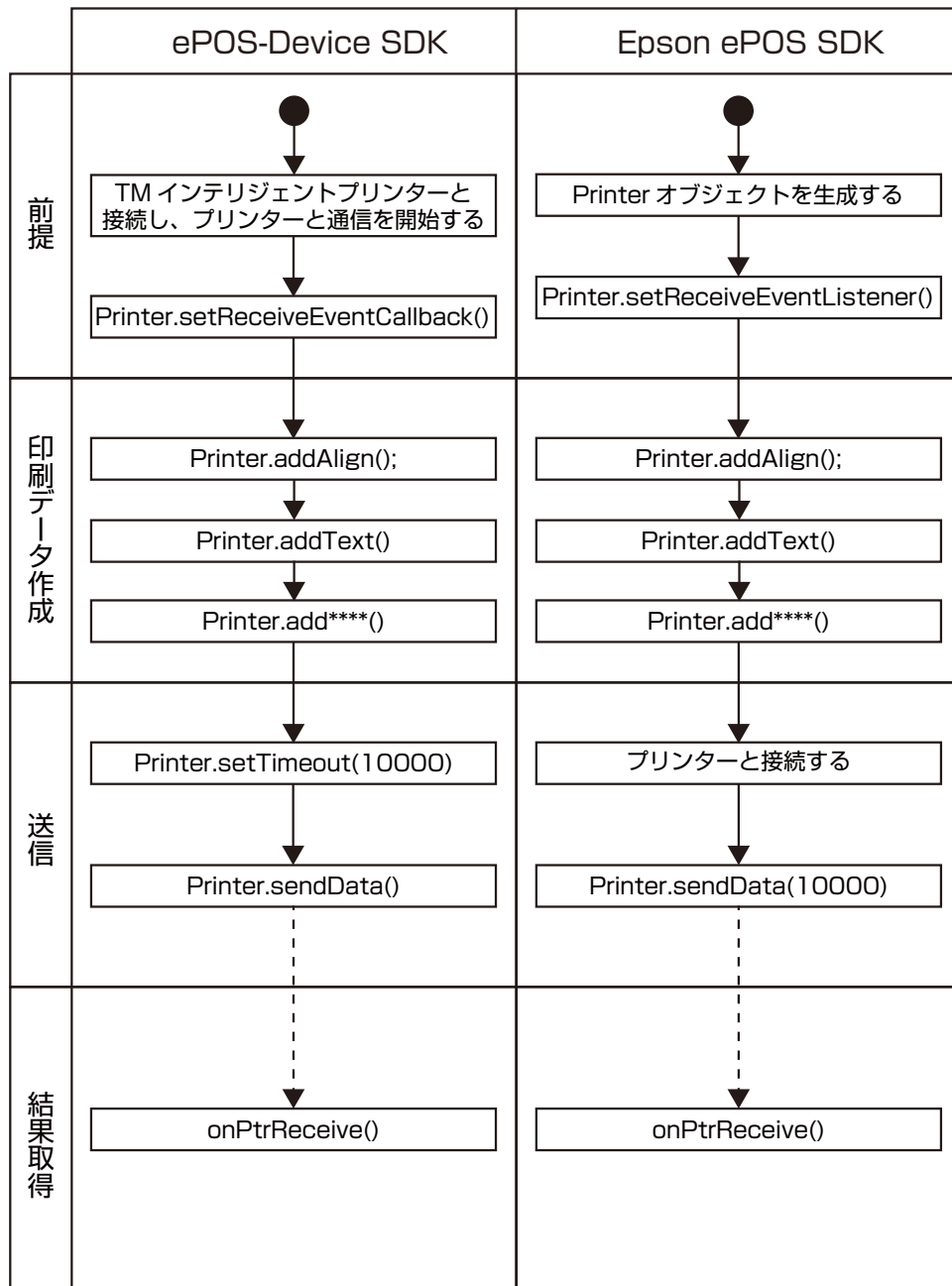
```
import android.widget.TextView;
public class SampleActivity extends Activity implements ConnectionListener {
    private Printer printer = NULL;
    private void openPrinter() {
        int errorStatus;
        try {
            printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK,
getApplicationContext());
            printer.setConnectionEventListener(this);
            . . . 接続 . . .
        } catch (Epos2Exception e) {
            errorStatus = e.getErrorStatus();
        }
    }
    @Override
    public void onConnection(Object deviceObj, int eventType) {
        if(eventType == Printer.EVENT_RECONNECTING) {
            . . . 再接続開始 . . .
        }
        if(eventType == Printer.EVENT_RECONNECT) {
            . . . 再接続完了 . . .
        }
        if(eventType == Printer.EVENT_DISCONNECT) {
            . . . 再接続失敗 . . .
        }
    }
}
```

印刷

ePOS-Device SDK では、プリンターと接続した後に印刷データ作成を行っていましたが、Epson ePOS SDK では、プリンターの接続前、後のどちらでも作成できます。
既存のプログラムの実行手順を修正しなくても、印刷できます。

実行手順の違い

Epson ePOS SDK の実行手順は、プリンターと接続する前に印刷データを作成する手順です。



コールバック : ----▶

プログラムの違い

□ ePOS-Device SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

    private void openPrinter() {
        . . . 接続 . . .

        printer.setReceiveEventCallback(this);
        int errorStatus;
        try {
            printer.addText("ABCDE");
            printer.sendData();
        } catch (EposException e){
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onPtrReceive (String ipAddress, String deviceId, int success, int code, int status,
int battery) {
        . . . 処理 . . .
    }
}
```

□ Epson ePOS SDK

以下のプログラムは、プリンターと接続する前に印刷データを作成しています。

```
import android.context.Context;
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

    private void openPrinter() {
        int errorStatus;
        try {
            printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK,
getApplicationContext());
            printer.setReceiveEventListener(this);
            printer.addText("ABCDE");
            . . . 接続 . . .
            printer.sendData(Printer.PARAM_DEFAULT);
        } catch (Epos2Exception e){
            errorStatus = e.getErrorStatus();
        }
    }

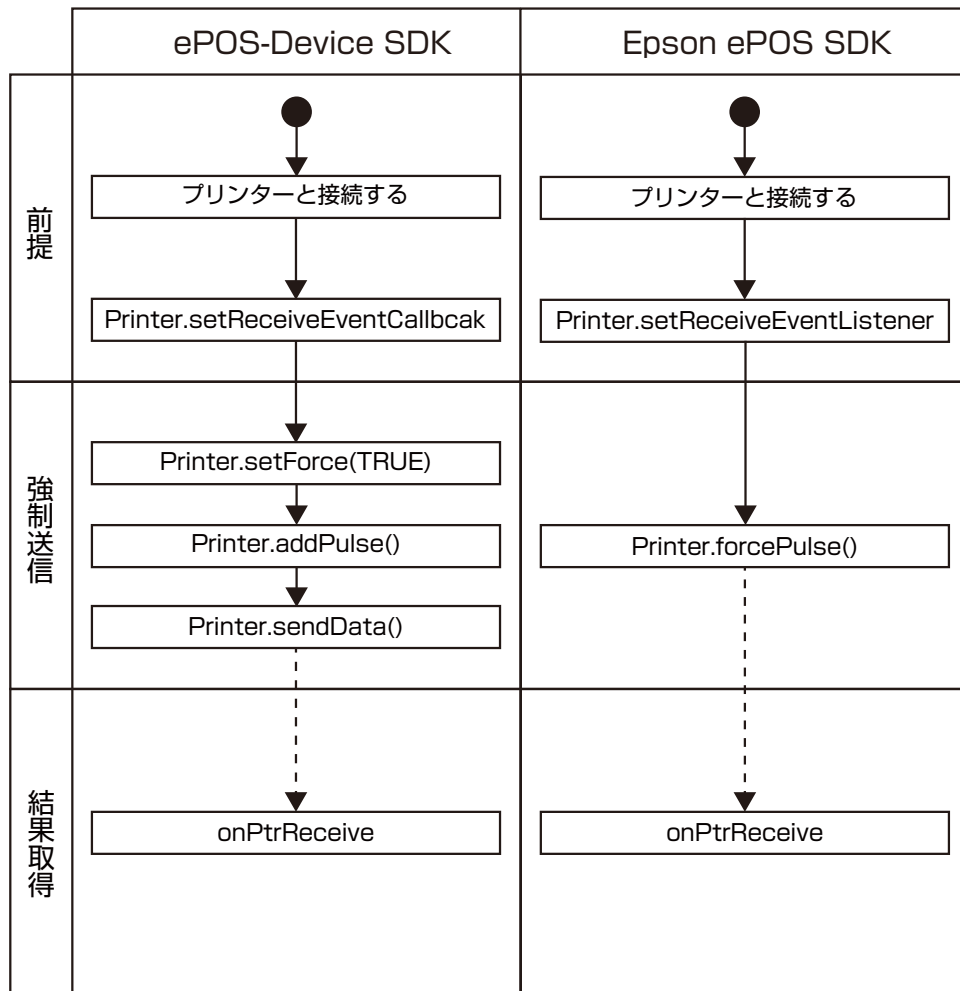
    @Override
    public void onPtrReceive (Printer printerObj, int code, PrinterStatusInfo status, String
printJobId)
    {
        . . . 処理 . . .
    }
}
```

強制送信

ePOS-Device SDK では、強制送信を 3 つの API を使って実行していましたが、Epson ePOS SDK では、1 つの API で実行します。

また ePOS-Device SDK では、強制送信はオフライン時のみ有効でしたが、Epson ePOS SDK では、オンライン、オフライン時の両方で使用できます。

実行手順の違い



コールバック : ----▶

プログラムの違い

□ ePOS-Device SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

    private void openPrinter() {
        . . . 接続 . . .

        printer.setReceiveEventCallback(this);
        int errorStatus;
        try {
            printer.setForce(false);
            printer.addPulse(Printer.DRAWER_1, Printer.PULSE_100);
            printer.sendData();
        } catch (EposException e){
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onPtrReceive (String ipAddress, String deviceId, int success, int code, int status,
int battery) {
        . . . 処理 . . .
    }
}
```

□ Epson ePOS SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

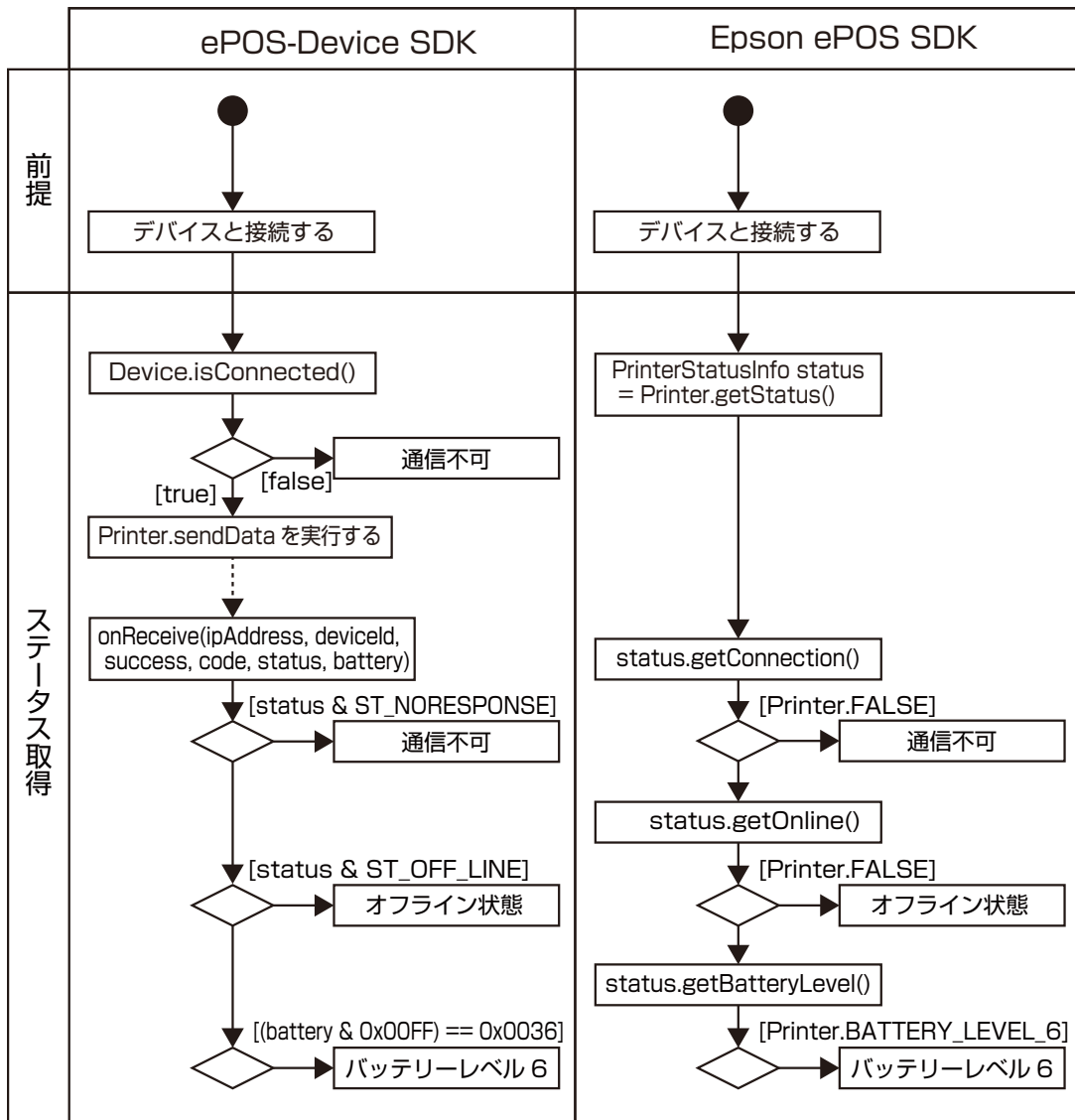
    private void openPrinter() {
        int errorStatus;
        try {
            . . . 接続 . . .
            printer.setReceiveEventListener(this);
            printer.forcePulse(Printer.DRAWER_2PIN, Printer.PULSE_100,
Printer.PARAM_DEFAULT);
        } catch (Epos2Exception e){
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onPtrReceive (Printer printerObj, int code, PrinterStatusInfo status, String
printJobId)
    {
        . . . 処理 . . .
    }
}
```

ステータスの取得

ePOS-Device SDK では、複数のプリンターステータスの組み合わせを戻り値で取得していましたが、Epson ePOS SDK では、PrinterStatusInfo 型のプロパティで各ステータスを取得します。

実行手順の違い



コールバック : ----▶

プログラムの違い

□ ePOS-Device SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

    private Device device = NULL;
    private Printer printer = NULL;

    private void getStatus() {
        . . . 接続 . . .
        if(device.isConnected() != true) {
            . . . 切断中 . . .
        }

        printer.setReceiveEventCallback(this);
        int errorStatus;
        try {
            printer.sendData();
        } catch (EposException e){
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onPtrReceive (String ipAddress, String deviceId, int success, int code, int status,
int battery) {
        if((status & Printer.ST_NO_RESPONSE) == Printer.ST_NO_RESPONSE) {
            // no response
        }
        if((status & Printer.ST_OFF_LINE) == Printer.ST_OFF_LINE){
            // status offline
        }
        if((battery & 0x00FF) == 0x0036){
            // battery level 6
        }
    }
}
```

□ Epson ePOS SDK

```
int errStatus;
Printer printer = NULL;
try {
    . . . 接続 . . .
    PrinterStatusInfo status = printer.getStatus();

    if(status.getConnection() != Printer.TRUE) {
        // no response
    }
    if(status.getOnline() != Printer.TRUE){
        // status offline
    }
    if(status.getBatteryLevel() == Printer.BATTERY_LEVEL_6){
        // battery level 6
    }

} catch (Epos2Exception e) {
    errStatus = e.getErrorStatus();
}
```


API の名称変更

ePOS-Device SDK から Epson ePOS SDK にマイグレーションする際に、名称を変更する必要がある API は下表のとおりです。API によっては複数の API が 1 つにまとめられたり、1 つの API が複数の API に分けられたりしたものが 있습니다。下表の API には、名称以外に仕様が変更になっている API もあります。変更内容は「ePOS-Device SDK for Android ユーザーズマニュアル」と「Epson ePOS SDK for Android ユーザーズマニュアル」で API を比較してください。

名称変更する API の一覧表

クラス		ePOS-Device SDK	Epson ePOS SDK
	機能		
クラス共通			
クラスの初期化		Device	Printer
			LineDisplay
			Keyboard
			BarcodeScanner
			SimpleSerial
			CommBox
デバイスのオブジェクトを取得		connect	connect
		createDevice	
デバイスのオブジェクトを破棄		disconnect	disconnect
		deleteDevice	
現在のステータス情報を取得		isConnected	getStatus
再接続処理開始イベントの通知先を登録		setReconnectingEventCallback	setConnectionEventListener
再接続終了イベントの通知先を登録		setReconnectEventCallback	
ネットワーク切断イベントの通知先を登録		setDisconnectEventCallback	

クラス		ePOS-Device SDK	Epson ePOS SDK
	機能		
Display クラス			
	表示領域の定義を命令バッファーに追加	createWindow	addCreateWindow
	表示領域の設定破棄を命令バッファーに追加	destroyWindow	addDestroyWindow
	表示領域の切り替えを命令バッファーに追加	setCurrentWindow	addSetCurrentWindow
	現在の表示領域の消去を命令バッファーに追加	clearWindow	addClearCurrentWindow
	カーソル位置を命令バッファーに追加	setCursorPosition	addSetCursorPosition
	表示領域内でカーソル位置を命令バッファーに追加	moveCursorPosition	addMoveCursorPosition
	カーソルの種類の変更を命令バッファーに追加	setCursorType	addSetCursorType
	マーキー表示を命令バッファーに追加	addMarquee	addMarqueeText
	表示の点滅情報を命令バッファーに追加	setBlink	addSetBlink
	表示輝度の情報を命令バッファーに追加	setBrightness	addSetBrightness
	時計の時刻を命令バッファーに追加	ShowClock	addShowClock
	ラインディスプレイを初期化	reset	addInitialze
	制御結果受信イベントの通知先を登録	SetReceiveEventCallback	setReceiveEventListener
Keyboard クラス			
	キー押下検出イベントの通知先を登録	setKeyPressEventCallback	setKeyPressEventListener
	文字列検出イベントの通知先を登録	setStringEventCallback	setReadStringEventListener

クラス		ePOS-Device SDK	Epson ePOS SDK
	機能		
Printer クラス			
改行量設定を命令バッファに追加	addTextLineSpace	addLineSpace	
文字倍角設定を命令バッファに追加	addTextSize	addTextSize	
	addTextDouble		
改行を命令バッファに追加	addFeedLine	addFeedLine	
	addFeed		
文字印字位置設定を命令バッファに追加	addTextPosition	addHPosition	
現在のステータスを取得	setOnlineEventCallback	getStatus	
ラスターイメージのハーフトーン処理方法を設定	halftone プロパティー	addImage	
ラスターイメージの明るさ補正値を設定	brightness プロパティー		
プリンターをリセット	reset	forceReset	
強制送信	force プロパティー	forceRecover	
		forcePulse	
		forceStopSound	
		forceCommand	
プリンターステータスの通知先を登録	setStatusChangeEventCall back	setStatusChangeEventList ener	
オンラインイベントの通知先を登録	setOnlineEventCallback		
オフラインイベントの通知先を登録	setOfflineEventCallback		
無応答イベントの通知先を登録	setPowerOffEventCallback		
カバークローズイベントの通知先を登録	setCoverOkEventCallback		
カバーオープンイベントの通知先を登録	setCoverOpenEventCallbac k		
用紙ありイベントの通知先を登録	setPaperOkEventCallback		
用紙残量少イベントの通知先を登録	setPaperNearEndEventCall back		
用紙なしイベントの通知先を登録	setPaperEndEventCallback		
ドロアークローズイベントの通知先を登録	setDrawerClosedEventCall back		

クラス		ePOS-Device SDK	Epson ePOS SDK
	機能		
Printer クラス			
	ドロアーオープンイベントの通知先を登録	setDrawerOpenEventCallback	setStatusChangeListener
	バッテリー残量なしイベントの通知先を登録	setBatteryLowEventCallback	
	バッテリー残量ありイベントの通知先を登録	setBatteryOkEventCallback	
	バッテリーステータスの通知先を登録	setBatteryStatusChangeEventCallback	
	応答ドキュメント受信イベントの通知先を登録	setReceiveEventCallback	
Scanner クラス			
	バーコードデータ入力イベントの通知先を登録	setDataEventCallback	setScanEventListener
SimpleSerial クラス			
	デバイスからの受信イベントの通知先を登録	setCommandReplyEventCallback	setReceiveEventListener
CommBox クラス			
	CommBox オブジェクトを取得	getCommBoxManager	connect
	コミュニケーションボックスを作成	openCommBox	
	コミュニケーションボックスを破棄	closeCommBox	disconnect
	メッセージをコミュニケーションボックスに送信	sendData	sendMessage
	コミュニケーションボックスのメッセージ受信の通知先を登録	setReceiveEventCallback	setCommandReplyEventCallback

API のパラメーター変更

ePOS-Device SDK から Epson ePOS SDK にマイグレーションする際に、パラメーターの変更が必要な API は下表のとおりです。変更内容は「ePOS-Device SDK for Android ユーザーズマニュアル」と「Epson ePOS SDK for Android ユーザーズマニュアル」で API を比較してください。

パラメーターを変更する API の一覧表

クラス		パラメーターの変更内容
API		
Display クラス		
addText	lang の設定値が追加	
addReverseText	lang の設定値が追加	
setReceiveEventListener	リスナー登録メソッドの code の値が追加	
Printer クラス		
sendData	timeout のみに変更	
addTextAlign	align の設定値が追加	
addTextRotate	rotate の設定値が追加	
addTextLang	lang の設定値が追加	
addTextFont	font の設定値が追加	
addTextSmooth	smooth の設定値が追加	
addTextSize	width/ height の設定値が追加	
addTextStyle	reverse/ ul/ em/ color の設定値が追加	
addImage	halftone/ brightness パラメーターが追加 compress の設定値が追加	
addBarcode	hri/ font/ width/ height の設定値が追加	
addSymbol	level/ width/ height/ size の設定値が追加	
addPageDirection	direction の設定値が追加	
addPagePosition	x/ y の設定値が追加	
setReceiveEventListener	リスナー登録メソッドの status/ battery パラメーターが統合 リスナー登録メソッドの code の値が削減	
interval プロパティー	設定値が追加	
CommBox クラス		
getCommHistory	リスナー登録メソッドの code の値が削減	

リスナーインターフェイスの変更

ePOS-Device SDK から Epson ePOS SDK にマイグレーションする際に、変更・削除する必要のあるリスナーインターフェイスは下表のとおりです。

変更内容は「ePOS-Device SDK for Android ユーザーズマニュアル」と「Epson ePOS SDK for Android ユーザーズマニュアル」でリスナーの通知先メソッドを登録する API を比較してください。

変更するリスナーインターフェイス

クラス	ePOS-Device SDK	Epson ePOS SDK
機能	イベントリスナー	
	通知先メソッド	
クラス共通		
再接続処理開始イベントの通知	public interface ReconnectingListener extends EventListener	public interface ConnectionListener extends EventListener
	onReconnecting	void onConnection(Object deviceObj, int eventType)
再接続終了イベントの通知	public interface ReconnectListener extends EventListener	「再接続処理開始イベントの通知」のリスナーインターフェイスに統合
	onReconnect	
ネットワーク切断イベントの通知	public interface DisconnectListener extends EventListener	
	onDisconnect	
Display クラス		
制御結果受信イベントの通知	public interface ReceiveListener extends EventListener	public interface ReceiveListener extends EventListener
	void onDspReceive(String ipAddress, String deviceId, int success, int code)	void onDispReceive (LineDisplay displayObj, int code)

クラス		ePOS-Device SDK	Epson ePOS SDK
機能	イベントリスナー		
	通知先メソッド		
Keyboard クラス			
キー押下検出イベントの通知	public interface KeyPressListener extends EventListener	public interface KeyPressListener extends EventListener	
	void onKbdKeyPress(String ipAddress, String deviceId, int keyCode, String ascii)	void onKbdKeyPress(Keyboard keyboardObj, int keyCode, String ascii)	
文字列検出イベントの通知	public interface StringListener extends EventListener	public interface ReadStringListener extends EventListener	
	void onKbdString(String ipAddress, String deviceId, String input, int prefix)	void onKbdReadString(Keyboar d keyboardObj, String readString, int prefix)	
Printer クラス			
プリンターステータスの通知	public interface StatusChangeEventListen er extends EventListener	public interface StatusChangeListener extends EventListener	
	void onStatusChangeEvent(Stri ng deviceName, int status)	void onPtrStatusChange(Printer printerObj, int eventType)	
応答ドキュメント受信イベントの通知	public interface ReceiveListener extends EventListener	public interface ReceiveListener extends EventListener	
	void onPtrReceive (String ipAddress, String deviceId, int success, int code, int status, int battery)	void onPtrReceive (Printer printerObj, int code, PrinterStatusInfo status, String printJobId)	
Scanner クラス			
バーコード情報の読み取りイベントの通知	public interface DataListener extends EventListener	public interface ScanListener extends EventListener	
	void onScanData(String ipAddress, String deviceId, String input)	void onScanData (BarcodeScanner scannerObj, String scanData)	

クラス		ePOS-Device SDK	Epson ePOS SDK
機能	イベントリスナー		
	通知先メソッド		
SimpleSerial クラス			
デバイスからの受信イベントの通知	public interface CommandReplyListener extends EventListener	public interface ReceiveListener extends EventListener	
	void onSimpleSerialCommandR epl (String ipAddress, String deviceId, byte[] data)	void onSimpleSerialReceive (SimpleSerial serialObj, byte[] data)	
CommBox クラス			
コミュニケーションボックスのメッ セージ受信イベントの通知	public interface ReceiveListener extends EventListener	public interface ReceiveListener extends EventListener	
	void onCommBoxReceive(String ipAddress, String senderId, String receiverId, String message)	void onCommBoxReceive(Com mBox commBoxObj, String senderId, String receiverId, String message)	

削除する ePOS-Device SDK のリスナーインターフェイス

クラス		
機能		イベントリスナー
		通知先メソッド
Printer クラス		
オンラインの通知		public interface OnlineEventListener extends EventListener
		void onOnlineEvent(String deviceName)
オフラインの通知		public interface OfflineEventListener extends EventListener
		void onOfflineEvent(String deviceName)
無応答の通知		public interface PowerOffEventListener extends EventListener
		void onPowerOffEvent(String deviceName)
カバークローズの通知		public interface CoverOkEventListener extends EventListener
		void onCoverOkEvent(String deviceName)

クラス		
機能	イベントリスナー	
	通知先メソッド	
Printer クラス		
カバーオープンの通知	public interface CoverOpenEventListener extends EventListener	
	void onCoverOpenEvent(String deviceName)	
用紙ありの通知	public interface PaperOkEventListener extends EventListener	
	void onPaperOkEvent(String deviceName)	
用紙残量少の通知	public interface PaperNearEndEventListener extends EventListener	
	void onPaperNearEndEvent(String deviceName)	
用紙なしの通知	public interface PaperEndEventListener extends EventListener	
	void onPaperEndEvent(String deviceName)	
ドロアークローズの通知	public interface DrawerClosedEventListener extends EventListener	
	void onDrawerClosedEvent(String deviceName)	
ドロアーオープンの通知	public interface DrawerOpenEventListener extends EventListener	
	void onDrawerOpenEvent(String deviceName)	
バッテリー残量なしの通知	public interface BatteryLowEventListener extends EventListener	
	void onBatteryLowEvent(String deviceName)	
バッテリー残量ありの通知	public interface BatteryOkEventListener extends EventListener	
	void onBatteryOkEvent(String deviceName)	
バッテリーステータスの通知	public interface BatteryStatusChangeListener extends EventListener	
	void onBatteryStatusChangeEvent(String deviceName, int battery)	
応答ドキュメント受信イベントの通知	public interface JobReceiveListener extends ReceiveListener	
	void onPtrReceive (String ipAddress, String deviceId, int success, int code, int status, int battery, String printJobId)	

付録

ePOS-Print SDK 互換 API

ePOS-Print SDK 互換 API でサポートしている、エプソン製 TM プリンターの新製品の機種情報とサポート API について説明します。

プリンターごとのサポート API 一覧

各プリンターのサポート API を一覧表で掲載します。

- ：対応している。
- ：対応していない。

API	TM-m10	TM-m30	TM-P80	TM-T88VI
addTextAlign	○	○	○	○
addTextLineSpace	○	○	○	○
addTextRotate	○	○	○	○
addText	○	○	○	○
addTextLang	○	○	○	○
addTextFont	○	○	○	○
addTextSmooth	○	○	○	○
addTextDouble	○	○	○	○
addTextSize	○	○	○	○
addTextStyle	○	○	○	○
addTextPosition	○	○	○	○
addFeedUnit	○	○	○	○
addFeedLine	○	○	○	○
addImage	○	○	○	○
addImage(旧フォーマット)	○	○	○	○
addImage(旧フォーマット)	○	○	○	○
addLogo	○	○	○	○
addBarcode	○	○	○	○
addSymbol	○	○	○	○

API	TM-m10	TM-m30	TM-P80	TM-T88VI
addPageBegin	○	○	○	○
addPageEnd	○	○	○	○
addPageArea	○	○	○	○
addPageDirection	○	○	○	○
addPagePosition	○	○	○	○
addPageLine	-	-	○	○
addPageRectangle	-	-	○	○
addCut	○	○	○	○
addPulse	○	○	-	○
addSound	○	○	○	○
addSound(旧フォーマット)	○	○	○	○
addFeedPosition	-	-	○	○
addLayout	-	-	○	○
addCommand	○	○	○	○

TM-m10

TM-m10 の機種情報は以下のとおりです。

		58 mm
解像度		203 x 203 dpi
言語		<ul style="list-style-type: none"> • ANK モデル • 日本語モデル • 繁体字中国語モデル
印字幅		420 ドット
印字桁数	フォント A	ANK: 35 桁 / 漢字: 17 桁
	フォント B	ANK: 42 / 漢字: 21 桁
	フォント C	ANK: 46 桁
文字サイズ	フォント A	ANK: 12 x 24 ドット / 漢字: 24 x 24 ドット
	フォント B	ANK: 10 x 24 ドット / 漢字: 20 x 24 ドット
	フォント C	ANK: 9 x 17 ドット
文字のベースライン	フォント A	文字の上端から 21 ドット目
	フォント B	文字の上端から 21 ドット目
	フォント C	文字の上端から 16 ドット目
初期改行量		30 ドット
色指定		第 1 色
ページモード初期領域		420 x 2400 ドット
ページモード最大領域		420 x 2400 ドット
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbolology not supported)
用紙のカット		カット / フィードカット
ドロアーキック		サポート
ブザー		オプション (パターン A ~ パターン E, エラー, 用紙なし, ストップ)
バッテリー		非サポート
Bluetooth 接続		TM-m10 Bluetooth モデルのみサポート

API 情報

API については、「ePOS-Print SDK for Android ユーザーズマニュアル」を参照してください。

API の差分情報は以下のとおりです。

API	パラメーター	指定可能な設定値	説明
Builder	printerModel	“TM-m10”	<ul style="list-style-type: none"> TM-m10 USB モデル TM-m10 Ethernet モデル TM-m10 Wi-Fi モデル TM-m10 <i>Bluetooth</i> モデル
	lang	Builder.MODEL_ANK	ANK モデル
		Builder.MODEL_JAPANESE	日本語モデル
		Builder.MODEL_TAIWAN	繁体字中国語モデル
addTextFont	font	Builder.FONT_A	フォント A
		Builder.FONT_B	フォント B
		Builder.FONT_C	フォント C
addImage	mode	Builder.MODE_MONO	モノクロ (2 階調)
		Builder.MODE_GRAY16	多階調 (16 階調)
		Builder.PARAM_DEFAULT	既定値 (モノクロ (2 階調))
	compress	Builder.COMPRESS_DEFLATE	画像を圧縮する。
		Builder.COMPRESS_NONE	画像を非圧縮する。
		Builder.PARAM_DEFAULT	既定値 (画像を非圧縮する。)

TM-m30

TM-m30 の機種情報は以下のとおりです。

		58 mm	80 mm
解像度		203 x 203 dpi	
言語		<ul style="list-style-type: none"> • ANK モデル • 日本語モデル • 簡体字中国語モデル • 繁体字中国語モデル • 韓国語モデル • 南アジアモデル 	
印字幅		420 ドット	576 ドット
印字桁数	フォント A	ANK 35 桁 / 漢字 17 桁	ANK 48 桁 / 漢字 24 桁
	フォント B	ANK 42 桁 / 漢字 21 桁	ANK 57 桁 / 漢字 28 桁
	フォント C	ANK 46 桁	ANK 64 桁
文字サイズ	フォント A	ANK 12 x 24 ドット / 漢字 24 x 24 ドット	
	フォント B	ANK 10 x 24 ドット / 漢字 20 x 24 ドット	
	フォント C	ANK 9 x 17 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 21 ドット目	
	フォント C	文字の上端から 16 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		420 x 2400 ドット	576 x 2400 ドット
ページモード最大領域		420 x 2400 ドット	576 x 2400 ドット
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded	
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbolism not supported)	
用紙のカット		カット / フィードカット	
ドロアーキック		サポート	
ブザー		オプション (パターン A ~ パターン E, エラー, 用紙なし, ストップ)	
バッテリー		非サポート	
Bluetooth 接続		TM-m30 Bluetooth モデルのみサポート	

API 情報

API については、「ePOS-Print SDK for Android ユーザーズマニュアル」を参照してください。

API の差分情報は以下のとおりです。

API	パラメーター	指定可能な設定値	説明
Builder	printerModel	“TM-m30”	<ul style="list-style-type: none"> TM-m30 標準モデル TM-m30 <i>Bluetooth</i> モデル
	lang	Builder.MODEL_ANK	ANK モデル
		Builder.MODEL_JAPANESE	日本語モデル
		Builder.MODEL_CHINESE	簡体字中国語モデル
		Builder.MODEL_TAIWAN	繁体字中国語モデル
		Builder.MODEL_KOREAN	韓国語モデル
		Builder.MODEL_SOUTHASI A	南アジアモデル
addTextFont	font	Builder.FONT_A	フォント A
		Builder.FONT_B	フォント B
		Builder.FONT_C	フォント C
addImage	mode	Builder.MODE_MONO	モノクロ (2 階調)
		Builder.MODE_GRAY16	多階調 (16 階調)
		Builder.PARAM_DEFAULT	既定値 (モノクロ (2 階調))
	compress	Builder.COMPRESS_DEFLA TE	画像を圧縮する。
		Builder.COMPRESS_NONE	画像を非圧縮する。
		Builder.PARAM_DEFAULT	既定値 (画像を非圧縮する。)

TM-P80

TM-P80 の機種情報は以下のとおりです。

		48 桁モード	42 桁モード
解像度		203 x 203 dpi	
言語		<ul style="list-style-type: none"> • ANK モデル • 日本語モデル • 繁体字中国語モデル 	
印字幅		576 ドット	546 ドット
印字桁数	フォント A	ANK 48 桁 / 漢字 24 桁	ANK 42 桁 / 漢字 21 桁
	フォント B	ANK 57 桁 / 漢字 28 桁	ANK 54 桁 / 漢字 27 桁
	フォント C	ANK 72 桁 / 漢字 36 桁	ANK 68 桁 / 漢字 34 桁
文字サイズ	フォント A	ANK 12 x 24 ドット 漢字 24 x 24 ドット	ANK 13 x 24 ドット 漢字 26 x 24 ドット
	フォント B	ANK 10 x 24 ドット / 漢字 20 x 24 ドット	
	フォント C	ANK 8 x 16 ドット / 漢字 16 x 16 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 21 ドット目	
	フォント C	文字の上端から 15 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		576 x 1662 ドット	
ページモード最大領域		576 x 1662 ドット	
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded	
2 次元シンボル		PDF417, QR Code, MaxiCode, Data Matrix, Aztec Code, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composite Symbology not supported)	
用紙のカット		カット / フィードカット	
ドロアーキック		非サポート	
ブザー		オプション (パターン 1 ~ パターン 10、ストップ)	
バッテリー		サポート	
Bluetooth 接続		TM-P80 Bluetooth モデルのみサポート	

API 情報

API については、「ePOS-Print SDK for Android ユーザーズマニュアル」を参照してください。

API の差分情報は以下のとおりです。

API	パラメーター	指定可能な設定値	説明
Builder	printerModel	"TM-P80"	<ul style="list-style-type: none"> TM-P80 Wi-Fi モデル TM-P80 <i>Bluetooth</i> モデル
	lang	Builder.MODEL_ANK	ANK モデル
		Builder.MODEL_JAPANESE	日本語モデル
		Builder.MODEL_TAIWAN	繁体字中国語モデル
addTextFont	font	Builder.FONT_A	フォント A
		Builder.FONT_B	フォント B
		Builder.FONT_C	フォント C
addTextSize	width	1 ～ 4 の整数	横方向の倍率を指定
		Builder.PARAM_UNSPECIFIED	設定を変更しない
	height	1 ～ 4 の整数	縦方向の倍率を指定
		Builder.PARAM_UNSPECIFIED	設定を変更しない
addImage	mode	Builder.MODE_MONO	モノクロ (2 階調)
		Builder.PARAM_DEFAULT	既定値 (モノクロ (2 階調))
	compress	Builder.COMPRESS_DEFLATE	画像を圧縮する。
		Builder.COMPRESS_NONE	画像を非圧縮する。
		Builder.PARAM_DEFAULT	既定値 (画像を非圧縮する。)

TM-T88VI

TM-T88VI の機種情報は以下のとおりです。

		58 mm 仕様	80 mm 仕様
解像度		180 x 180 dpi	
言語		<ul style="list-style-type: none"> • ANK モデル • 日本語モデル • 簡体字中国語モデル • 繁体字中国語モデル • 韓国語モデル • 南アジアモデル 	
印字幅		360 ドット	512 ドット
印字桁数	フォント A	ANK: 30 桁、漢字: 15 桁	ANK: 42 桁、漢字: 21 桁
	フォント B	ANK: 40 桁	ANK: 56 桁
文字サイズ	フォント A	ANK: 12 x 24 ドット / 漢字: 24 x 24 ドット	
	フォント B	ANK: 9 x 17 ドット	
文字のベースライン	フォント A	文字の上端から 21 ドット目	
	フォント B	文字の上端から 16 ドット目	
初期改行量		30 ドット	
色指定		第 1 色	
ページモード初期領域		360 x 831 ドット	512 x 831 ドット
ページモード最大領域		360 x 2400 ドット	512 x 2400 ドット
バーコード		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded	
2 次元シンボル		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbology は非サポート)	
用紙のカット		カット、フィードカット	
ドロアーキック		サポート	
ブザー		オプション (パターン A ~ パターン E、エラー、紙なし、停止)	
バッテリー		非サポート	
Bluetooth 接続		TM-T88VI Bluetooth モデルのみサポート	

API 情報

API については、「ePOS-Print SDK for Android ユーザーズマニュアル」を参照してください。

API の差分情報は以下のとおりです。

API	パラメーター	指定可能な設定値	説明
Builder	printerModel	"TM-T88VI"	TM-T88VI
	lang	Builder.MODEL_ANK	ANK モデル
		Builder.MODEL_JAPANESE	日本語モデル
		Builder.MODEL_CHINESE	簡体字中国語モデル
		Builder.MODEL_TAIWAN	繁体字中国語モデル
		Builder.MODEL_KOREAN	韓国語モデル
		Builder.MODEL_SOUTHASI A	南アジアモデル
addTextFont	font	Builder.FONT_A	フォント A
		Builder.FONT_B	フォント B
addImage	mode	Builder.MODE_MONO	モノクロ (2 階調)
		Builder.MODE_GRAY16	多階調 (16 階調)
		Builder.PARAM_DEFAULT	既定値 (モノクロ (2 階調))
	compress	Builder.COMPRESS_DEFLA TE	画像を圧縮する。
		Builder.COMPRESS_NONE	画像を非圧縮する。
		Builder.PARAM_DEFAULT	既定値 (画像を非圧縮する。)