

Epson ePOS SDK for Android

Migration Guide

Migration Overview

Migration from ePOS-Print SDK

Migration from ePOS-Device SDK

Appendix

Cautions

- No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Seiko Epson Corporation.
- The contents of this document are subject to change without notice. Please contact us for the latest information.
- While every precaution has been taken in the preparation of this document, Seiko Epson Corporation assumes no responsibility for errors or omissions.
- Neither is any liability assumed for damages resulting from the use of the information contained herein.
- Neither Seiko Epson Corporation nor its affiliates shall be liable to the purchaser of this product or third parties for damages, losses, costs, or expenses incurred by the purchaser or third parties as a result of: accident, misuse, or abuse of this product or unauthorized modifications, repairs, or alterations to this product, or (excluding the U.S.) failure to strictly comply with Seiko Epson Corporation's operating and maintenance instructions.
- Seiko Epson Corporation shall not be liable against any damages or problems arising from the use of any options or any consumable products other than those specified as Original Epson Products or Epson Approved Products by Seiko Epson Corporation.

Trademarks

EPSON is a registered trademark of Seiko Epson Corporation.

Exceed Your Vision is registered trademark or trademark of Seiko Epson Corporation.

Android™ is a trademark of Google Inc.

The *Bluetooth*® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Seiko Epson Corporation is under license.

All other trademarks are the property of their respective owners and used for identification purpose only.

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

Restriction of Use

When this product is used for applications requiring high reliability/safety, such as transportation devices related to aviation, rail, marine, automotive, etc.; disaster prevention devices; various safety devices, etc.; or functional/precision devices, etc., you should use this product only after giving consideration to including fail-safes and redundancies into your design to maintain safety and total system reliability. Because this product was not intended for use in applications requiring extremely high reliability/safety, such as aerospace equipment, main communication equipment, nuclear power control equipment, or medical equipment related to direct medical care, etc., please make your own judgment on this product's suitability after a full evaluation.

Contents

■ Restriction of Use	3
■ Contents.....	4
<hr/>	
<i>Migration Overview.....</i>	6
■ Migration Types.....	6
<hr/>	
<i>Migration from ePOS-Print SDK.....</i>	8
■ Migration with ePOS-Print SDK-compatible APIs	8
Migration procedure.....	8
SDK file replacement.....	8
Application development with ePOS-Print SDK-compatible APIs.....	8
■ Migration with the Epson ePOS SDK APIs	9
Migration procedure.....	9
SDK file replacement.....	10
Changing package names	10
Changing classes.....	11
Establishing and cutting the communication with the printer.....	12
Printing	13
Obtaining status.....	15
Printer search.....	17
Monitoring of the status.....	19
Changing API name.....	21
Changing API parameters	23
Changing Listener Interfaces.....	24
<hr/>	
<i>Migration from ePOS-Device SDK.....</i>	25
■ Migration with ePOS-Device SDK-compatible APIs	25
Migration procedure.....	25
SDK file replacement.....	25
Application development with ePOS-Device SDK-compatible APIs.....	25
■ Migration with the Epson ePOS SDK APIs	26
Migration procedure.....	26
SDK file replacement.....	27
Changing package names	27
Changing classes.....	28
Establishing and cutting the communication with the device	29
Reconnection notifications	32
Printing	35
Forced sending	37
Obtaining status.....	39

Changing API names.....	41
Changing API parameters	44
Changing Listener interfaces	45

Appendix..... 48

■ ePOS-Print SDK-compatible API..... 48

A list of support APIs for each printer model.....	48
TM-m10.....	50
TM-m30.....	52
TM-T60	54
TM-T88VI.....	55
TM-T88VI-iHUB	57

Migration Overview

This manual explains how to modify applications created with following development tools in order to run these applications in Epson ePOS SDK for Android ("Epson ePOS SDK").

- ePOS-Print SDK for Android ("ePOS-Print SDK")
- ePOS-Device SDK for Android ("ePOS-Device SDK")

ePOS-Print SDK and ePOS-Device SDK will no longer be optimized for new products and new functions. Refer to this document to perform migration to ePOS SDK-compatible application.

Migration Types

The following 2 methods are available for migration to Epson ePOS SDK-compatible application.

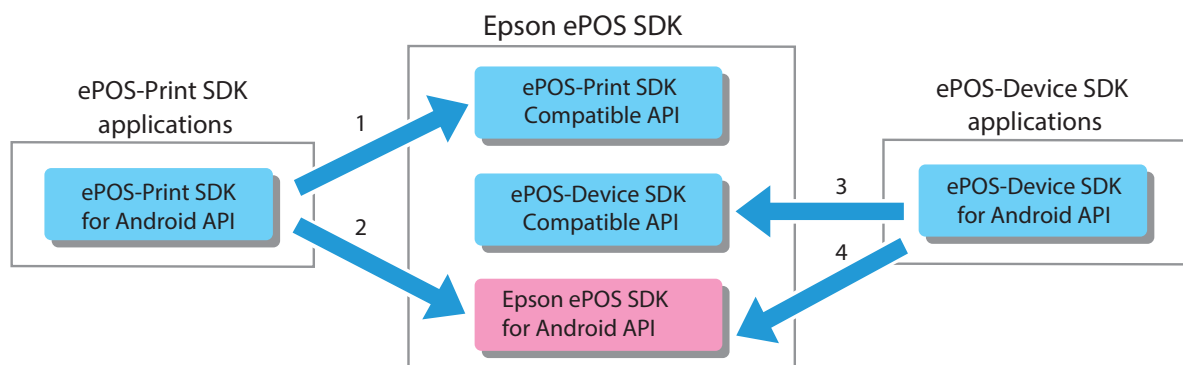
- ❑ Migration with compatible APIs for Epson ePOS SDK

Epson ePOS SDK includes both ePOS-Print SDK-compatible APIs and ePOS-Device SDK-compatible APIs. You can perform migration to an Epson ePOS SDK-compatible application by updating and building the configuration file without modifying the existing application program. If you want to use printing and other basic functions, you can also apply applications to new TM printers.

- ❑ Migration with Epson ePOS SDK APIs

You can perform migration of an existing application to a Epson ePOS SDK-compatible application by turning it into a program that uses Epson ePOS SDK APIs. Many program modifications are necessary, but applications can be applied to new models and functions of TM printers and peripheral devices.

Migration Types



- 1: Migration from ePOS-Print SDK with ePOS-Print SDK-compatible APIs of Epson ePOS SDK
- 2: Migration from ePOS-Print SDK with the APIs of Epson ePOS SDK
- 3: Migration from ePOS-Device SDK with ePOS-Device SDK-compatible APIs of Epson ePOS SDK
- 4: Migration from ePOS-Device SDK with the APIs of Epson ePOS SDK

Compatibility with new products and functions

New products and functions	ePOS-Print SDK ePOS-Device SDK	Epson ePOS SDK	
			ePOS-Print SDK-compatible API ePOS-Device SDK-compatible API
Epson TM printers and peripheral devices	-	✓	✓
New functions of Epson TM printers and peripheral devices	-	✓	- *
New Android version	-	✓	✓
New tablets and smartphone models	-	✓	✓

✓ : Compatible -: Incompatible

* Not compatible with the new functions that require new APIs or API modifications.

Migration from ePOS-Print SDK

This chapter explains the method to perform migration of the applications that use ePOS-Print SDK to the applications that are compatible with Epson ePOS SDK.

Migration with ePOS-Print SDK-compatible APIs

You can perform migration to an Epson ePOS SDK-compatible application by replacing the configuration file without modifying the existing application program.

Migration procedure

The migration procedure is described below.

Procedure		Description
1	SDK file replacement	Class file and library file replacement Refer to " SDK file replacement ".
2	Application building	Building a project for the application with SDK files being replaced

This completes the migration with the ePOS-Print SDK-compatible API.

SDK file replacement

Replace the following files that are included in the application project with Epson ePOS SDK files.

Type	ePOS-Print SDK	ePOS-Print SDK-compatible API
Class File *	ePOS-Print.jar	ePOS2.jar
Library	libeposprint.so	libepos2.so

* You do not need to replace the ePOSEasySelect.jar file with the file included in the Epson ePOS SDK package.

Application development with ePOS-Print SDK-compatible APIs

Refer to the following manuals for information that is necessary for the development and maintenance of the applications that use ePOS-Print SDK-compatible APIs.

- Specifications for ePOS-Print SDK-compatible APIs

"ePOS-Print SDK for Android User's Manual"

The specifications for ePOS-Print SDK-compatible APIs are the same as the specifications for the ePOS-Print SDK APIs.

- Device information and support APIs for the new Epson TM printer models

The [Appendix](#) of "Epson ePOS SDK for Android Migration Guide" (this document)

Migration with the Epson ePOS SDK APIs

You can perform migration to an Epson ePOS SDK-compatible application by modifying the existing application program. Many program modifications are necessary, but applications can be applied to new models and functions of TM printers.

Migration procedure

The migration procedure is described below.

Procedure		Description
1	SDK file replacement	Class file and library file replacement Refer to " SDK file replacement ".
2	Changing package names	Changing the package name of the ePOS-Print SDK to the package name of Epson ePOS SDK Refer to " Changing package names ".
3	Changing classes	Changing the ePOS-Print SDK class to the Epson ePOS SDK class Refer to " Changing classes ".
4	Changing APIs	<p>Modifying the program or changing the ePOS-Print SDK APIs that have different specifications from those of Epson ePOS SDK</p> <p>The changes are described below.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Modifying the program to enable specific functions Modify the functions listed below. <ul style="list-style-type: none"> • Establishing and cutting the communication with the printer Refer to "Establishing and cutting the communication with the printer". • Printing Refer to "Printing". • Obtaining status Refer to "Obtaining status". • Printer search Refer to "Printer search". • Status monitoring Refer to "Monitoring of the status". <input type="checkbox"/> Changing API name API names to be changed (You may need to change parameters too) Refer to "Changing API name". <input type="checkbox"/> Changing API parameters API names do not change, but parameter changes are necessary. Refer to "Changing API parameters". <input type="checkbox"/> Changing Listener interfaces Refer to "Changing Listener Interfaces".
5	Application building	Building a project for the modified application.

This completes the migration with the Epson ePOS SDK API.

SDK file replacement

Replace the following files that are included in the application project with Epson ePOS SDK files.

Type	ePOS-Print SDK	Epson ePOS SDK
Class File	ePOS-Print.jar	ePOS2.jar
	ePOSEasySelect.jar	ePOSEasySelect.jar *
Library	libeposprint.so	libepos2.so
	libeposeasyselect.so	libeposeasyselect.so *

* The file names do not change. Use the files included in the Epson ePOS SDK package.

Changing package names

Change the package name specified in the application project to the package name of Epson ePOS SDK.

Package names of ePOS-Print SDK to delete

Package Name to Delete	Description
com.epson.epsonio	This package is necessary for the migration with ePOS-Print SDK-compatible APIs. Since it is not necessary for the migration with the Epson ePOS SDK APIs, delete it.

Package names of Epson ePOS SDK to add

Package Name to Add	Description
com.epson.epos2	This package is shared by each class of Epson ePOS SDK.
com.epson.epos2.discovery	This package is used for printer search of Epson ePOS SDK.

Package names to change

Type	ePOS-Print SDK	Epson ePOS SDK
Print class	com.epson.eposprint	com.epson.epos2.printer
Builder class		

Changing classes

When performing migration from ePOS-Print SDK to Epson ePOS SDK, change the class as follows.

- Change the classes of ePOS-Print SDK used in the application project to the classes of Epson ePOS SDK.

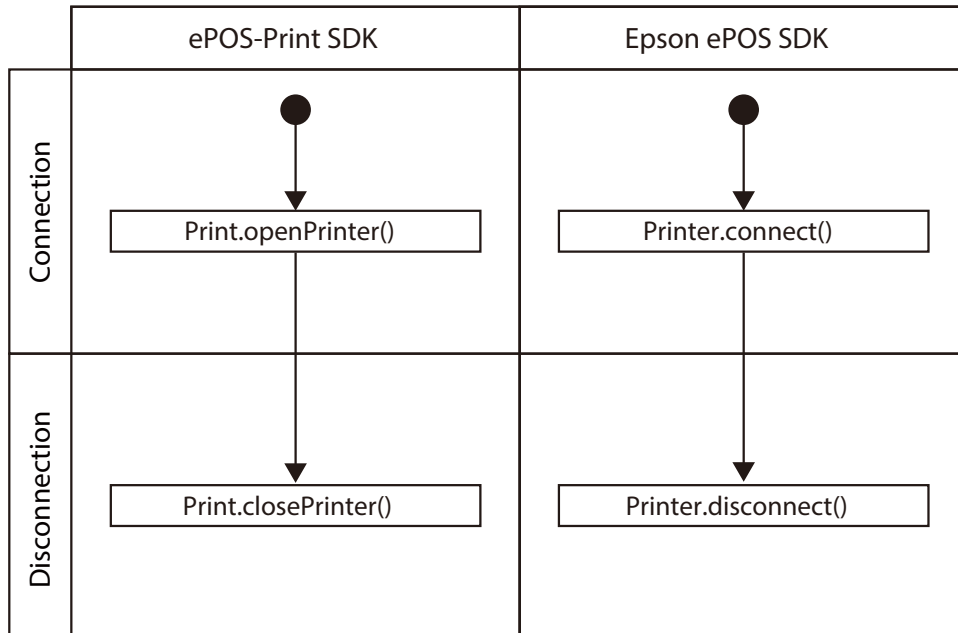
Type	ePOS-Print SDK	Epson ePOS SDK
Print function	Builder class	Printer class
	Print class	
Printer search	Finder class	Discovery class
Exception	EposException class	Epos2Exception class
	EpsonIoException class	

- Change a part of the definition value inside the Print class used in the application project to EasySelectDeviceType class.

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

Establishing and cutting the communication with the printer

Execution procedure differences



Program differences

❑ 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);
    ...processing...

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

```

❑ Epson ePOS SDK

```

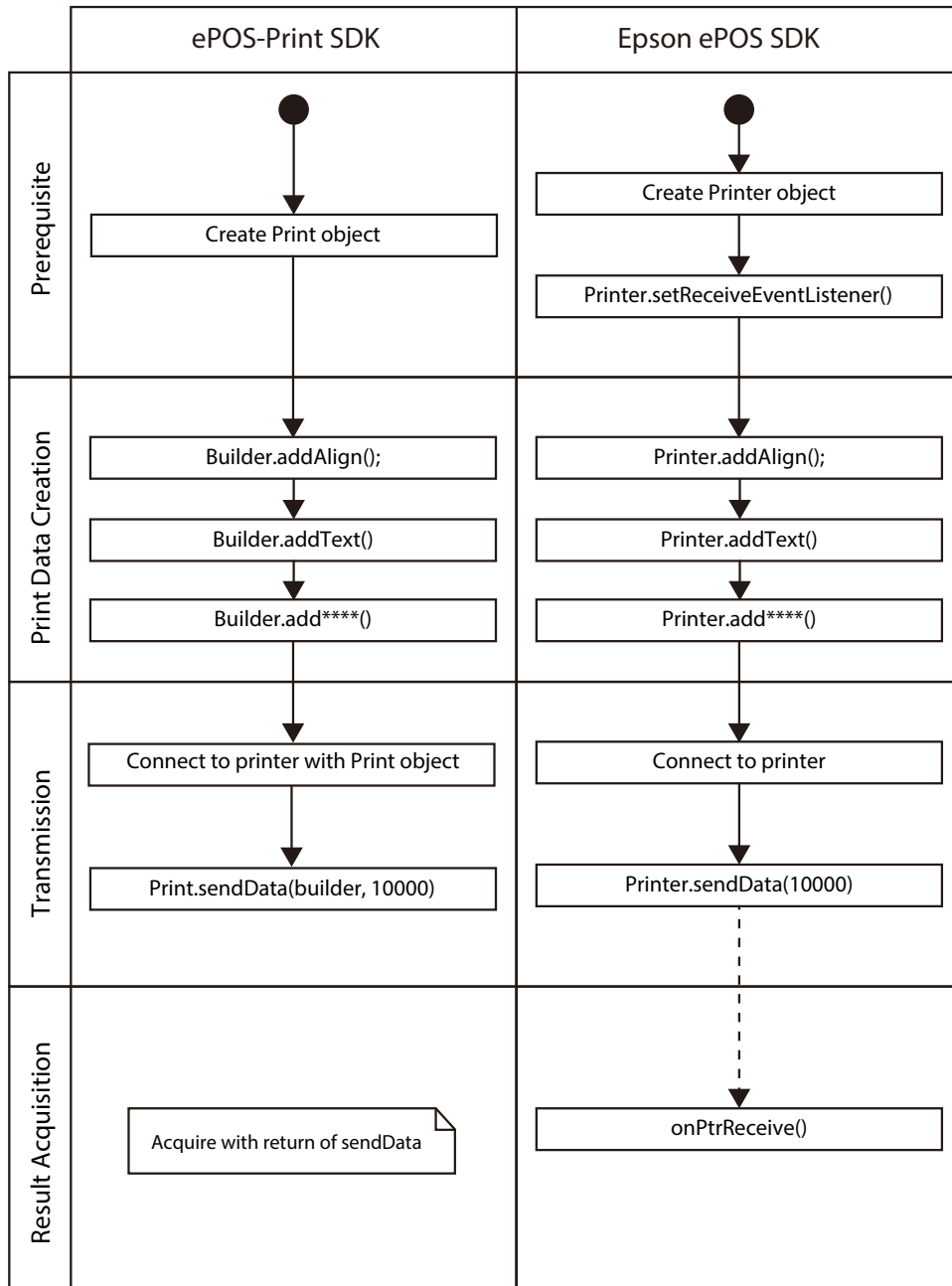
Printer printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK, getApplicationContext());
try {
    printer.connect("TCP:192.168.192.168", Printer.PARAM_DEFAULT);
    ...processing...
    printer.disconnect();
} catch (Epos2Exception e) {
    int errStatus = e.getErrorStatus();
}

```

Printing

In ePOS-Print SDK, the printing results were obtained with the return value of the printing data transmission processing. whereas in Epson ePOS SDK, the printing results are obtained through callback.

Execution procedure differences



Callback: - - - - ►

Program differences

❑ 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;

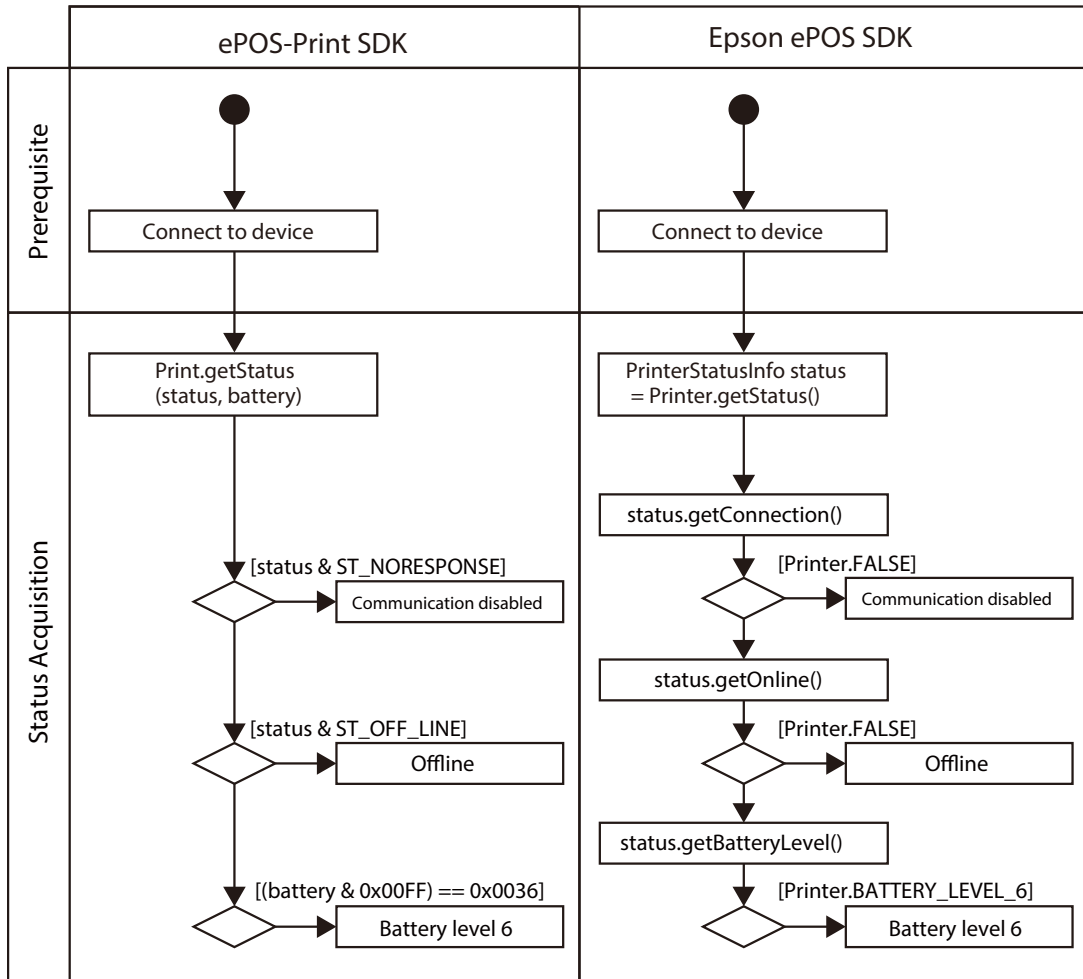
    private void openPrinter() {
        int errorStatus;
        try {
            printer = new Printer(Printer.TM_T88, Printer.MODEL_ANK,
                getApplicationContext());
            printer.setReceiveEventListener(this);
            ...connection...
            printer.addText("ABCDE");
   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

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

Obtaining status

In ePOS-Print SDK, the combinations of multiple printer statuses were obtained with the return value, whereas in Epson ePOS SDK, each status is obtained from PrinterStatusInfo-type properties.

Execution procedure differences



Program differences

❑ 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

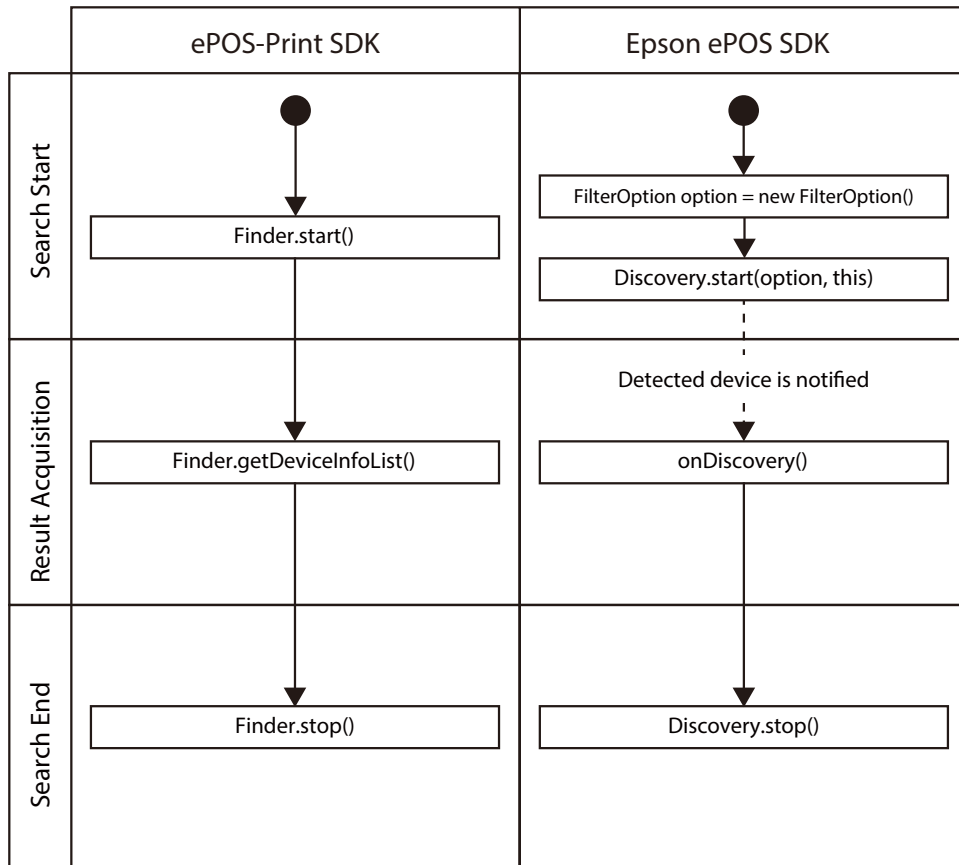
```
int errStatus;
Printer printer = NULL;
try {
    ...connection...
    PrinterStatusInfo status = printer.getStatus();

    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();
}
```


Printer search

In ePOS-Print SDK, the printer search results were obtained through the API, whereas in Epson ePOS SDK the search results are obtained with the callback method after the filtering settings have been configured.

Execution procedure differences



Callback: - - - ->

Program differences

❑ 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);

            ...searching...

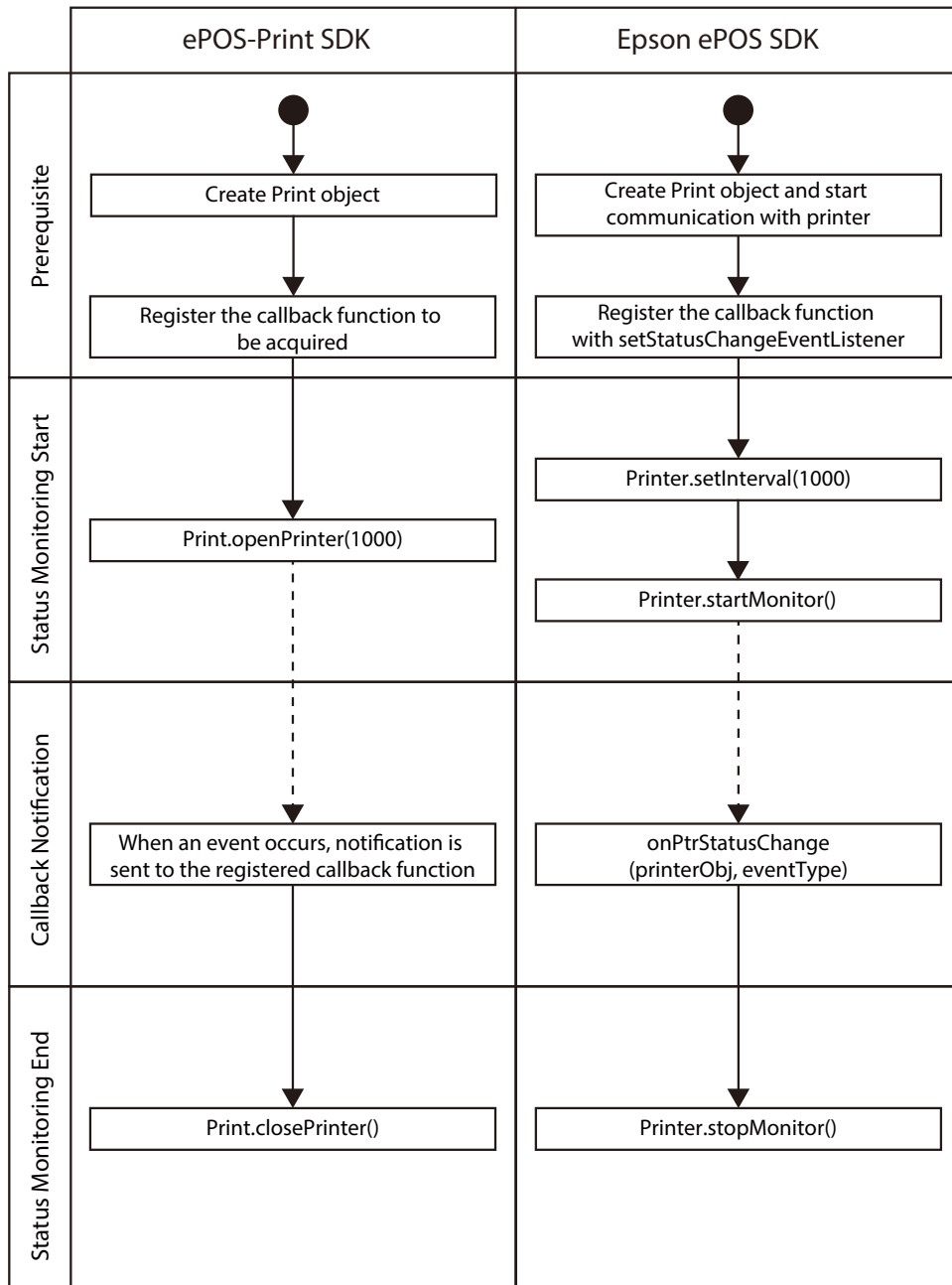
            Discovery.stop();
        } catch (Epos2Exception e) {
            int errStatus = e.getErrorStatus();
        }
    }

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

Monitoring of the status

In ePOS-Print SDK, the communication with the printer and status monitoring started at the same time, whereas in Epson ePOS SDK, status monitoring starts after the communication with the printer has been established.

Execution procedure differences



Callback: ----▶

Program differences

❑ ePOS-Print SDK

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

            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);
        ...connection...
        printer.startMonitor();

        printer.stopMonitor();
    }

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

Changing API name

During the migration from ePOS-Print SDK to Epson ePOS SDK, the APIs that need to be renamed are listed in the table below. In some cases, multiple APIs are bundled in one API, or one API is divided into several APIs. Among the APIs in the table below, there are APIs where specifications other than the name have changed. To see what has changed, compare APIs in the "ePOS-Print SDK for Android User's Manual" and "Epson ePOS SDK for Android User's Manual".

A list of API names to be changed

Function	ePOS-Print SDK	Epson ePOS SDK
Class initialization	Print	Printer
	Builder	
Adding the text line space settings to the instruction buffer	addTextLineSpace	addLineSpace
Adding the character size settings to the instruction buffer	addTextDouble	addTextSize
Adding the text printing position settings to the instruction buffer	addTextPosition	addHPosition
Starting communication	openPrinter	connect
Starting status monitoring		startMonitor
Terminating communication	closePrinter	disconnect
Terminating status monitoring		stopMonitor
Registering the notification destination of printer statuses	setStatusChangeEventCallback	setStatusChangeListener
Registering the notification destination of online events	setOnlineEventCallback	
Registering the notification destination of offline events	setOfflineEventCallback	
Registering the notification destination of power-off events	setPowerOffEventCallback	
Registering the notification destination of cover-closed events	setCoverOkEventCallback	
Registering the notification destination of cover-open events	setCoverOpenEventCallback	
Registering the notification destination of paper OK events	setPaperOkEventCallback	
Registering the notification destination of paper near-end events	setPaperNearEndEventCallback	
Registering the notification destination of paper end events	setPaperEndEventCallback	
Registering the notification destination of drawer-closed events	setDrawerClosedEventCallback	
Registering the notification destination of drawer-open events	setDrawerOpenEventCallback	
Registering the notification destination of battery-low events	setBatteryLowEventCallback	
Registering the notification destination of battery level OK events	setBatteryOkEventCallback	

Function	ePOS-Print SDK	Epson ePOS SDK
Registering the notification destination of battery statuses	setBatteryStatusChangeEventCall-back	setStatusChangeListener
Obtaining printer search results	start	start
	getDeviceInfoList	

Changing API parameters

The APIs where the parameters need to be changed when migrating from ePOS-Print SDK to Epson ePOS SDK are listed in the table below. To see what has changed, compare APIs in the "ePOS-Print SDK for Android User's Manual" and "Epson ePOS SDK for Android User's Manual".

A list of APIs with parameters to be changed

API	Parameter change
addTextAlign	Change from Builder class to Printer class
addTextRotate	Change from Builder class to Printer class
addTextLang	Change from Builder class to Printer class
addTextFont	Change from Builder class to Printer class
addTextSmooth	Change from Builder class to Printer class
addTextSize	Change from Builder class to Printer class
addTextStyle	Change from Builder class to Printer class
addImage	Change from Builder class to Printer class compress setting added
addBarcode	Change from Builder class to Printer class
addSymbol	Change from Builder class to Printer class
addPageDirection	Change from Builder class to Printer class
addPagePosition	Change from Builder class to Printer class
addPageLine	Change from Builder class to Printer class
addPageRectangle	Change from Builder class to Printer class
addCut	Change from Builder class to Printer class
addPulse	Change from Builder class to Printer class
addSound	Change from Builder class to Printer class
addLayout	Change from Builder class to Printer class
sendData	Change to timeout only
getStatus	No parameter, and value obtained with the return value
parseNFC	timeout parameter added
createQR	The deviceType setting name changed

Changing Listener Interfaces

The listener interfaces that need to be changed or deleted when migrating from ePOS-Print SDK to Epson ePOS SDK are listed in the tables below.

To see what changes, compare APIs where the notification destination method of the listener is registered in the "ePOS-Print SDK for Android User's Manual" and "Epson ePOS SDK for Android User's Manual".

Listener interfaces to be changed

Function	ePOS-Print SDK	Epson ePOS SDK
	Event Listener	
	Notification Method	
Notification of printer status	public interface StatusChangeEventListener extends EventListener	public interface StatusChangeListener extends EventListener
	void onStatusChangeEvent(String deviceName, int status)	void onPtrStatusChange(Printer printerObj, int eventType)

ePOS-Print SDK listener interfaces to be deleted

Function	Event Listener
	Notification Method
Notification of online event	public interface OnlineEventListener extends EventListener
	void onOnlineEvent(String deviceName)
Notification of offline event	public interface OfflineEventListener extends EventListener
	void onOfflineEvent(String deviceName)
Notification of power-off event	public interface PowerOffEventListener extends EventListener
	void onPowerOffEvent(String deviceName)
Notification of cover closed	public interface CoverOkEventListener extends EventListener
	void onCoverOkEvent(String deviceName)
Notification of cover open	public interface CoverOpenEventListener extends EventListener
	void onCoverOpenEvent(String deviceName)
Notification of paper OK	public interface PaperOkEventListener extends EventListener
	void onPaperOkEvent(String deviceName)
Notification of paper near-end	public interface PaperNearEndEventListener extends EventListener
	void onPaperNearEndEvent(String deviceName)
Notification of paper end	public interface PaperEndEventListener extends EventListener
	void onPaperEndEvent(String deviceName)
Notification of drawer closed	public interface DrawerClosedEventListener extends EventListener
	void onDrawerClosedEvent(String deviceName)
Notification of drawer open	public interface DrawerOpenEventListener extends EventListener
	void onDrawerOpenEvent(String deviceName)
Notification of battery low	public interface BatteryLowEventListener extends EventListener
	void onBatteryLowEvent(String deviceName)
Notification of battery level OK	public interface BatteryOkEventListener extends EventListener
	void onBatteryOkEvent(String deviceName)
Notification of battery status	public interface BatteryStatusChangeEventListener extends EventListener
	void onBatteryStatusChangeEvent(String deviceName, int battery)

Migration from ePOS-Device SDK

This chapter explains the method to perform migration of the applications that use ePOS-Device SDK to the applications that are compatible with Epson ePOS SDK.

Migration with ePOS-Device SDK-compatible APIs

You can perform migration to an Epson ePOS SDK-compatible application by replacing the configuration file without modifying the existing application program.

Migration procedure

The migration procedure is described below.

Procedure		Description
1	SDK file replacement	Class file and library file replacement Refer to " SDK file replacement ".
2	Application building	Building a project for the application with SDK files being replaced

This completes the migration with ePOS-Device SDK-compatible APIs.

SDK file replacement

Replace the following files that are included in the application project with Epson ePOS SDK files.

Type	ePOS-Device SDK	ePOS-Device SDK-compatible API
Class File	ePOS-Device.jar	ePOS2.jar
Library	libeposdevice.so	libepos2.so

Application development with ePOS-Device SDK-compatible APIs

Refer to the following manuals for information that is necessary for the development and maintenance of the applications that use ePOS-Device SDK-compatible APIs.

Specifications for ePOS-Device SDK-compatible APIs: "ePOS-Device SDK for Android User's Manual"

Migration with the Epson ePOS SDK APIs

You can perform migration to an Epson ePOS SDK-compatible application by modifying the existing application program. Many program modifications are necessary, but applications can be applied to new models and functions of TM printers or peripheral devices.

Migration procedure

The migration procedure is described below.

Procedure		Description
1	SDK file replacement	Class file and library file replacement Refer to " SDK file replacement ".
2	Changing package names	Changing the package name of the ePOS-Device SDK to the package name of Epson ePOS SDK Refer to " Changing package names ".
3	Changing classes	Changing the ePOS-Device SDK classes to the Epson ePOS SDK classes Refer to " Changing classes ".
4	Changing APIs	<p>Modifying the program or changing the ePOS-Device SDK APIs that have different specifications from those of Epson ePOS SDK</p> <p>The changes are described below.</p> <ul style="list-style-type: none"> <input type="checkbox"/> Modifying the program to enable specific functions Modify the functions listed below <ul style="list-style-type: none"> • Establishing and cutting the communication with the device Refer to "Establishing and cutting the communication with the device". • Reconnection notifications Refer to "Reconnection notifications". • Printing Refer to "Printing". • Forced sending Refer to "Forced sending". • Obtaining status Refer to "Obtaining status". <input type="checkbox"/> Changing API names API names to be changed (You may need to change parameters too) Refer to "Changing API names". <input type="checkbox"/> Changing API parameters API names do not change, but parameter changes are necessary. Refer to "Changing API parameters". <input type="checkbox"/> Changing Listener interfaces Refer to "Changing Listener interfaces".
5	Application building	Building a project for the modified application.

This completes the migration with the Epson ePOS SDK APIs.

SDK file replacement

Replace the following files that are included in the application project with Epson ePOS SDK files.

Type	ePOS-Device SDK	Epson ePOS SDK
Class File	ePOS-Device.jar	ePOS2.jar
Library	libeposdevice.so	libepos2.so

Changing package names

Change the package name set in the application project to the package name of Epson ePOS SDK.

Package names to be changed

Type	ePOS-Device SDK	Epson ePOS SDK
Package shared by each class	com.epson.eposdevice	com.epson.epos2
Display class	com.epson.eposdevice.display	com.epson.epos2.linedisplay
Keyboard class	com.epson.eposdevice.keyboard	com.epson.epos2.keyboard
Printer class	com.epson.eposdevice.printer	com.epson.epos2.printer
Scanner class	com.epson.eposdevice.scanner	com.epson.epos2.barcodescanner
SimpleSerial class	com.epson.eposdevice.simpleserial	com.epson.epos2.simpleserial
CommBoxManager class	com.epson.eposdevice.commbox	com.epson.epos2.commbox
CommBox class		

Changing classes

Change the classes of ePOS-Device SDK used in the application project to the classes of Epson ePOS SDK.

ePOS-Device SDK class names to be deleted

Class name	Details
Device class	<p>The following functions of the Device class are assigned to the APIs of each class.</p> <ul style="list-style-type: none"> • Establishing the communication path • Disconnecting the communication path • Obtaining the establishment status of the communication path • Obtaining administrator information • Obtaining installation location information • Registering the callback method for the reconnection processing start event • Registering the callback method for the reconnection end event • Registering the callback method for the network disconnection event
CommBoxManager class	This is integrated in the CommBox class

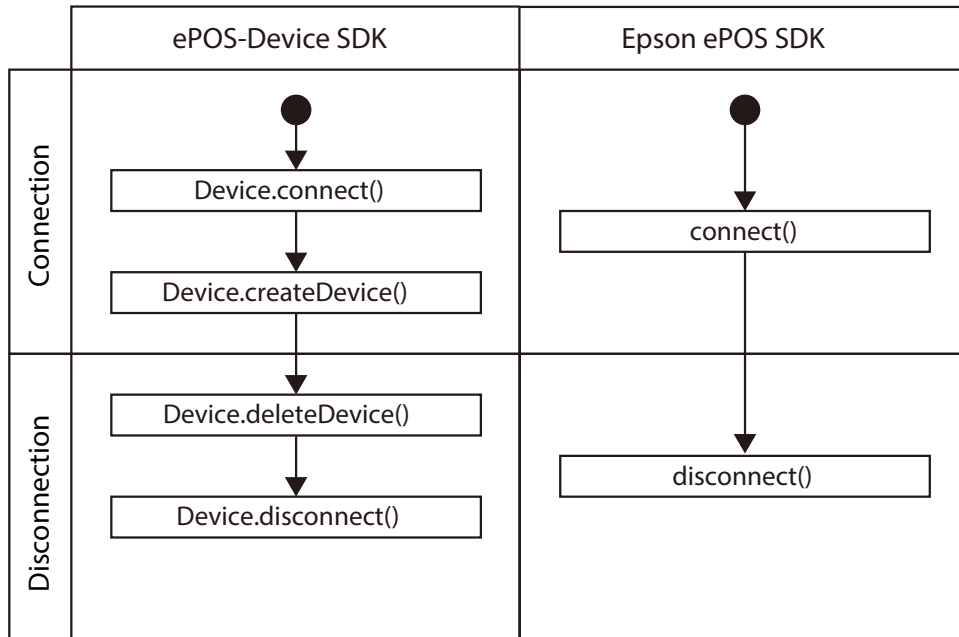
Class names to be changed

Type	ePOS-Device SDK	Epson ePOS SDK
Device control	Display class	LineDisplay class
	Scanner class	BarcodeScanner class
Exception	EposException class	Epos2Exception class

Establishing and cutting the communication with the device

In ePOS-Device SDK, the communication with each device used to start after the connection with ePOS-Device Service had been established, whereas in Epson ePOS SDK the communication starts for each individual device. Also, in ePOS-Device SDK, the connection with ePOS-Device Service used to terminate after the communication with the device had ended, whereas in Epson ePOS SDK the communication terminates for each individual device.

Execution procedure differences



Program differences

□ 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();
            }
        }
    }
}

```

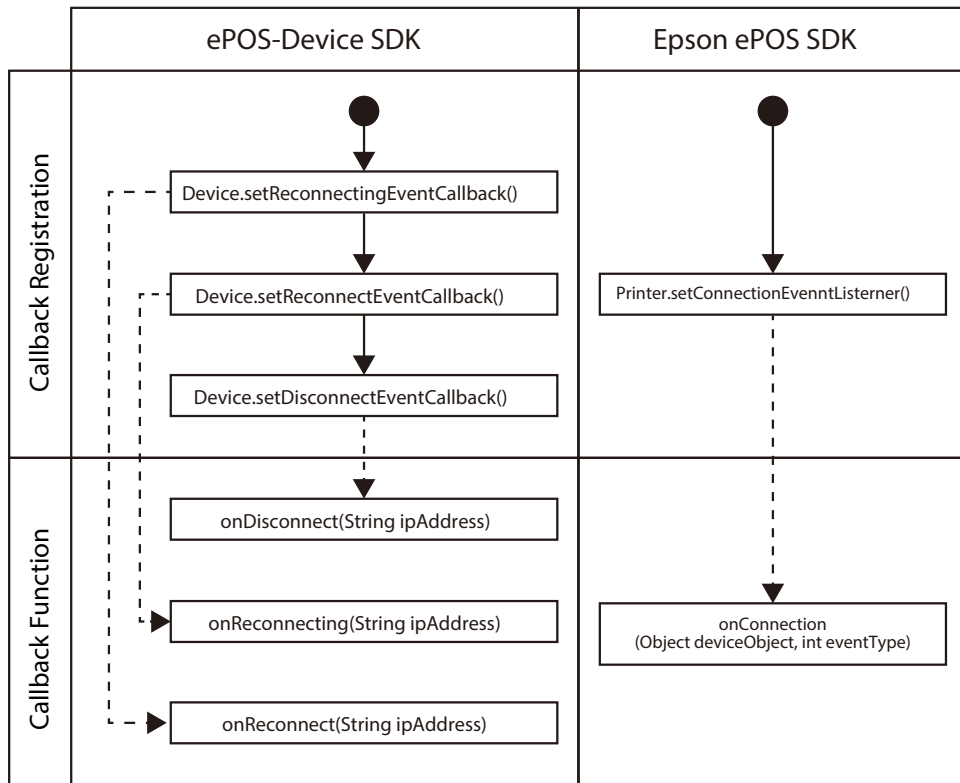
□ Epson ePOS SDK

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

Reconnection notifications

In ePOS-Device SDK, the registration was carried out through the API for each notification type, whereas in Epson ePOS SDK the APIs for the registration of notification destination are merged in one API and the processing is carried out for each notification type using the notification destination method.

Execution procedure differences



Callback: - - - ->

Program differences

- ❑ 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);
            ...connection...
        } catch (EposException e) {
            errorStatus = e.getErrorStatus();
        }
    }

    @Override
    public void onReconnecting(String ipAddress) {
        ...starting reconnection...
    }
    @Override
    public void onReconnect(String ipAddress) {
        ...reconnection end...
    }
    @Override
    public void onDisconnect(String ipAddress) {
        ...reconnection failed...
    }
}
```

□ 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,
                                getContext());
            printer.setConnectionEventListener(this);
            ...connection...
        } catch (Epos2Exception e) {
            errorStatus = e.getErrorStatus();
        }
    }
    @Override
    public void onConnection(Object deviceObj, int eventType) {
        if(eventType == Printer.EVENT_RECONNECTING) {
            ...starting reconnection...
        }
        if(eventType == Printer.EVENT_RECONNECT) {
            ...reconnection end...
        }
        if(eventType == Printer.EVENT_DISCONNECT) {
            ...reconnection failed...
        }
    }
}
```

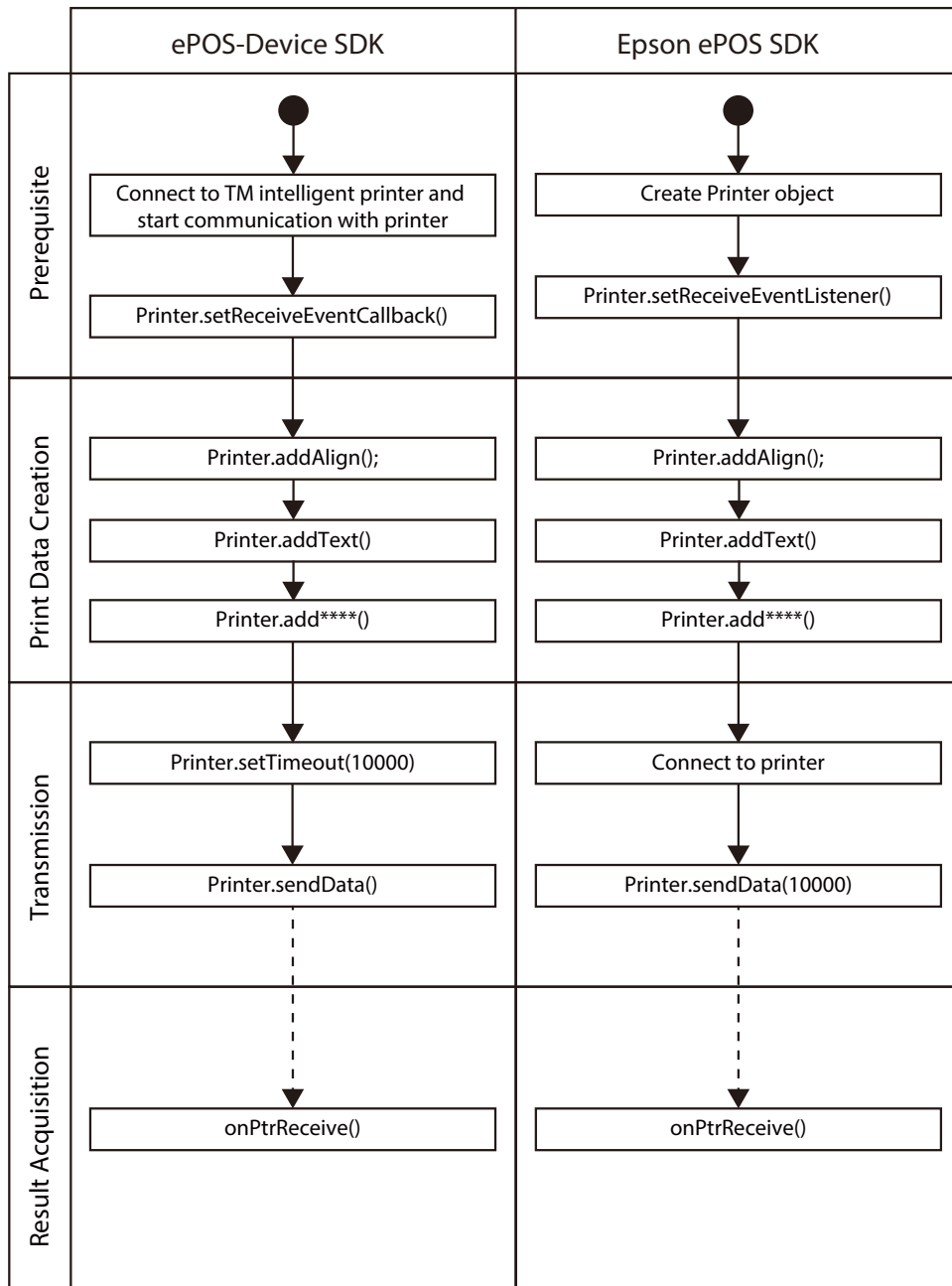
Printing

In ePOS-Device SDK, the printing data was created after the connection with the printer had been established, whereas in Epson ePOS SDK, the data can be created either before or after the connection with the printer is established.

Printing is possible without any modifications to the execution procedures of the existing programs.

Execution procedure differences

The execution procedure of Epson ePOS SDK is the procedure for creating printing data before establishing a connection with the printer.



Callback: - - - - ►

Program differences

❑ ePOS-Device SDK

```

public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

    private void openPrinter() {
        ...connection...

        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) {
        ...processing...
    }
}

```

❑ Epson ePOS SDK

The following program creates printing data before establishing a connection with the printer.

```

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");
            ...connection...
            printer.sendData(Printer.PARAM_DEFAULT);
        } catch (Epos2Exception e){
            errorStatus = e.getErrorStatus();
        }
    }

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

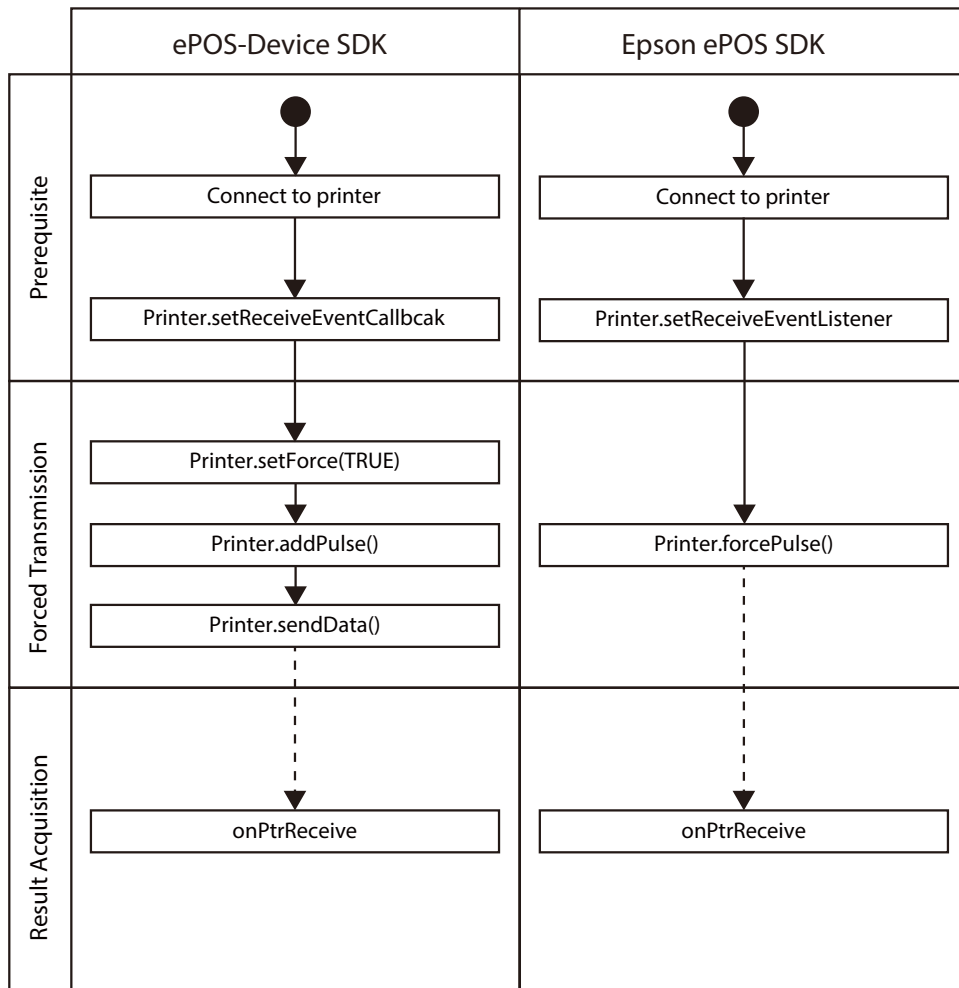
```

Forced sending

In ePOS-Device SDK, forced sending was carried out with 3 APIs, whereas in Epson ePOS SDK it is carried out with 1 API.

Also, in ePOS-Device SDK, forced sending was only enabled in offline mode, whereas in Epson ePOS SDK it can be carried out either in online or offline mode.

Execution procedure differences



Callback: - - - ->

Program differences

❑ ePOS-Device SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

    private void openPrinter() {
        ...connection...

        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) {
        ...processing...
    }
}
```

❑ Epson ePOS SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

    private Printer printer = NULL;

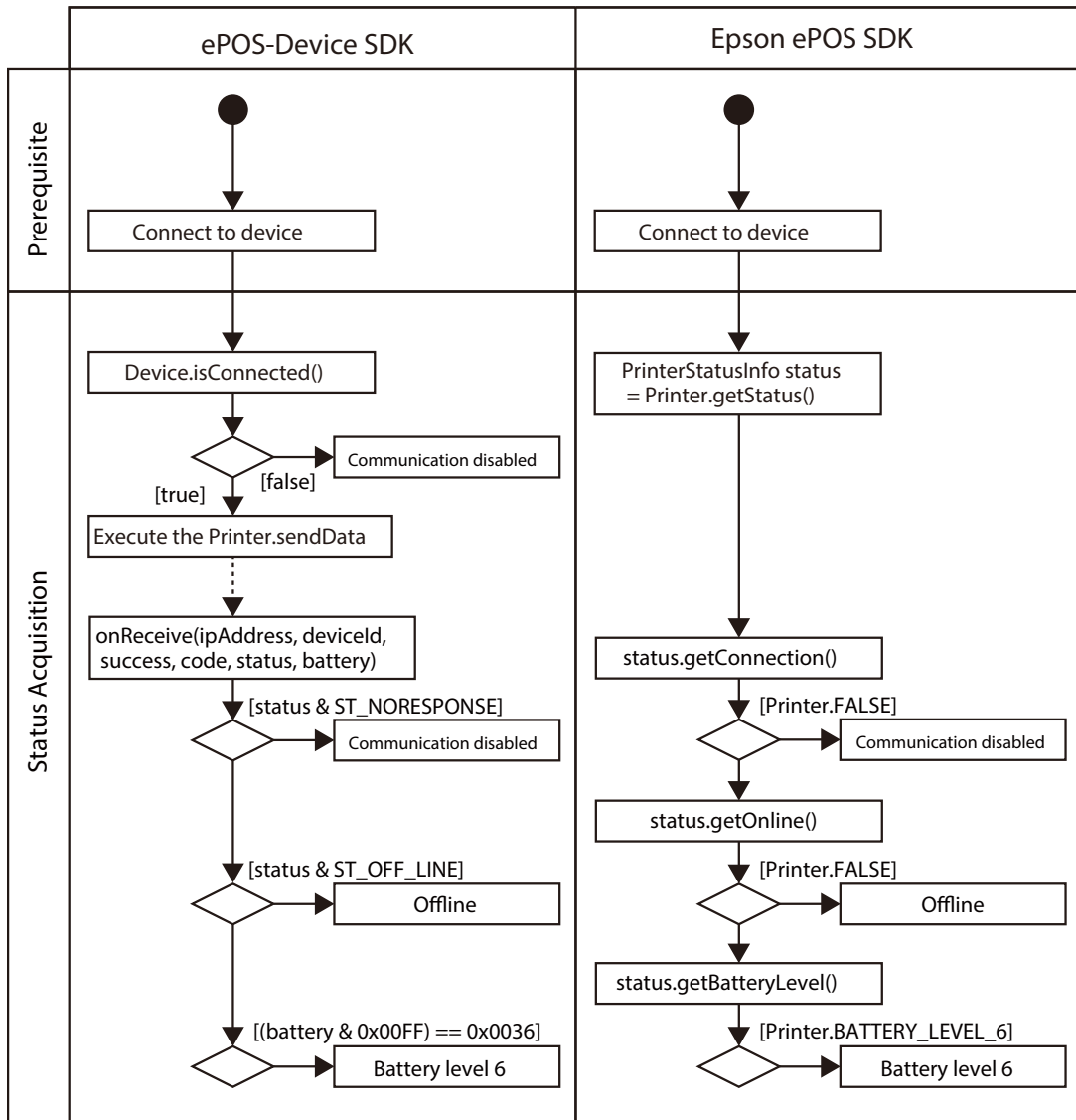
    private void openPrinter() {
        int errorStatus;
        try {
            ...connection...
            printer.setReceiveEventListener(this);
            printer.forcePulse(Printer.DRAWER_2PIN, Printer.PULSE_100,
Printer.PARAM_DEFAULT);
        } catch (Epos2Exception e){
            errorStatus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

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

Obtaining status

In ePOS-Device SDK, the combinations of multiple printer statuses were obtained with the return value, whereas in Epson ePOS SDK, each status is obtained from PrinterStatusInfo-type properties.

Execution procedure differences



Callback: ---->

Program differences

❑ ePOS-Device SDK

```
public class SampleActivity extends Activity implements ReceiveListener {

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

    private void getStatus() {
        ...connection...
        if(device.isConnected() != true) {
            ...disconnecting...
        }

        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 {
    ...connection...
    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();
}
```


Changing API names

During the migration from ePOS-Device SDK to Epson ePOS SDK, the APIs that need to be renamed are listed in the table below. In some cases, multiple APIs are bundled in one API, one API is divided into several APIs, or some APIs have been deleted. Among the APIs in the table below, there are APIs where specifications other than the name have changed. To see what has changed, compare APIs in the "ePOS-Device SDK for Android User's Manual" and "Epson ePOS SDK for Android User's Manual".

A list of API names to be changed

Class	Function	ePOS-Device SDK	Epson ePOS SDK
Class - common			
Class initialization		init	Printer
			LineDisplay
			Keyboard
			BarcodeScanner
			SimpleSerial
			CommBox
Obtaining the device object		connect	connect
		createDevice	
Discarding the device object		disconnect	disconnect
		deleteDevice	
Obtaining the present status information		isConnected	getStatus
Registering the notification destination of reconnection processing start event		setReconnectingEventCallback	setConnectionEventListener
Registering the notification destination of reconnection end event		setReconnectEventCallback	
Registering the notification destination of network disconnection event		setDisconnectEventCallback	
Display class			
Adding display area definition to the instruction buffer		createWindow	addCreateWindow
Adding display area settings cancellation to the instruction buffer		destroyWindow	addDestroyWindow
Adding display area switching to the instruction buffer		setCurrentWindow	addSetCurrentWindow
Adding display area settings deletion to the instruction buffer.		clearWindow	addClearCurrentWindow
Adding the cursor position to the instruction buffer		setCursorPosition	addSetCursorPosition
Adding the cursor position in the display area to the instruction buffer		moveCursorPosition	addMoveCursorPosition
Adding cursor type change to the instruction buffer		setCursorType	addSetCursorType
Adding the marquee view to the instruction buffer		addMarquee	addMarqueeText
Adding blinking display information to the instruction buffer		setBlink	addSetBlink

Class	Function	ePOS-Device SDK	Epson ePOS SDK
		Display class	
	Adding display brightness information to the instruction buffer	setBrightness	addSetBrightness
	Adding time displayed on the clock to the instruction buffer	ShowClock	addShowClock
	Resetting the customer display	reset	addInitialze
	Registering the notification destination of control result reception events	SetReceiveEventCallback	setReceiveEventListener
Keyboard class			
	Registering the notification destination of pressed key detection events	setKeyPressEventCallback	setKeyPressEventListener
	Registering the notification destination of character string detection events	setStringEventCallback	setReadStringEventListener
Printer class			
	Adding the text line space settings to the instruction buffer	addTextLineSpace	addLineSpace
	Adding the character size settings to the instruction buffer	addTextSize	addTextSize
		addTextDouble	
	Adding line feeding to the instruction buffer	addFeedLine	addFeedLine
		addFeed	
	Adding the text printing position settings to the instruction buffer	addTextPosition	addHPosition
	Acquire the current status	setOnlineEventCallback	getStatus
	Specify the halftone processing method for raster image	halftone property	addImage
	Specify the brightness correction value for raster image	brightness property	
	Reset the printer	reset	forceReset
	Forced sending	force property	forceRecover
			forcePulse
			forceStopSound
			forceCommand
	Registering the notification destination of printer statuses	setStatusChangeEventCallback	setStatusChangeListener
	Registering the notification destination of online events	setOnlineEventCallback	
	Registering the notification destination of offline events	setOfflineEventCallback	
	Registering the notification destination of power-off events	setPowerOffEventCallback	
	Registering the notification destination of cover-closed events	setCoverOkEventCallback	
	Registering the notification destination of cover-open events	setCoverOpenEventCallback	
	Registering the notification destination of paper presence events	setPaperOkEventCallback	

Class		ePOS-Device SDK	Epson ePOS SDK
	Function		
Printer class			
	Registering the notification destination of paper near-end events	setPaperNearEndEventCallback	setStatusChangeListener
	Registering the notification destination of paper end events	setPaperEndEventCallback	
	Registering the notification destination of drawer-closed events	setDrawerClosedEventCallback	
	Registering the notification destination of drawer-open events	setDrawerOpenEventCallback	
	Registering the notification destination of battery-low events	setBatteryLowEventCallback	
	Registering the notification destination of battery level OK events	setBatteryOkEventCallback	
	Registering the notification destination of battery statuses	setBatteryStatusChangeEvent-Callback	
	Registering the notification destination of response document reception events	setReceiveEventCallback	setReceiveEventListener
Scanner class			
	Registering the notification destination of barcode data input events	setDataEventCallback	setScanEventListener
SimpleSerial class			
	Registering the notification destination of reception events from the device	setCommandReplyEventCallback	setReceiveEventListener
CommBox class			
	Acquiring CommBox object	getCommBoxManager	connect
	Creating a communication box	openCommBox	
	Discarding the communication box	closeCommBox	disconnect
	Sending a message to the communication box	sendData	sendMessage
	Registering the notification destination for receiving a message in the communication box	setReceiveEventCallback	setReceiveEventListener

Changing API parameters

The APIs where the parameters need to be changed when migrating from ePOS-Device SDK to Epson ePOS SDK are listed in the table below. To see what has changed, compare APIs in the "ePOS-Device SDK for Android User's Manual" and "Epson ePOS SDK for Android User's Manual".

A list of APIs with parameters to be changed

Class	API	Parameter changes
Display class		
	addText	lang settings added
	addReverseText	lang settings added
	setReceiveEventListener	code value (for callback method) added
Printer class		
	sendData	change to timeout only
	addTextAlign	align settings added
	addTextRotate	rotate settings added
	addTextLang	lang settings added
	addTextFont	font settings added
	addTextSmooth	smooth settings added
	addTextSize	width/height settings added
	addTextStyle	reverse/ ul/ em/ color settings added
	addImage	halftone/ brightness parameters added compress settings added
	addBarcode	hri/ font/ width/ height settings added
	addSymbol	level/ width/ height/ size settings added
	addPageDirection	direction settings added
	addPagePosition	x/ y settings added
	setReceiveEventListener	Callback method status/ battery parameters merged code value (for callback method) deleted
	interval properties	Settings added
CommBox class		
	getCommHistory	code value (for callback method) deleted

Changing Listener interfaces

The listener interfaces that need to be changed or deleted when migrating from ePOS-Device SDK to Epson ePOS SDK are listed in the tables below.

To see what has changed, compare APIs in the "ePOS-Device SDK for Android User's Manual" and "Epson ePOS SDK for Android User's Manual".

Listener interfaces to be changed

Class	ePOS-Device SDK	Epson ePOS SDK
Function	Event Listener	
	Notification Method	
Class - common		
Notification of reconnection processing start event	public interface ReconnectingListener extends EventListener	public interface ConnectionListener extends EventListener
	onReconnecting	void onConnection(Object deviceObj, int eventType)
Notification of reconnection end event	public interface ReconnectListener extends EventListener	Integrated into Listener interface of "Notification of reconnection processing start event".
	onReconnect	
Notification of network disconnection event	public interface DisconnectListener extends EventListener	
	onDisconnect	
Display class		
Notification of control result reception event	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)
Keyboard class		
Notification of key press detection event	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)
Notification of string detection event	public interface StringListener extends EventListener	public interface ReadStringListener extends EventListener
	void onKbdString(String ipAddress, String deviceId, String input, int prefix)	void onKbdReadString(Keyboard keyboardObj, String readString, int prefix)
Printer class		
Notification of printer status	public interface StatusChangeEventListener extends EventListener	public interface StatusChangeListener extends EventListener
	void onStatusChangeEvent(String deviceName, int status)	void onPtrStatusChange(Printer printerObj, int eventType)
Notification of response document reception event	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)

Class	ePOS-Device SDK	Epson ePOS SDK
Function	Event Listener	
	Notification Method	
Scanner class		
Notification of barcode data read event	public interface DataListener extends EventListener	public interface ScanListener extends EventListener
	void onScanData(String ipAddress, String deviceId, String input)	void onScanData (BarcodeScanner scannerObj, String scanData)
SimpleSerial class		
Notification of reception from device event	public interface CommandReplyListener extends EventListener	public interface ReceiveListener extends EventListener
	void onSimpleSerialCommandReply (String ipAddress, String deviceId, byte[] data)	void onSimpleSerialReceive (SimpleSerial serialObj, byte[] data)
CommBox class		
Notification of message reception event in Communication Box	public interface ReceiveListener extends EventListener	public interface ReceiveListener extends EventListener
	void onCommBoxReceive(String ipAddress, String senderId,?String receiverId, String message)	void onCommBoxReceive(CommBox commBoxObj, String senderId, String receiverId, String message)

ePOS-Device SDK listener interfaces to be deleted

Class	ePOS-Device SDK	Epson ePOS SDK
Function	Event Listener	
	Notification Method	
Printer class		
Notification of online event	public interface OnlineEventListener extends EventListener	
	void onOnlineEvent(String deviceName)	
Notification of offline event	public interface OfflineEventListener extends EventListener	
	void onOfflineEvent(String deviceName)	
Notification of power-off event	public interface PowerOffEventListener extends EventListener	
	void onPowerOffEvent(String deviceName)	
Notification of cover close	public interface CoverOkEventListener extends EventListener	
	void onCoverOkEvent(String deviceName)	
Notification of cover open	public interface CoverOpenEventListener extends EventListener	
	void onCoverOpenEvent(String deviceName)	
Notification of paper presence	public interface PaperOkEventListener extends EventListener	
	void onPaperOkEvent(String deviceName)	
Notification of paper near-end	public interface PaperNearEndEventListener extends EventListener	
	void onPaperNearEndEvent(String deviceName)	
Notification of paper end	public interface PaperEndEventListener extends EventListener	
	void onPaperEndEvent(String deviceName)	
Notification of drawer closed	public interface DrawerClosedEventListener extends EventListener	
	void onDrawerClosedEvent(String deviceName)	
Notification of drawer open	public interface DrawerOpenEventListener extends EventListener	
	void onDrawerOpenEvent(String deviceName)	

Class	
Function	Event Listener
	Notification Method
Printer class	
Notification of battery low	public interface BatteryLowEventListener extends EventListener
	void onBatteryLowEvent(String deviceName)
Notification of battery level OK	public interface BatteryOkEventListener extends EventListener
	void onBatteryOkEvent(String deviceName)
Notification of battery status	public interface BatteryStatusChangeListener extends EventListener
	void onBatteryStatusChangeEvent(String deviceName, int battery)
Notification of response document reception event	public interface JobReceiveListener extends ReceiveListener
	void onPtrReceive (String ipAddress, String deviceId, int success, int code, int status, int battery, String printJobId)

Appendix

ePOS-Print SDK-compatible API

A list of support APIs for each printer model

The following table lists the support APIs available for each model of printer.

The symbols used in the table represent the following:

- ✓: Supported
- -: Not supported.

API	TM-m10	TM-m30	TM-T60	TM-T88VI	TM-T88VI-iHUB
addTextAlign	✓	✓	✓	✓	✓
addTextLineSpace	✓	✓	✓	✓	✓
addTextRotate	✓	✓	✓	✓	✓
addText	✓	✓	✓	✓	✓
addTextLang	✓	✓	✓	✓	✓
addTextFont	✓	✓	✓	✓	✓
addTextSmooth	✓	✓	✓	✓	✓
addTextDouble	✓	✓	✓	✓	✓
addTextSize	✓	✓	✓	✓	✓
addTextStyle	✓	✓	✓	✓	✓
addTextPosition	✓	✓	✓	✓	✓
addFeedUnit	✓	✓	✓	✓	✓
addFeedLine	✓	✓	✓	✓	✓
addImage	✓	✓	✓	✓	✓
addImage(Previous format)	✓	✓	✓	✓	✓
addImage(Previous format)	✓	✓	✓	✓	✓
addLogo	✓	✓	✓	✓	✓
addBarcode	✓	✓	✓	✓	✓
addSymbol	✓	✓	✓	✓	✓
addPageBegin	✓	✓	✓	✓	✓
addPageEnd	✓	✓	✓	✓	✓
addPageArea	✓	✓	✓	✓	✓
addPageDirection	✓	✓	✓	✓	✓
addPagePosition	✓	✓	✓	✓	✓
addPageLine	-	-	-	✓	✓
addPageRectangle	-	-	-	✓	✓
addCut	✓	✓	✓	✓	✓

API	TM-m10	TM-m30	TM-T60	TM-T88VI	TM-T88VI-iHUB
addPulse	✓	✓	✓	✓	✓
addSound	✓	✓	-	✓	✓
addSound(Previous format)	✓	✓	-	✓	✓
addFeedPosition	-	-	-	✓	✓
addLayout	-	-	-	✓	✓
addCommand	✓	✓	✓	✓	✓

TM-m10

The TM-m10 model information is listed in the table below.

		58 mm
Resolution		203 dpi x 203 dpi (W x H)
Country		ANK model Japanese model Traditional Chinese model
Print Width		420 dots
Characters in a Line	Font A	ANK: 35 characters Kanji ^{*1} : 17 characters
	Font B	ANK: 42 characters Kanji ^{*2} : 21 characters
	Font C	ANK: 46 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H) Kanji ^{*1} : 24 dots x 24 dots (W x H)
	Font B	ANK: 10 dots x 24 dots (W x H) Kanji ^{*2} : 20 dots x 24 dots (W x H)
	Font C	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	ANK: At the 21st dot from the top of the character Kanji ^{*1} : At the 21st dot from the top of the character
	Font B	ANK: At the 21st dot from the top of the character Kanji ^{*2} : At the 21st dot from the top of the character
	Font C	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Page Mode Default Area		420 dots x 2400 dots (W x H)
Page Mode Maximum Area		420 dots x 2400 dots (W x H)
Barcode		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
Two-Dimensional Code		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbology not supported)
Paper Cut		Cut, Feed cut
Drawer Kick-Out		Supported
Buzzer		Option (Pattern A ~ Pattern E, Error, No paper, Stop)
Battery		Not supported
Bluetooth® connection		Supported only by the TM-m10 Bluetooth® model.

*1 Differs depending on the Multilingual Model specifications.

*2 Only for Japanese model.

API information

For information about the APIs, refer to "ePOS-Print SDK for Android User's Manual".

The differential information for the APIs is listed in the table below.

API	Parameter	Specifiable Settings Value	Description
Builder	printerModel	"TM-m10"	<ul style="list-style-type: none"> • TM-m10 USB model • TM-m10 Ethernet model • TM-m10 Wi-Fi model • TM-m10 <i>Bluetooth</i>® model
	lang	Builder.MODEL_ANK	ANK model
		Builder.MODEL_JAPANESE	Japanese model
		Builder.MODEL_TAIWAN	Traditional Chinese model
addTextFont	font	Builder.FONT_A	Font A
		Builder.FONT_B	Font B
		Builder.FONT_C	Font C
addImage	mode	Builder.MODE_MONO	Monochrome (2 tone)
		Builder.MODE_GRAY16	Multiple tone (16 tone)
		Builder.PARAM_DEFAULT	Default value((Monochrome (2 tone))
	compress	Builder.COMPRESS_DEFLATE	Image compression is carried out.
		Builder.COMPRESS_NONE	Image compression is not carried out.
		Builder.PARAM_DEFAULT	Default value (Image compression is not carried out)

TM-m30

The TM-m30 model information is listed in the table below.

		58 mm	80 mm
Resolution		203 dpi x 203 dpi (W x H)	
Country		ANK model Japanese model Simplified Chinese model Traditional Chinese model Korean model	
Print Width		420 dots	576 dots
Characters in a Line	Font A	ANK: 35 characters Kanji ^{*1} : 17 characters	ANK: 48 characters Kanji ^{*1} : 24 characters
	Font B	ANK: 42 characters Kanji ^{*2} : 21 characters Kanji ^{*3} : 26 characters	ANK: 57 characters Kanji ^{*2} : 28 characters Kanji ^{*3} : 36 characters
	Font C	ANK: 46 characters	ANK: 64 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H) / Kanji ^{*1} : 24 dots x 24 dots (W x H)	
	Font B	10 dots x 24 dots (W x H) Kanji ^{*2} : 20 dots x 24 dots (W x H) / Kanji ^{*3} : 16 dots x 16 dots (W x H)	
	Font C	9 dots x 17 dots (W x H)	
Character Baseline	Font A	ANK: At the 21st dot from the top of the character Kanji ^{*1} : At the 21st dot from the top of the character	
	Font B	At the 21st dot from the top of the character Kanji ^{*2} : At the 21st dot from the top of the character Kanji ^{*3} : At the 15th dot from the top of the character	
	Font C	At the 16th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		420 dots x 2400 dots (W x H)	576 dots x 2400 dots (W x H)
Page Mode Maximum Area		420 dots x 2400 dots (W x H)	576 dots x 2400 dots (W x H)
Barcode		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	
Two-Dimensional Code		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbology not supported)	
Paper Cut		Cut, Feed cut	
Drawer Kick-Out		Supported	
Buzzer		Option (Pattern A ~ Pattern E, Error, No paper, Stop)	
Battery		Not supported	
Bluetooth connection		Supported only by the TM-m30 Bluetooth® model.	

*1 Differs depending on the Multilingual Model specifications.

*2 Only for Japanese model.

*3 Only for Korean model.

API information

For information about the APIs, refer to "ePOS-Print SDK for Android User's Manual".

The differential information for the APIs is listed in the table below.

API	Parameter	Specifiable Settings Value	Description
Builder	printerModel	"TM-m30"	<ul style="list-style-type: none"> TM-m30 Standard model TM-m30 <i>Bluetooth</i>[®] model
	lang	Builder.MODEL_ANK	ANK model
		Builder.MODEL_JAPANESE	Japanese model
		Builder.MODEL_CHINESE	Simplified Chinese model
		Builder.MODEL_TAIWAN	Traditional Chinese model
Builder.MODEL_KOREAN	Korean model		
addTextFont	font	Builder.FONT_A	Font A
		Builder.FONT_B	Font B
		Builder.FONT_C	Font C
addImage	mode	Builder.MODE_MONO	Monochrome (2 tone)
		Builder.MODE_GRAY16	Multiple tone (16 tone)
		Builder.PARAM_DEFAULT	Default value((Monochrome (2 tone))
	compress	Builder.COMPRESS_DEFLATE	Image compression is carried out.
		Builder.COMPRESS_NONE	Image compression is not carried out.
		Builder.PARAM_DEFAULT	Default value (Image compression is not carried out)

TM-T60

The TM-T60 model information is listed in the table below.

		80 mm
Resolution		203 dpi x 203 dpi (W x H)
Country		Simplified Chinese model
Print Width		576 dots
Characters in a Line	Font A	ANK: 48 characters Kanji: 24 characters
	Font B	ANK: 64 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H) Kanji: 24 dots x 24 dots (W x H)
	Font B	9 dots x 17 dots (W x H)
Character Baseline	Font A	ANK: At the 21st dot from the top of the character Kanji: At the 21th dot from the top of the character
	Font B	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Page Mode Default Area		576 dots x 831 dots (W x H)
Page Mode Maximum Area		576 dots x 1662 dots (W x H)
Barcode		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128
Two-Dimensional Code		PDF417, QR Code
Paper Cut		Cut, Feed cut
Drawer Kick-Out		Supported
Buzzer		Not supported
Battery		Not supported

API information

For information about the APIs, refer to "ePOS-Print SDK for Android User's Manual".

The differential information for the APIs is listed in the table below.

API	Parameter	Specifiable Settings Value	Description
Builder	printerModel	"TM-T60"	TM-T60
	lang	Builder.MODEL_CHINESE	Simplified Chinese model
addTextFont	font	Builder.FONT_A	Font A
		Builder.FONT_B	Font B
addImage	mode	Builder.MODE_MONO	Monochrome (2 tone)
		Builder.PARAM_DEFAULT	Default value((Monochrome (2 tone))
	compress	Builder.COMPRESS_NONE	Image compression is not carried out.
		Builder.PARAM_DEFAULT	Default value (Image compression is not carried out)

TM-T88VI

The TM-T88VI model information is listed in the table below.

		58 mm	80 mm
Resolution		180 dpi x 180 dpi (W x H)	
Country		ANK model Japanese model Simplified Chinese model Traditional Chinese model Korean model South Asian model	
Print Width		360 dots	512 dots
Characters in a Line	Font A	ANK: 30 characters Kanji ^{*1} : 15 characters	ANK: 42 characters Kanji ^{*1} : 21 characters
	Font B	ANK: 40 characters Kanji ^{*2} : 22 characters	ANK: 56 characters Kanji ^{*2} : 32 characters
	Special font A ^{*3}	30 characters	42 characters
	Special font B ^{*3}	40 characters	56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H) / Kanji ^{*1} : 24 dots x 24 dots (W x H)	
	Font B	ANK: 9 dots x 17 dots (W x H) / Kanji ^{*2} : 16 dots x 16 dots (W x H)	
	Special font A ^{*3}	12 dots x 24 dots (W x H)	
	Special font B ^{*3}	9 dots x 24 dots (W x H)	
Character Baseline	Font A	ANK: At the 21st dot from the top of the character Kanji ^{*1} : At the 21st dot from the top of the character	
	Font B	ANK: At the 16th dot from the top of the character Kanji ^{*2} : At the 15th dot from the top of the character	
	Special font A ^{*3}	At the 20th dot from the top of the character	
	Special font B ^{*3}	At the 20th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		360 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
Page Mode Maximum Area		360 dots x 2400 dots (W x H)	512 dots x 2400 dots (W x H)
Barcode		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	
Two-Dimensional Code		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbology not supported)	
Paper Cut		Cut, Feed cut	
Drawer Kick-Out		Supported	
Buzzer		Option (Pattern A ~ Pattern E, Error, No paper, Stop)	
Battery		Not supported	
Bluetooth connection		Supported only by the TM-T88VI Bluetooth® model.	

*1 Differs depending on the Multilingual Model specifications.

*2 Only for Korean model.

*3 Only for South Asian model.

API information

For information about the APIs, refer to "ePOS-Print SDK for Android User's Manual".

The differential information for the APIs is listed in the table below.

API	Parameter	Specifiable Settings Value	Description
Builder	printerModel	"TM-T88VI"	TM-T88VI
	lang	Builder.MODEL_ANK	ANK model
		Builder.MODEL_JAPANESE	Japanese model
		Builder.MODEL_CHINESE	Simplified Chinese model
		Builder.MODEL_TAIWAN	Traditional Chinese model
		Builder.MODEL_KOREAN	Korean model
		Builder.MODEL_SOUTHASIA	South Asian model
addTextFont	font	Builder.FONT_A	Font A
		Builder.FONT_B	Font B
addImage	mode	Builder.MODE_MONO	Monochrome (2 tone)
		Builder.MODE_GRAY16	Multiple tone (16 tone)
		Builder.PARAM_DEFAULT	Default value((Monochrome (2 tone))
	compress	Builder.COMPRESS_DEFLATE	Image compression is carried out.
		Builder.COMPRESS_NONE	Image compression is not carried out.
		Builder.PARAM_DEFAULT	Default value (Image compression is not carried out)

TM-T88VI-iHUB

The TM-T88VI-iHUB model information is listed in the table below.

		58 mm	80 mm
Resolution		180 dpi x 180 dpi (W x H)	
Country		ANK model Japanese model Simplified Chinese model Traditional Chinese model Korean model South Asian model	
Print Width		360 dots	512 dots
Characters in a Line	Font A	ANK: 30 characters Kanji ^{*1} : 15 characters	ANK: 42 characters Kanji ^{*1} : 21 characters
	Font B	ANK: 40 characters Kanji ^{*2} : 22 characters	ANK: 56 characters Kanji ^{*2} : 32 characters
	Special font A ^{*3}	30 characters	42 characters
	Special font B ^{*3}	40 characters	56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H) / Kanji ^{*1} : 24 dots x 24 dots (W x H)	
	Font B	ANK: 9 dots x 17 dots (W x H) / Kanji ^{*2} : 16 dots x 16 dots (W x H)	
	Special font A ^{*3}	12 dots x 24 dots (W x H)	
	Special font B ^{*3}	9 dots x 24 dots (W x H)	
Character Baseline	Font A	ANK: At the 21st dot from the top of the character Kanji ^{*1} : At the 21st dot from the top of the character	
	Font B	ANK: At the 16th dot from the top of the character Kanji ^{*2} : At the 15th dot from the top of the character	
	Special font A ^{*3}	At the 20th dot from the top of the character	
	Special font B ^{*3}	At the 20th dot from the top of the character	
Default Line Feed Space		30 dots	
Color Specification		First color	
Page Mode Default Area		360 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
Page Mode Maximum Area		360 dots x 2400 dots (W x H)	512 dots x 2400 dots (W x H)
Barcode		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	
Two-Dimensional Code		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Aztec Code, Data Matrix (Composite Symbology not supported)	
Paper Cut		Cut, Feed cut	
Drawer Kick-Out		Supported	
Buzzer		Option (Pattern A ~ Pattern E, Error, No paper, Stop)	
Battery		Not supported	

*1 Differs depending on the Multilingual Model specifications.

*2 Only for Korean model.

*3 Only for South Asian model.

API information

For information about the APIs, refer to "ePOS-Print SDK for Android User's Manual".

The differential information for the APIs is listed in the table below.

API	Parameter	Specifiable Settings Value	Description
Builder	printerModel	"TM-T88VI"	TM-T88VI
	lang	Builder.MODEL_ANK	ANK model
		Builder.MODEL_JAPANESE	Japanese model
		Builder.MODEL_CHINESE	Simplified Chinese model
		Builder.MODEL_TAIWAN	Traditional Chinese model
		Builder.MODEL_KOREAN	Korean model
		Builder.MODEL_SOUTHASIA	South Asian model
addTextFont	font	Builder.FONT_A	Font A
		Builder.FONT_B	Font B
addImage	mode	Builder.MODE_MONO	Monochrome (2 tone)
		Builder.MODE_GRAY16	Multiple tone (16 tone)
		Builder.PARAM_DEFAULT	Default value((Monochrome (2 tone))
	compress	Builder.COMPRESS_DEFLATE	Image compression is carried out.
		Builder.COMPRESS_NONE	Image compression is not carried out.
		Builder.PARAM_DEFAULT	Default value (Image compression is not carried out)