

Epson ePOS SDK for JavaScript

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.

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

© Seiko Epson Corporation 2016-2018. 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

Migration Overview.....5

■ Migration Types.....	5
Migration from ePOS-Print SDK	5
Migration from ePOS-Device SDK.....	5

Migration from ePOS-Print SDK.....6

■ Migration with ePOS-Print SDK-compatible APIs	6
Migration procedure.....	6
SDK file replacement.....	6
Application development with ePOS-Print SDK-compatible APIs	6
■ Migration with the Epson ePOS SDK APIs	7
Migration procedure.....	7
SDK file replacement.....	8
Changing package names	8
Changing objects.....	8
Establishing and cutting the communication with the printer.....	9
Printing from ePOS-Print API	11
Printing from Canvas API.....	14
Monitoring of the status.....	17
Changing API name.....	20
Changing API objects	22
Changing events	23

Migration from ePOS-Device SDK.....24

■ Migration with Epson ePOS SDK APIs	24
Migration procedure.....	24
SDK file replacement.....	24

Appendix.....25

■ ePOS-Print SDK-compatible API.....	25
TM-m10.....	25
TM-m30.....	28
TM-T88VI.....	31
TM-T88VI-iHUB	33

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

Migration from ePOS-Print SDK

You can use the following methods for applications that use ePOS-Print SDK.

- ☐ Migration with compatible APIs for Epson ePOS SDK

Epson ePOS SDK contains ePOS-Print SDK-compatible APIs and can use existing APIs as they are. Existing applications can be migrated into an Epson ePOS SDK-compatible application by making minimal changes, such as modifying the referenced library file names.

- ☐ Migration with Epson ePOS SDK APIs

By modifying the existing application into a program that uses Epson ePOS SDK APIs, you can migrate to an Epson ePOS SDK-compatible application. This requires many program modifications, such as changing the programming sequence and APIs.

The support policy for ePOS-Print SDK-compatible APIs and Epson ePOS SDK towards new products and new functions is described below.

New products and functions	ePOS-Print SDK-compatible API	Epson ePOS SDK
Epson TM printers	✓	✓
New functions of Epson TM printers	- *	✓
New tablets and smartphone models	✓	✓

✓ : Compatible -: Incompatible

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

Migration from ePOS-Device SDK

Existing applications can be migrated into an Epson ePOS SDK-compatible application by making minimal changes, such as modifying the referenced library file names.

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 ".

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.

Modify the file referenced from the program.

Type	ePOS-Print SDK	ePOS-Print SDK-compatible API
Library	epos-print-5.x.x.js	epos-2.x.x.js

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 JavaScript 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 JavaScript 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.

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 objects	Changing the ePOS-Print SDK object to the Epson ePOS SDK object Refer to " Changing objects ".
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 from ePOS-Print API Refer to "Printing from ePOS-Print API". • Printing from Canvas API Refer to "Printing from Canvas API". • 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 objects Refer to "Changing API objects". <input type="checkbox"/> Changing events Refer to "Changing events".

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.

Modify the file referenced from the program.

Type	ePOS-Print SDK	Epson ePOS SDK
Library	epos-print.5.0.0.js	epos-2.0.0.js

Changing package names

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

Package names to change

Type	ePOS-Print SDK	Epson ePOS SDK
ePOSPrint object	window.epson.ePOSPrint	window.epson.ePOSDevice
ePOSBuilder object	window.epson.ePOSBuilder	

Changing objects

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

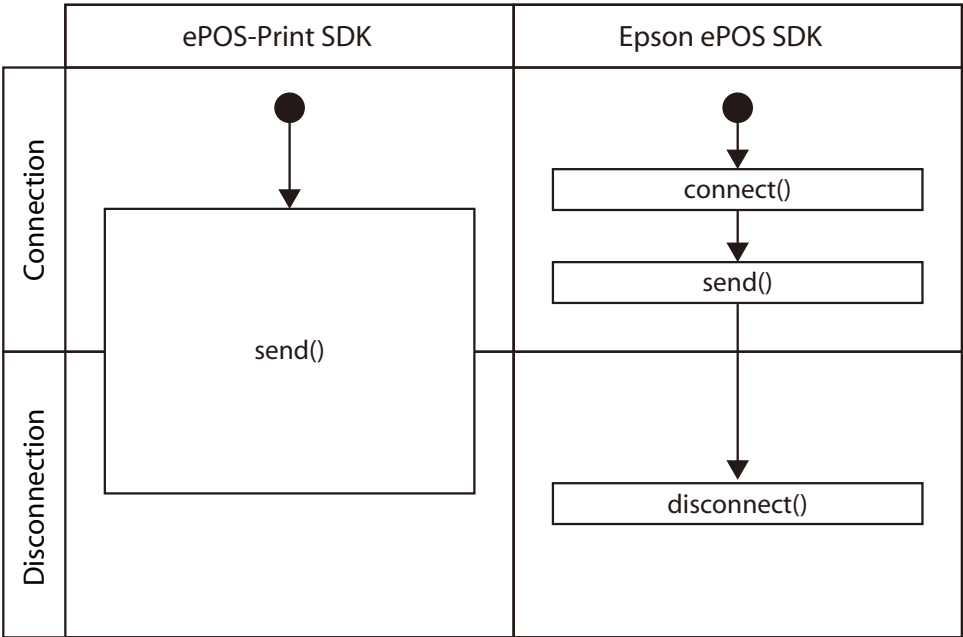
Package names to change

Type	ePOS-Print SDK	Epson ePOS SDK
Print function	ePOSBuilder	Printer
	ePOSPrint	
	CanvasPrint	

Establishing and cutting the communication with the printer

ePOS-Print SDK and Epson ePOS SDK have different APIs for connecting and disconnecting communication with the printer.
Use the following as a reference when modifying your program.

Execution procedure differences



Program differences

❑ ePOS-Print SDK

```
var address = 'http://192.168.192.168/cgi-bin/epos/service.cgi?devid=local_printer&time?out=60000';
//Create ePOSPrint object
var epos = new epson.ePOSPrint(address);
//Create ePOSBuilder object
var builder = new epson.ePOSBuilder();

//Register the event
epos.onreceive = function (res) { alert(res.success); };

//Create the printing data
builder.addText('Hello\n');

//Send the printing data
epos.send(builder.toString());
```

❑ Epson ePOS SDK

```

var ePosDev = new epos.ePOSDevice();
var printer = null;

function connect(){
    //Connect to device
    ePosDev.connect('192.168.192.168', '8008', callback_connect);
}

function callback_connect(resultConnect){
    if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
        //Get the Printer object
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_PRINTER, {'crypto' :
            false, 'buffer' : false}, callback_createDevice);
    }
    else {
        //Display the error message
    }
}

function callback_createDevice(deviceObj, retcode){
    if( retcode == 'OK' ) {
        printer = deviceobj;
        printer.timeout = 60000;
        //Register the printing complete event
        printer.onreceive = function (res) { alert(res.success); };
        print();
    } else {
        alert(retcode);
    }
}

function print(){
    //Create the printing data
    printer.addText('Hello\n');

    //Send the printing data
    printer.send();
}

function disconnect(){
    //Discard the Printer object
    ePosDev.deleteDevice(printer, callback_deleteDevice);
}

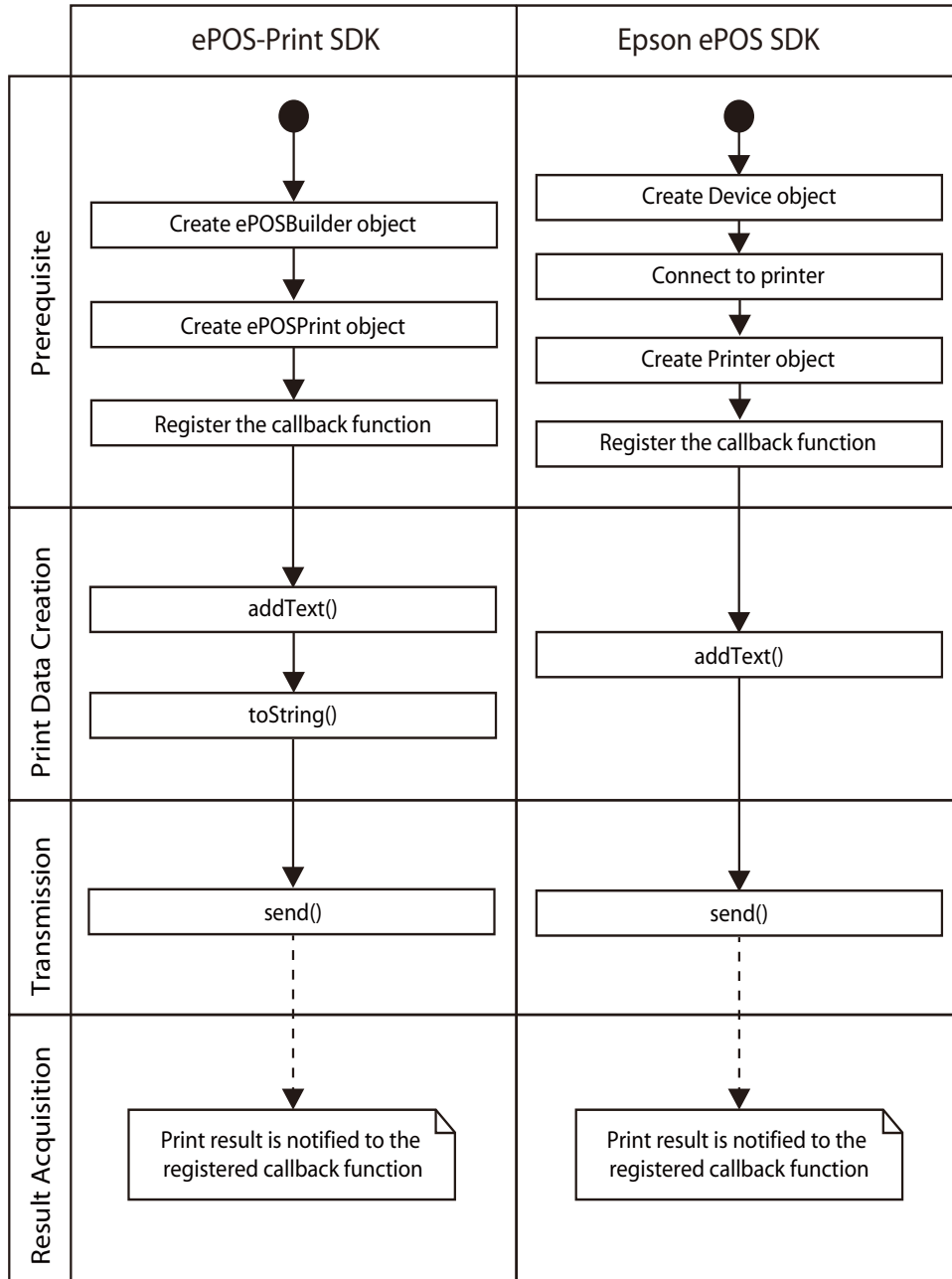
function callback_deleteDevice(errorCode){
    //Disconnect to device
    ePosDev.disconnect();
}

```

Printing from ePOS-Print API

ePOS-Print and Epson ePOS SDK have different procedures for executing the printing function. Use the following as a reference when modifying your program.

Execution procedure differences



Callback: ----▶

Program differences

❑ ePOS-Print SDK

```
var address = 'http://192.168.192.168/cgi-bin/epos/service.cgi?devid=local_printer&time?out=60000';
//Create ePOSPrint object
var epos = new epson.ePOSPrint(address);
//Create ePOSBuilder object
var builder = new epson.ePOSBuilder();

//Register the event
epos.onreceive = function (res) { alert(res.success); };

//Create the printing data
builder.addText('Hello\n');

//Send the printing data
epos.send(builder.toString());
```

❑ Epson ePOS SDK

```

var ePosDev = new epos.ePOSDevice();
var printer = null;
function connect(){
    //Connect to device
    ePosDev.connect('192.168.192.168', '8008', callback_connect);
}

function callback_connect(resultConnect){
    if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
        //Get the Printer object
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_PRINTER, {'crypto' :
            false, 'buffer' : false}, callback_createDevice);
    }
    else {
        //Display the error message
    }
}

function callback_createDevice(deviceObj, retcode){
    if( retcode == 'OK' ) {
        printer = deviceobj;
        printer.timeout = 60000;
        //Register the printing complete event
        printer.onreceive = function (res) { alert(res.success); };
        print();
    } else {
        alert(retcode);
    }
}

function print(){
    //Create the printing data
    printer.addText('Hello\n');

    //Send the printing data
    printer.send();
}

function disconnect(){
    //Discard the Printer object
    ePosDev.deleteDevice(printer, callback_deleteDevice);
}

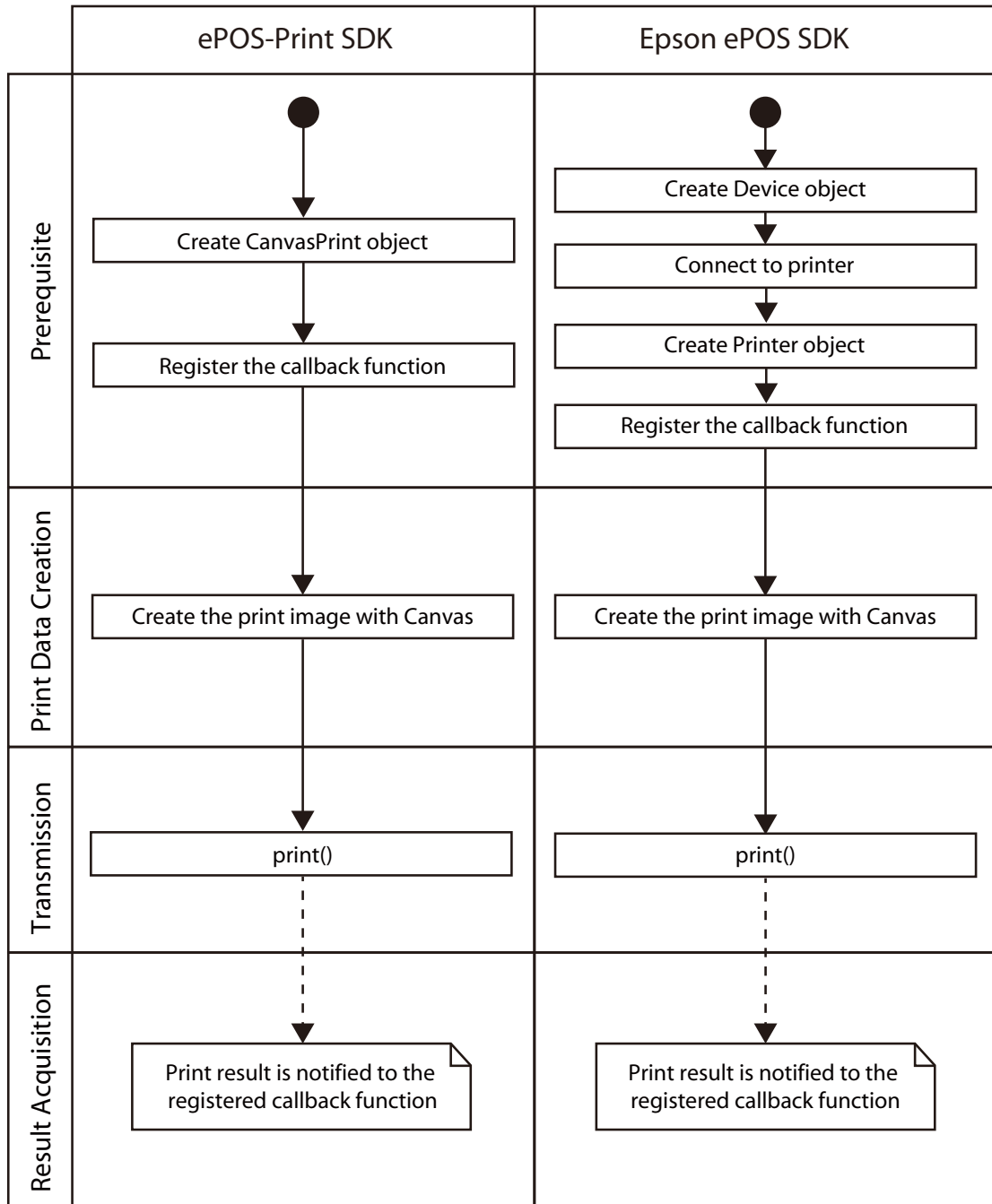
function callback_deleteDevice(errorCode){
    //Disconnect to device
    ePosDev.disconnect();
}

```

Printing from Canvas API

ePOS-Print and Epson ePOS SDK have different procedures for executing the printing function from Canvas API. Use the following as a reference when modifying your program.

Execution procedure differences



Callback: ----▶

Program differences

❑ ePOS-Print SDK

```
//Draw in Canvas
var canvas = document.getElementById('canvas');
if (canvas.getContext) {
    var context = canvas.getContext('2d');
    context.fillText("Test", 100, 100);
}

var address = 'http://192.168.192.168/cgi-bin/epos/service.cgi?devid=local_printer&time
out=60000';
var epos = new Epson.CanvasPrint(address);

//Register the event
epos.onreceive = function (res) { alert(res.success); };
epos.cut = true;
epos.mode = epos.MODE_MONO;

//Perform printing
epos.print(canvas);
```

❑ Epson ePOS SDK

```

var ePosDev = new epson.ePOSDevice();
var printer = null;

function connect(){
    //Connect to device
    ePosDev.connect('192.168.192.168', '8008', callback_connect);
}

function callback_connect(resultConnect){
    if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
        //Get the Printer object
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_PRINTER, {'crypto' :
false, 'buffer' : false}, callback_createDevice);
    }
    else {
        //Display the error message
    }
}

function callback_createDevice(deviceObj, retcode){
    if( retcode == 'OK' ) {
        printer = deviceObj;
        printer.timeout = 60000;
        //Register the printing complete event
        printer.onreceive = function (res) { alert(res.success); };
        print();
    } else {
        alert(retcode);
    }
}

function print(){
    //Create the printing data
    //Draw in Canvas
    var canvas = document.getElementById('canvas');
    if (canvas.getContext) {
        var context = canvas.getContext('2d');
        context.fillText("Test", 100, 100);
    }

    var cut = true;
    var mode = printer.MODE_MONO;

    //Send the printing data
    printer.print(canvas, cut, mode);
}

function disconnect(){
    //Discard the Printer object
    ePosDev.deleteDevice(printer, callback_deleteDevice);
}

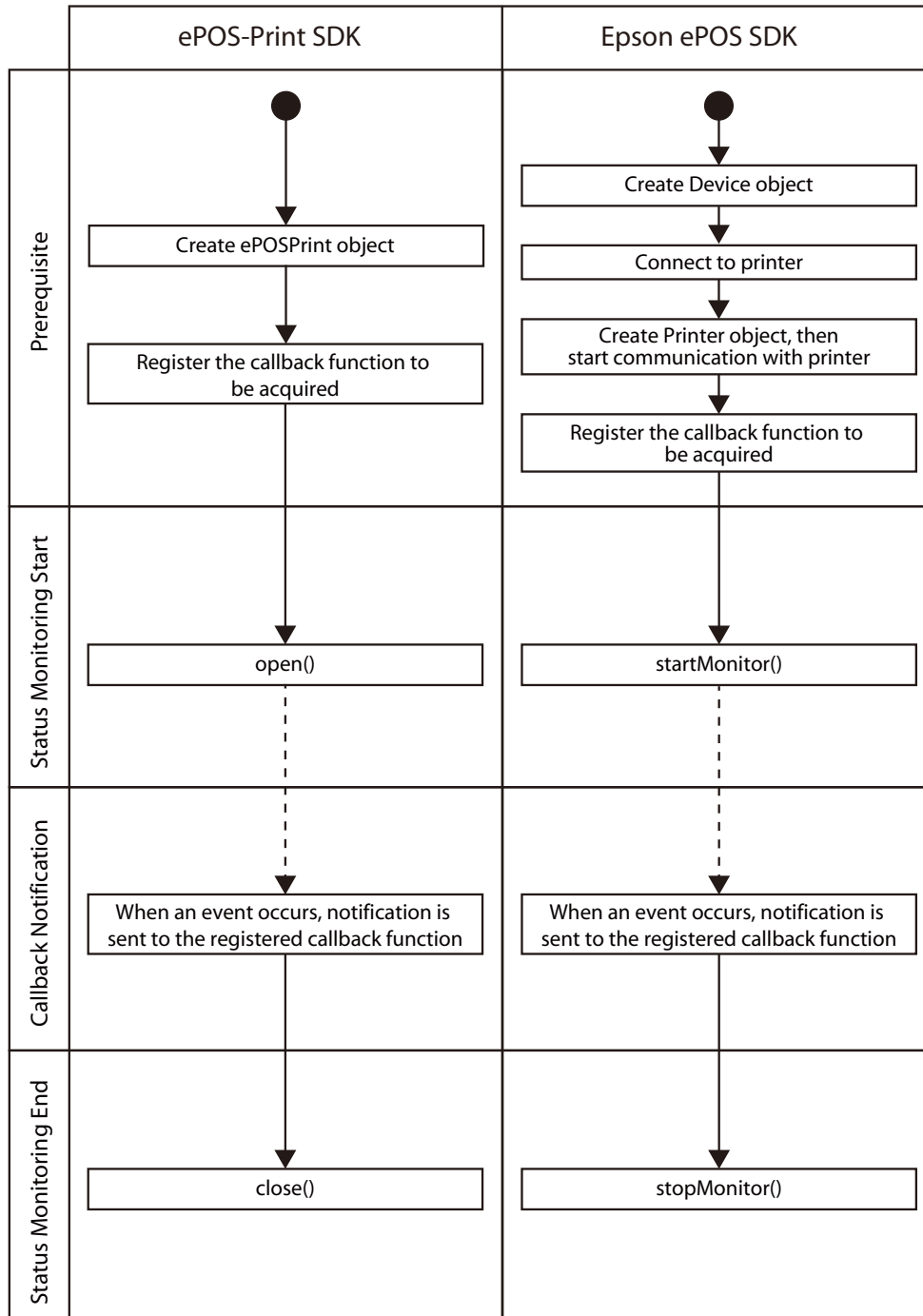
function callback_deleteDevice(errorCode){
    //Disconnect to device
    ePosDev.disconnect();
}

```


Monitoring of the status

ePOS-Print and Epson ePOS SDK have different procedures for executing the status obtaining function. Use the following as a reference when modifying your program.

Execution procedure differences



Callback: ----▶

Program differences

❑ ePOS-Print SDK

```
var address = 'http://192.168.192.168/cgi-bin/epos/service.cgi?devid=local_printer&time
out=60000';
var epos = new epos.ePOSPrint(address);

//Register the event
epos.onstatuschange = function (status) { alert(status); };
epos.oncoveropen = function () { alert('coveropen'); };

function open(){
    //Status monitoring start
    epos.open();
}

//Cover open

function close(){
    //Status monitoring End
    epos.close();
}
```

❑ Epson ePOS SDK

```

var ePosDev = new epos.ePOSDevice();
var printer = null;

function connect(){
    //Connect to device
    ePosDev.connect('192.168.192.168', '8008', callback_connect);
}

function callback_connect(resultConnect){
    if ((resultConnect == 'OK') || (resultConnect == 'SSL_CONNECT_OK')) {
        //Get the Printer object
        ePosDev.createDevice('local_printer', ePosDev.DEVICE_TYPE_PRINTER, {'crypto' :
false, 'buffer' : false}, callback_createDevice);
    }
    else {
        //Display the error message
    }
}

function callback_createDevice(deviceObj, retcode){
    printer = deviceObj;
    if( retcode == 'OK' ) {
        printer = devobj;
        printer.timeout = 60000;
        //Register the printing complete event
        printer.onstatuschange = function (status) { alert(status); };
        printer.oncoveropen = function () { alert('coveropen'); };
        print();
    } else {
        alert(retcode);
    }
}

function startMonitor(){
    //Status monitoring start
    printer.startMonitor();
}

//Printer cover open

function startMonitor(){
    //Status monitoring end
    printer.stopMonitor();
}

function disconnect(){
    //Discard Printer object
    ePosDev.deleteDevice(printer, callback_deleteDevice);
}

function callback_deleteDevice(errorCode){
    //Disconnect to device
    ePosDev.disconnect();
}

```

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 JavaScript User's Manual" and "Epson ePOS SDK for JavaScript User's Manual".

A list of API names to be changed

Object	ePOS-Print SDK	Epson ePOS SDK
Function		
ePOSPrint object and CanvasPrint object		
Enables status event operation	open	startMonitor
Disables status event operation	close	stopMonitor
The URL of the TM printer is specified	address	connect
CanvasPrint object		
Adds the paper cut to command buffer	cut	print
It specifies the color mode	mode	

A list of APIs to be deleted

Object	ePOS-Print SDK	Response
Function		
ePOSBuilder object		
Obtains a print document	toString	This is not required for Epson ePOS SDK. For detail, refer to "Printing from ePOS-Print API" on page 11 .
ePOSPrint object and CanvasPrint object		
Retains the enabled/disabled setting for status event operation	enabled	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) If you want to confirm whether status events are enabled or disabled, carry out management in the application.
Holds the TM printer's status	status	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) Use the status event and obtain the status.
Holds the TM printer's battery status	battery	
Obtains a communication error event	onerror	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) Use the ondisconnect method and obtain the communication error event.
CanvasPrint object		
Sets the position alignment	align	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) If you want to set the placement for the image data rendered in the canvas, you cannot use the print method. Use the addTextAlign, addImage, and send methods to set the placement.

Object	Function	ePOS-Print SDK	Response
	Sets the printing color	color	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) If you want to set the color for the image data rendered in the canvas, you cannot use the print method. Use the addImage and send methods to set the color.
	Sets the feed of label paper/black mark paper	feed	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) If you want to set paper feeding for label paper or black-mark paper, you cannot use the print method. Use the addImage, addFeed, and send methods to set paper feeding for label paper or black-mark paper.
	Sets the paper type	paper	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) If you want to set the paper type, you cannot use the print method. Use the addLayout, addFeed, and send methods to set the paper type.
	Sets the paper layout	layout	This property cannot be used with Epson ePOS SDK. (Only ePOS-Print SDK-compatible APIs can be used.) If you want to set the paper layout, you cannot use the print method. Use the addLayout, addImage, and send methods to set the paper layout.

Changing API objects

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

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

List of APIs to be changed from ePOSBuilder objects to Printer objects

API		
addText	addTextLang	addTextLang
addTextRotate	addTextLineSpace	addTextFont
addTextSmooth	addTextDouble	addTextSize
addTextStyle	addTextPosition	addTextVPosition
addFeedUnit	addFeedLine	addFeed
addFeedPosition	addImage	addLogo
addBarcode	addSymbol	addHLine
addVLineBegin	addVLineEnd	addPageBegin
addPageEnd	addPageArea	addPageDirection
addPagePosition	addPageLine	addPageRectangle
addCut	addPulse	addSound
addLayout	addRecovery	addReset
addCommand	halftone	brightness
force	message	

List of APIs to be changed from ePOSPrint objects to Printer objects

API		
send	interval	onreceive
onstatuschange	onbatterystatuschange	ononline
onoffline	onpoweroff	oncoverok
oncoveropen	onpaperok	onpaperend
onpapernearend	ondrawerclosed	ondraweropen
onbatteryok	onbatterylow	timeout

List of APIs to be changed from CanvasPrint objects to Printer objects

API		
print	interval	timeout
halftone	brightness	onreceive
onstatuschange	onbatterystatuschange	ononline
onoffline	onpoweroff	oncoverok
oncoveropen	onpaperok	onpaperend
onpapernearend	ondrawerclosed	ondraweropen
onbatteryok	onbatterylow	recover
reset		

Changing events

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

To see what has changed, compare APIs in the “ePOS-Print SDK for JavaScript User’s Manual” and “Epson ePOS SDK for JavaScript User’s Manual”.

Events to be changed

Function	ePOS-Print SDK	Epson ePOS SDK
Notification of printer disconnect status	onerror	ondisconnect

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 Epson ePOS SDK 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	Library file replacement Refer to " SDK file replacement ".

This completes the migration with Epson ePOS SDK APIs.

SDK file replacement

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

Modify the file referenced from the program.

Type	ePOS-Device SDK	Epson ePOS SDK
Library	epos-device-3.x.x.js	epos-2.x.x.js

Appendix

ePOS-Print SDK-compatible API

This section explains model information for new Epson TM printer products and APIs supported by ePOS-Print SDK-compatible APIs.

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 Symbolology 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.

ePOS-Print settings

Item	Value
Printing Method	Thermal (203 dpi)
Character Code Tables	Page 0-5,16-19,20-21,26,30-31,11-15,32-53

ePOS-Print Builder API supported API list

ePOS-Print Builder object		
Constructor	addTextAlign method	addTextLineSpace method
addTextRotate method	addText method	addTextLang method
addTextFont method	addTextSmooth method	addTextDouble method
addTextSize method	addTextStyle method	addTextPosition method
addTextVPosition method	addFeedUnit method	addFeedLine method
addFeed method	addImage method	addLogo method
addBarcode method	addSymbol method	addPageBegin method
addPageEnd method	addPageArea method	addPageDirection method
addPagePosition method	addCut method	addPulse method
addSound method	addRecovery method	addReset method
addCommand method	toString method	halftone property
brightness property	force property	message property

ePOS-Print API supported API list

ePOSPrint object		
Constructor	send method	open method
close method	address property	enabled property
interval property	status property	timeout property
onreceive event	onerror event	onstatuschange event
ononline event	onoffline event	onpoweroff event
oncoverok event	oncoveropen event	onpaperok event
onpapernearend event	onpaperend event	ondrawerclosed event
ondraweropen event		

ePOS-Print Canvas API supported API list

ePOS-Print Canvas object		
Constructor	print method	open method
close method	recover method	reset method
address property	enabled property	interval property
status property	timeout property	halftone property
brightness property	cut property	mode property
align property	onreceive event	onerror event
onstatuschange event	ononline event	onoffline event
onpoweroff event	oncoverok event	oncoveropen event
onpaperok event	onpapernearend event	onpaperend event
ondrawerclosed event	ondraweropen event	

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.

ePOS-Print settings

Item	Value
Printing Method	Thermal (203 dpi)
Character Code Tables	Page 0-5,16-19,20-21,26,30-31,11-15,32-53

ePOS-Print Builder API supported API list

ePOS-Print Builder object		
Constructor	addTextAlign method	addTextLineSpace method
addTextRotate method	addText method	addTextLang method
addTextFont method	addTextSmooth method	addTextDouble method
addTextSize method	addTextStyle method	addTextPosition method
addTextVPosition method	addFeedUnit method	addFeedLine method
addFeed method	addImage method	addLogo method
addBarcode method	addSymbol method	addPageBegin method
addPageEnd method	addPageArea method	addPageDirection method
addPagePosition method	addCut method	addPulse method
addSound method	addRecovery method	addReset method
addCommand method	toString method	halftone property
brightness property	force property	message property

ePOS-Print API supported API list

ePOSPrint object		
Constructor	send method	open method
close method	address property	enabled property
interval property	status property	timeout property
onreceive event	onerror event	onstatuschange event
ononline event	onoffline event	onpoweroff event
oncoverok event	oncoveropen event	onpaperok event
onpapernearend event	onpaperend event	ondrawerclosed event
ondraweropen event		

ePOS-Print Canvas API supported API list

ePOS-Print Canvas object		
Constructor	print method	open method
close method	recover method	reset method
address property	enabled property	interval property
status property	timeout property	halftone property
brightness property	cut property	mode property
align property	onreceive event	onerror event
onstatuschange event	ononline event	onoffline event
onpoweroff event	oncoverok event	oncoveropen event
onpaperok event	onpapernearend event	onpaperend event
ondrawerclosed event	ondraweropen event	

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	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	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 Symbolism 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.

ePOS-Print settings

Item	Value
Printing Method	Thermal (203 dpi)
Character Code Tables	Page 0-5,16-19,20-21,26,30-31,11-15,32-53

ePOS-Print Builder API supported API list

ePOS-Print Builder object		
Constructor	addTextAlign method	addTextLineSpace method
addTextRotate method	addText method	addTextLang method
addTextFont method	addTextSmooth method	addTextDouble method
addTextSize method	addTextStyle method	addTextPosition method
addTextVPosition method	addFeedUnit method	addFeedLine method
addFeed method	addFeedPosition method	addImage method
addLogo method	addBarcode method	addSymbol method
addPageBegin method	addPageEnd method	addPageArea method
addPageDirection method	addPagePosition method	addPageLine method
addPageRectangle method	addCut method	addPulse method
addSound method	addLayout method	addRecovery method
addReset method	addCommand method	toString method
halftone property	brightness property	force property
message property		

ePOS-Print API supported API list

ePOSPrint object		
Constructor	send method	open method
close method	address property	enabled property
interval property	status property	timeout property
onreceive event	onerror event	onstatuschange event
ononline event	onoffline event	onpoweroff event
oncoverok event	oncoveropen event	onpaperok event
onpapernearend event	onpaperend event	ondrawerclosed event
ondraweropen event		

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	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	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 Symbolism 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.

ePOS-Print settings

Item	Value
Printing Method	Thermal (203 dpi)
Character Code Tables	Page 0-5,16-19,20-21,26,30-31,11-15,32-53

ePOS-Print Builder API supported API list

ePOS-Print Builder object		
Constructor	addTextAlign method	addTextLineSpace method
addTextRotate method	addText method	addTextLang method
addTextFont method	addTextSmooth method	addTextDouble method
addTextSize method	addTextStyle method	addTextPosition method
addTextVPosition method	addFeedUnit method	addFeedLine method
addFeed method	addFeedPosition method	addImage method
addLogo method	addBarcode method	addSymbol method
addPageBegin method	addPageEnd method	addPageArea method
addPageDirection method	addPagePosition method	addPageLine method
addPageRectangle method	addCut method	addPulse method
addSound method	addLayout method	addRecovery method
addReset method	addCommand method	toString method
halftone property	brightness property	force property
message property		

ePOS-Print API supported API list

ePOSPrint object		
Constructor	send method	open method
close method	address property	enabled property
interval property	status property	timeout property
onreceive event	onerror event	onstatuschange event
ononline event	onoffline event	onpoweroff event
oncoverok event	oncoveropen event	onpaperok event
onpapernearend event	onpaperend event	ondrawerclosed event
ondraweropen event		

ePOS-Print Canvas API supported API list

ePOS-Print Canvas object		
Constructor	print method	open method
close method	recover method	reset method
address property	enabled property	interval property
status property	timeout property	halftone property
brightness property	cut property	mode property
align property	onreceive event	onerror event
onstatuschange event	ononline event	onoffline event
onpoweroff event	oncoverok event	oncoveropen event
onpaperok event	onpapernearend event	onpaperend event
ondrawerclosed event	ondraweropen event	