

# ePOS-Print SDK for iOS

## User's Manual

---

### Overview

Describes the features and development environment.

### Sample Program

Describes how to use the sample program.

### Programming Guide

Describes how to write programs in application development.

### API Reference

Describes the APIs provided in ePOS-Print SDK for iOS.

### Command Transmission/Reception

Describes the APIs for transmitting and receiving commands.

### Appendix

Describes the specifications for printers used for the ePOS-Print SDK for iOS.

## 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 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 designated 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 and ESC/POS are registered trademarks or trademarks of Seiko Epson Corporation.

Xcode®, iPhone®, iPod touch®, iPad® and iTunes® are either registered trademarks or trademarks of Apple Inc. in the United States and other countries.

IOS® is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Wi-Fi® is a registered trademark of the Wi-Fi Alliance®.

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.

QR Code® is a registered trademark of DENSO Wave Incorporated.

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

## ESC/POS® Command System

Epson took the initiative by introducing ESC/POS, a proprietary POS printer command system, which includes patented or patent pending commands and enables versatile POS system construction with high scalability.

Compatible with all types of Epson POS printers and displays, this proprietary control system also offers the flexibility to easily make future upgrades. Its popularity is worldwide.



ESC/POS is designed to reduce the processing load on the host computer in POS environments. It comprises a set of highly functional and efficient commands that enables the full realization of the potential of printers.

© Seiko Epson Corporation 2012-2015. All rights reserved.

## For Safety

### Key to Symbols

The symbols in this manual are identified by their level of importance, as defined below. Read the following carefully before handling the product.

|   |   |
|---|---|
|  | Provides information that must be observed to avoid damage to your equipment or a mal-function. |
|  | Provides important information and useful tips.   |

## 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.

# About this Manual

## Aim of the Manual

This manual aims to provide development engineers with all the information necessary for the construction and design of a printing system that uses ePOS-Print SDK, and for the development and design of printer applications.

## Manual Content

The manual is made up of the following sections:

|           |   |
|-----------|---|
| Chapter 1 | <a href="#">Overview</a>                                      |
| Chapter 2 | <a href="#">Sample Program</a>                                |
| Chapter 3 | <a href="#">Programming Guide</a>                             |
| Chapter 4 | <a href="#">API Reference</a>                                 |
| Chapter 5 | <a href="#">Command Transmission/Reception</a>                |
| Appendix  | <a href="#">List of Supported APIs for Each Printer Model</a> |
|           | <a href="#">Support Information by Printer</a>                |
|           | <a href="#">Cautions</a>                                      |

# Contents

|                            |   |
|----------------------------|---|
| ■ For Safety .....         | 3 |
| Key to Symbols .....       | 3 |
| ■ Restriction of Use ..... | 3 |
| ■ About this Manual .....  | 4 |
| Aim of the Manual .....    | 4 |
| Manual Content .....       | 4 |
| ■ Contents.....            | 5 |

---

## Overview ..... 9

|   |    |
|---|----|
| ■ Overview of ePOS-Print SDK.....                                     | 9  |
| Features.....   | 9  |
| Function .....  | 10 |
| Developing an Application that Performs Bluetooth Communication ..... | 11 |
| ■ Operating Environment .....   | 12 |
| iOS Version .....   | 12 |
| iOS Device .....  | 12 |
| Printer .....   | 13 |
| Development Environment.....  | 13 |
| ■ Contents in the Package.....  | 14 |
| Package .....   | 14 |
| Manual.....   | 14 |
| Sample Program .....  | 15 |
| Download .....  | 15 |
| ■ Restrictions.....   | 16 |

---

## Sample Program ..... 17

|   |    |
|---|----|
| ■ Overview.....                         | 17 |
| ■ Usage Environment .....               | 18 |
| Usage Environment.....                  | 18 |
| Printer .....                           | 18 |
| Target device .....                     | 18 |
| ■ Environmental Construction.....       | 19 |
| ■ How to Use the Program Sample .....   | 20 |
| Search for printers and printing.....   | 20 |
| Perform Bluetooth Pairing .....         | 27 |
| Acquisition of Printer Model Name ..... | 28 |
| Printer Selection Using QR Code.....    | 29 |
| QR Code Printing .....                  | 30 |

---

## **Programming Guide..... 31**

|   |           |
|---|-----------|
| ■ <b>How to Incorporate the ePOS-Print SDK for iOS.....</b> | <b>31</b> |
| ■ <b>ePOS-Print SDK.....</b>                                | <b>33</b> |
| Print Mode .....  | 33        |
| Programming Flow .....                                      | 33        |
| Printer Selection .....                                     | 34        |
| Print Document Creation .....                               | 36        |
| Transmission of Print Document.....                         | 39        |
| Printing After Checking the Printer Status.....             | 41        |
| ■ <b>Automatic Acquisition of Printer Status .....</b>      | <b>43</b> |
| Event List .....  | 44        |
| ■ <b>Status .....</b>                                       | <b>45</b> |
| Error Statuses and Actions to Take .....                    | 45        |
| Printer Statuses and Actions to Take.....                   | 47        |
| Battery Status .....  | 48        |

---

## **API Reference ..... 49**

|                                 |           |
|---------------------------------|-----------|
| ■ <b>ePOS-Print API.....</b>    | <b>49</b> |
| initWithPrinterModel .....      | 52        |
| clearCommandBuffer .....        | 54        |
| addTextAlign .....              | 55        |
| addTextLineSpace .....          | 56        |
| addTextRotate .....             | 57        |
| addText .....                   | 58        |
| addTextLang .....               | 59        |
| addTextFont .....               | 60        |
| addTextSmooth .....             | 61        |
| addTextDouble .....             | 62        |
| addTextSize .....               | 63        |
| addTextStyle.....               | 64        |
| addTextPosition .....           | 66        |
| addFeedUnit .....               | 67        |
| addFeedLine.....                | 68        |
| addImage .....                  | 69        |
| addImage(Previous format).....  | 72        |
| addImage(Previous format).....  | 75        |
| addLogo.....                    | 77        |
| addBarcode .....                | 78        |
| addSymbol.....                  | 84        |
| addPageBegin .....              | 89        |
| addPageEnd.....                 | 90        |
| addPageArea.....                | 91        |
| addPageDirection.....           | 92        |
| addPagePosition .....           | 94        |
| addPageLine .....               | 95        |
| addPageRectangle .....          | 97        |
| addCut .....                    | 99        |
| addPulse.....                   | 100       |
| addSound .....                  | 101       |
| addSound(Previous format) ..... | 103       |

|   |            |
|---|------------|
| addFeedPosition .....                     | 105        |
| addLayout .....                           | 106        |
| addCommand.....                           | 108        |
| init .....                                | 109        |
| openPrinter .....                         | 110        |
| openPrinter(Previous format).....         | 112        |
| openPrinter(Previous format).....         | 115        |
| closePrinter .....                        | 117        |
| sendData .....                            | 118        |
| sendData(Previous format).....            | 120        |
| setStatusChangeEventCallback .....        | 122        |
| setOnlineEventCallback.....               | 123        |
| setOfflineEventCallback.....              | 124        |
| setPowerOffEventCallback .....            | 125        |
| setCoverOkEventCallback .....             | 126        |
| setCoverOpenEventCallback .....           | 127        |
| setPaperOkEventCallback .....             | 128        |
| setPaperNearEndEventCallback .....        | 129        |
| setPaperEndEventCallback.....             | 130        |
| setDrawerClosedEventCallback .....        | 131        |
| setDrawerOpenEventCallback .....          | 132        |
| setBatteryLowEventCallback .....          | 133        |
| setBatteryOkEventCallback .....           | 134        |
| setBatteryStatusChangeEventCallback ..... | 135        |
| <b>■ Printer Search API.....</b>          | <b>136</b> |
| start.....                                | 136        |
| stop.....                                 | 137        |
| getDeviceInfoList.....                    | 138        |
| getResult(Previous format) .....          | 140        |
| <b>■ Printer Easy Select API .....</b>    | <b>141</b> |
| parseQR .....                             | 141        |
| createQR .....                            | 142        |
| deviceType .....                          | 143        |
| printerName .....                         | 143        |
| macAddress .....                          | 143        |
| <b>■ Log Setting API.....</b>             | <b>144</b> |
| setLogSettings.....                       | 144        |
| <b>■ Bluetooth Connection API .....</b>   | <b>147</b> |
| init .....                                | 149        |
| connectDevice .....                       | 150        |
| disconnectDevice .....                    | 151        |

---

## **Command Transmission/Reception..... 153**

|                                     |            |
|-------------------------------------|------------|
| <b>■ Programming.....</b>           | <b>153</b> |
| Programming Flow .....              | 153        |
| Initializing the EpsonIo Class..... | 154        |
| Opening a Device Port .....         | 154        |
| Sending Data .....                  | 154        |
| Receive data .....                  | 155        |
| Closing the Device Port.....        | 155        |
| <b>■ List of Error Values.....</b>  | <b>156</b> |

|   |     |
|---|-----|
| ■ Command Transmission/Reception API Reference..... | 157 |
| init .....  | 157 |
| open.....   | 158 |
| close.....  | 159 |
| write .....   | 160 |
| read.....   | 162 |

---

## Appendix..... 163

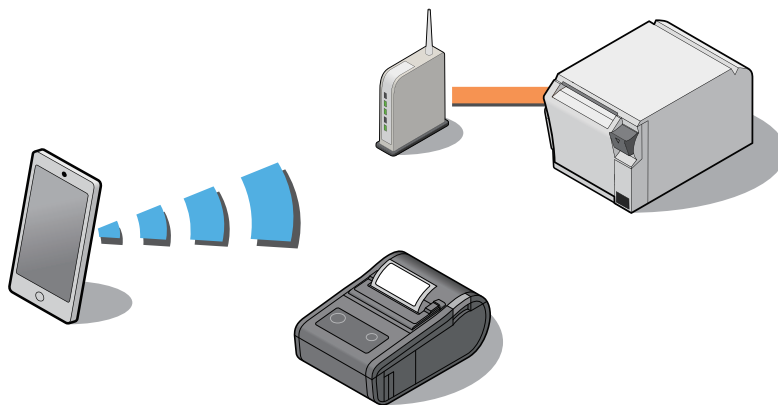
|   |     |
|---|-----|
| ■ List of Supported APIs for Each Printer Model .....       | 163 |
| ■ Support Information by Printer .....                      | 164 |
| TM-P20.....   | 164 |
| TM-P20 iOS Bluetooth model .....                            | 166 |
| TM-P60.....   | 168 |
| TM-P60(Peeler) iOS Bluetooth model.....                     | 170 |
| TM-P60(Receipt) iOS Bluetooth model .....                   | 172 |
| TM-P60II .....  | 174 |
| TM-P60II iOS Bluetooth model .....                          | 176 |
| TM-P80.....   | 178 |
| TM-P80 iOS Bluetooth model .....                            | 180 |
| TM-T20 .....  | 182 |
| TM-T20II.....   | 183 |
| TM-T20II iOS Bluetooth model .....                          | 184 |
| TM-T70 .....  | 185 |
| TM-T70II.....   | 186 |
| TM-T70II iOS Bluetooth model .....                          | 187 |
| TM-T81II.....   | 188 |
| TM-T82 .....  | 189 |
| TM-T82II.....   | 190 |
| TM-T88V .....   | 191 |
| TM-T88V iOS Bluetooth model .....                           | 192 |
| TM-T90II.....   | 193 |
| TM-U220 .....   | 194 |
| TM-U330 .....   | 195 |
| ■ Cautions .....  | 196 |
| If you Use the Printer from Multiple Mobile Terminals ..... | 196 |



# Overview

This chapter describes the features of and the specifications for ePOS-Print SDK for iOS.

## Overview of ePOS-Print SDK



The ePOS-Print SDK for iOS is an SDK aimed at development engineers who are developing iOS applications for printing on an Epson TM printer. Applications are developed using the APIs provided by ePOS-Print SDK. The ePOS-Print SDK also has the "ePOS-Print SDK for Android" for Android applications.



APIs for transmitting/receiving commands to/from TM printers are also provided. A command transmission/reception API cannot be used with the ePOS-Print API, EposPrint class. For details on the command transmission/reception APIs, refer to [Command Transmission/Reception \(p.153\)](#).

### Features

- ❑ Allows printing to TM printers from iOS applications.
- ❑ Allows acquisition of TM printer status from iOS applications.

## Function

### **ePOS-Print API**

- ☐ Print setting (alignment/line feed space/text rotation/page mode)
- ☐ Character data setting (language/font (device font)/double-sizing/scale/smoothing/print position)
- ☐ Character style setting (inversion of black and white/underline/bold)
- ☐ Paper feed setting (in dots/in lines)
- ☐ Image printing (raster image/NV graphics)
- ☐ Barcode printing  
(For barcodes that can be printed by each model, refer to [Support Information by Printer \(p.164\).](#))
- ☐ 2D-Code printing  
(For 2D-Code that can be printed by each model, refer to [Support Information by Printer \(p.164\).](#))
- ☐ Drawer kick function
- ☐ Buzzer function
- ☐ Paper layout setting
- ☐ Label / black mark paper feed setting
- ☐ ESC/POS command transmission
- ☐ Acquisition of response from printer (printing result / printer status / battery status)
- ☐ Compatible with Asian languages (simplified Chinese, traditional Chinese, Korean, Thai, Vietnamese)

### **Printer Search API**

- ☐ Search for printers

### **Printer Easy Select API**

- ☐ Select a printer easily  
(You can select a printer easily by using QR code.)

### **Log Setting API**

- ☐ Log output setting  
(This API allows to output log data to an iOS device's storage and a server that can establish TCP connection.)



Log data output to an iOS device can be saved on other computers using a USB connection.

### **Bluetooth® Connection API**

- ☐ Pairing connection for *Bluetooth*

## Developing an Application that Performs *Bluetooth* Communication

If registering an application that uses *Bluetooth* in the App Store, Epson must submit an application to Apple in advance. Please apply for each application you want to register in the App Store from the URL below.

<https://c4b.epson-biz.com/ais/E>

# Operating Environment

## iOS Version

- ❑ iOS Ver.4.2 to 4.3.5
- ❑ iOS Ver.5.0 to 5.1.1
- ❑ iOS Ver.6.0 to 6.1.4
- ❑ iOS Ver.7.0 to 7.1.2
- ❑ iOS Ver.8.0 to 8.0.2
- ❑ iOS Ver.8.2



For the latest version, refer to the README file.

## iOS Device

- ❑ iPhone 3G/ iPhone 3GS/ iPhone 4/ iPhone 4s/ iPhone 5/ iPhone 5s/ iPhone 5c/ iPhone 6/ iPhone 6 Plus
- ❑ iPod touch (2nd generation)/ iPod touch (3rd generation) / iPod touch (4th generation) / iPod touch (5th generation)
- ❑ iPad/ iPad 2/ iPad (3rd generation)/ iPad 4/ iPad Air/ iPad Air 2/ iPad mini/ iPad mini 2 (iPad mini with Retina display)/ iPad mini 3

## Printer

| TM Printer                                   | Interface |        |           |
|--|-----------|--------|-----------|
|  | Wired LAN | Wi-Fi® | Bluetooth |
| TM-P20                                       | -         | ✓      | -         |
| TM-P20 iOS <i>Bluetooth</i> model            | -         | -      | ✓         |
| TM-P60(Receipt) Wi-Fi                        | -         | ✓      | -         |
| TM-P60(Receipt) iOS <i>Bluetooth</i> model   | -         | -      | ✓         |
| TM-P60(Peeler) Wi-Fi                         | -         | ✓      | -         |
| TM-P60(Peeler) iOS <i>Bluetooth</i> model    | -         | -      | ✓         |
| TM-P60II(Receipt) Wi-Fi                      | -         | ✓      | -         |
| TM-P60II(Receipt) iOS <i>Bluetooth</i> model | -         | -      | ✓         |
| TM-P60II(Peeler) Wi-Fi                       | -         | ✓      | -         |
| TM-P60II(Peeler) iOS <i>Bluetooth</i> model  | -         | -      | ✓         |
| TM-P80 Wi-Fi                                 | -         | ✓      | -         |
| TM-P80 iOS <i>Bluetooth</i> model            | -         | -      | ✓         |
| TM-T20                                       | ✓         | -      | -         |
| TM-T20II                                     | ✓         | -      | -         |
| TM-T20II iOS <i>Bluetooth</i> model          | -         | -      | ✓         |
| TM-T70                                       | ✓         | ✓      | -         |
| TM-T70II                                     | ✓         | ✓      | -         |
| TM-T70II iOS <i>Bluetooth</i> model          | -         | -      | ✓         |
| TM-T81II                                     | ✓         | -      | -         |
| TM-T82                                       | ✓         | -      | -         |
| TM-T82II                                     | ✓         | -      | -         |
| TM-T88V                                      | ✓         | ✓      | -         |
| TM-T88V iOS <i>Bluetooth</i> model           | -         | -      | ✓         |
| TM-T90II                                     | ✓         | ✓      | -         |
| TM-U220 Series                               | ✓         | ✓      | -         |
| TM-U330 Series                               | ✓         | ✓      | -         |



- In the TM printer settings, set only Receive Buffer Full for the Busy Condition. Regarding the settings, see the Technical Reference Guide for the printer.
- For a wireless LAN, Infrastructure mode and Adhoc mode can be used.

## Development Environment

The following are necessary to develop an iOS application.

- ❑ Xcode Ver.4.2 or later

# Contents in the Package

## Package

| File                                       | Description  |
|--|--|
| ePOS-Print.h                               | Header file that includes class definitions and error value / device type constant definitions.    |
| libeposprint.a                             | Library for function execution.<br>(armv6, armv7, armv7s, arm64, i386, x86_64 supported)           |
| ePOSEasySelect.h                           | Header file for selecting a printer easily.  |
| libeposeasyselect.a                        | Library for selecting a printer easily.<br>(armv7, armv7s, arm64, i386, x86_64 supported)          |
| ePOS-Print_Sample_iOS.zip                  | A sample program file.   |
| README.en.txt                              | A readme file.   |
| README.jp.txt                              | A readme file. (The Japanese-language edition)   |
| EULA.en.txt                                | Contains the SOFTWARE LICENSE AGREEMENT.   |
| EULA.jp.txt                                | Contains the SOFTWARE LICENSE AGREEMENT.<br>(The Japanese-language edition)                        |
| ePOS-Print_SDK_iOS_en_revx.pdf             | This manual.   |
| ePOS-Print_SDK_iOS_ja_revx.pdf             | The Japanese-language edition of this manual.  |
| ePOS-Print_SDK_iOS_AppDevGuide_en_revx.pdf | Describes the procedure for building a development environment.                                    |
| ePOS-Print_SDK_iOS_AppDevGuide_ja_revx.pdf | Describes the procedure for building a development environment.<br>(The Japanese-language edition) |

## Manual

The following manuals are available for ePOS-Print SDK for iOS.

- ❑ ePOS-Print SDK for iOS User's Manual (This Document)
- ❑ ePOS-Print SDK for iOS Application Development - Setup Guide

## Sample Program

For an iOS application for TM printers developed using ePOS-Print SDK, the following program is available.

- ❑ ePOS-Print\_Sample\_iOS.zip
  - Basic function sample (ePOSPrintSample)
  - Easy Select sample (ePOSEasySelectSample)

## Download

For customers in North America, go to the following web site:

<http://www.epsonexpert.com/>

For customers in other countries, go to the following web site:

<https://download.epson-biz.com/?service=pos>

# Restrictions

- ❑ A communication API ([p.51](#)) and command transmission/reception API ([p.153](#)) in the ePOS-Print APIs cannot be used for the same device at the same time.
- ❑ A maximum of 16 device ports can be opened in the same application at the same time.
- ❑ If the device goes into sleep mode while communicating with a printer via *Bluetooth*, the connection will be lost.
- ❑ If you are using *Bluetooth* connection, depending on the iOS specifications, print data may be deleted if it is sent while the printer is not ready to print.

The printer is unable to print in cases such as:

- when the roll paper cover is open.
- when the paper is out.
- when the printer is waiting for a label to be removed. (Peeler model only)



# Sample Program

This chapter describes how to use the sample program.



- The sample program is provided for iOS application development engineers as an implementation sample of an iOS application that used ePOS-Print for iOS API.
- The sample program package is provided as an iOS application project for Xcode including Objective-C source files.

## Overview

The Sample Program has the following functionality.

| Sample Program                          | Description  |
|---|--|
| <b>&lt;ePOSPrintSample&gt;</b><br>      | <p>The following functions are implemented in the sample program:<br/>(The sample program does not contain a functionality for turning text/ images/ barcodes/ etc.)</p> <ul style="list-style-type: none"> <li>• Searching for printers</li> <li>• Pairing connection for <i>Bluetooth</i></li> <li>• Opening of port</li> <li>• Closing of port</li> <li>• Closing of port and releasing pairing of <i>Bluetooth</i></li> <li>• Text printing</li> <li>• Graphic printing (image file printing)</li> <li>• Barcode printing</li> <li>• 2D-Code printing</li> <li>• Printing in page mode</li> <li>• Paper cutting</li> <li>• Printer status acquisition</li> <li>• Acquisition of printer model name/language information</li> <li>• Log output setting</li> <li>• Display of status event</li> <li>• Display of battery status event</li> </ul> |
| <b>&lt;ePOSEasySelectSample&gt;</b><br> | <p>Connects to a printer easily by using QR code.</p> <ul style="list-style-type: none"> <li>• Obtain printer information using QR code.</li> <li>• Analyze the obtained printer information.</li> <li>• Open a port using the results of analysis.</li> <li>• Create printer information QR code from printer search results.</li> </ul>  |

# Usage Environment

## Usage Environment

- Xcode Ver.4.2 or later



For details about ways to construct a development environment, please refer to the "ePOS-Print SDK for iOS Application Development - Setup Guide".

## Printer

- TM printer supported in ePOS-Print SDK.

## Target device

- Device connected to a computer via USB

# Environmental Construction

Follow the procedures below to use the sample program.

- 1** Extract the sample program zip file to a directory of your choosing.
- 2** Double-click "ePOSPrintSample.xcodeproj", or "ePOSEasySelectSample.xcodeproj" in the directory to which the zip file has been extracted.
- 3** Xcode will start up. Select your target device as the "Scheme."
- 4** Click the (Run) button on the upper left.
- 5** The sample program will be installed to the target iOS device, and then the program will start up.

# How to Use the Program Sample

This section describes how to use the program sample for the following operations:

❑ ePOSPrintSample

- Search for printers and printing ([p. 20](#))
- Perform *Bluetooth* Pairing ([p. 27](#))
- Acquisition of Printer Model Name ([p. 28](#))

❑ ePOSEasySelectSample

- Search for printers and printing ([p. 20](#))
- Acquisition of Printer Model Name ([p. 28](#))

## Search for printers and printing

Use the sample program as follows:

- 1** Start the sample program. For details, refer to [Environmental Construction \(p.19\)](#).
- 2** Search for printers. Tap (Printer Discovery) on the main screen.  
When you select (Device Type), the IP addresses/Mac addresses/ Printer name for the detected printers are listed.
- 3** Tap the printer to use from the displayed list.
- 4** Open the printer's port. Tap (Open) on the main screen.  
The "Device Type" and "IP Address/Mac Address" of the printer selected in procedure 3 are displayed. Select (Printer Name) and (Language).
- 5** Set (Status Monitor).

| Item     | Description   |
|----------|---|
| Enabled  | <ul style="list-style-type: none"><li>• ON: The status monitor is enabled and the printer status is monitored.</li><li>• OFF: The status monitor is disabled.</li></ul> |
| Interval | When Enabled is turned ON, the status monitoring interval is set in units of milliseconds.  |

- 6** Tap (Open).

## 7 Execute the following processes:

| Process                    | Description   |
|----------------------------|---|
| Text printing              | Tap (Text) on the main screen.<br>For details, refer to <a href="#">Text printing (p.22)</a> .              |
| Graphic printing           | Tap (Image) on the main screen.<br>For details, refer to <a href="#">Graphic printing (p.22)</a> .          |
| Barcode printing           | Tap (Barcode) on the main screen.<br>For details, refer to <a href="#">Barcode printing (p.23)</a> .        |
| 2D-Code printing           | Tap (2D Code) on the main screen.<br>For details, refer to <a href="#">2D-Code printing (p.23)</a> .        |
| Printing in page mode      | Tap (Page Mode) on the main screen.<br>For details, refer to <a href="#">Printing in page mode (p.24)</a> . |
| Paper cutting              | Tap (Cut) on the main screen.<br>For details, refer to <a href="#">Paper cutting (p.24)</a> .               |
| Printer status acquisition | Tap (Get Status) on the main screen.  |
| Log output setting         | Tap (Log Settings) on the main screen.<br>For details, refer to <a href="#">Log output setting (p.24)</a> . |

## 8 The following execution results will be displayed:

- Process execution result (error status / printer status / battery status)  
For details, refer to [Process execution result \(p.25\)](#).
- Method (API) execution error  
For details, refer to [Method \(API\) execution error \(p.26\)](#).

## 9 When all processing is finished, tap (Close) on the main screen, and close the printer's port.

---

### ***Text printing***

Execute the text printing according to the following procedure:

- 1** Enter a string to print for (Print Characters).
- 2** Specifies the character properties for the string to print. The following properties can be specified:

| Property     | Description                                       |
|--------------|---|
| Font         | Set the character font.                           |
| Align        | Set the alignment.                                |
| Line Spacing | Set the line feed space.                          |
| Language     | Set the language.                                 |
| Size         | Set the character scales (vertical / horizontal). |
| Style        | Set the character style (bold / underlining).     |
| X Position   | Set the horizontal start position.                |
| Feed Unit    | Set the paper feed amount.                        |

- 3** Tap (Print) to print.

---

### ***Graphic printing***

Execute the graphic printing according to the following procedure:

- 1** Tap (Select Image) to select an image file to print.
- 2** Tap (Color Mode) to select the tone.



[Gray 16] (multiple tone printing) can only be selected on the TM-T88V/ TM-T70II/ TM-T90II model.

- 3** Tap (Halftone Method) to select the halftone treatment method.
- 4** Tap (Brightness) and input a value to specify brightness.
- 5** Tap (Print) to print.

**Barcode printing**

Execute the barcode printing according to the following procedure:

- 1** Set the following for barcodes:

| Setting                    | Description                                   |
|----------------------------|---|
| Type                       | Select the barcode type.                      |
| Data                       | Enter the barcode data.                       |
| HRI                        | Set the HRI position.                         |
| Font                       | Set the HRI font.                             |
| Module Size(Width, Height) | Set the barcode module size (width / height). |

- 2** Tap (Print) to print.

**2D-Code printing**

Execute the 2D-Code printing according to the following procedure:

- 1** Select the 2D-Code type using (Type).
- 2** Enter the 2D-Code data for (Data).
- 3** Set the following for each 2D-Code:

| Setting  | Description                                  |
|--|--|
| Error Correction Level<br>(PDF417, QR Code, Aztec<br>Code, DataMatrix) | Set the error correction level.              |
| Module Size(Width, Height)   | Set the 2D-Code module size (width / height) |
| Max Size   | Set the maximum 2D-Code size.                |

- 4** Tap (Print) to print.

---

### ***Printing in page mode***

Execute the printing in page mode according to the following procedure:

- 1** Enter a string to print for (Print Characters).
- 2** Set the print area using (Print Area).

| Setting | Description                        |
|---------|------------------------------------|
| X       | Set the origin of horizontal axis. |
| Y       | Set the origin of vertical axis.   |
| Width   | Set the width for the print area.  |
| Height  | Set the height for the print area. |

- 3** Tap (Print) to print.

---

### ***Paper cutting***

Execute the paper cutting according to the following procedure:

- 1** Set whether to cut after feeding paper using (Type).
- 2** Tap (Print) and execute cutting operation.

---

### ***Log output setting***

Use the following procedures:

- 1** Set whether to enable the log output function and the log output destination in (Enabled).
- 2** Set the following items according to the log output destination.

| Setting    | Description   |
|------------|---|
| IP Address | Specify the IP address for TCP communication.                                   |
| Port       | Specify the port number for TCP communication.                                  |
| Log Size   | Specify the maximum size of log data that can be saved on the device's storage. |
| Log Level  | Set the level of log data to be output.   |

- 3** Set the method of saving the settings in (Save Settings Permanently).
- 4** Tap (Setting) to enable the log output settings.
- 5** After printing, check the log file.  
For details, refer to [setLogSettings \(p.144\)](#).



## Execution result

### Process execution result

Any of the following will be displayed:

- Result: Any of the following statuses will be displayed:

| String displayed | Description   |
|------------------|---|
| SUCCESS          | Succeeded   |
| ERR_PARAM        | An invalid parameter was passed.                      |
| ERR_ILLEGAL      | Used in an illegal manner.                            |
| ERR_PROCESSING   | Failed to execute the process.                        |
| ERR_TIMEOUT      | The process was timed out.                            |
| ERR_CONNECT      | Failed to connect to the device.                      |
| ERR_MEMORY       | Could not secure the memory required for the process. |
| ERR_OFF_LINE     | Offline.  |
| ERR_FAILURE      | Another error occurred.                               |

- Status: Any of the following printer statuses will be displayed:

| String displayed | Description   |
|------------------|---|
| NO_RESPONSE      | No response from the printer  |
| PRINT_SUCCESS    | Printing is successfully completed  |
| DRAWER_KICK      | Status of the 3rd pin of the drawer kick-out connector = "H"<br>(Other than TM-P60, TM-P60II, TM-P80) |
| BATTERY_OFFLINE  | Battery offline (TM-P60, TM-P60II, TM-P80)  |
| OFF_LINE         | Offline   |
| COVER_OPEN       | The cover is open   |
| PAPER_FEED       | Paper is being fed by a paper feed switch operation   |
| WAIT_ON_LINE     | Waiting to be brought back online   |
| PANEL_SWITCH     | The paper feed switch is being pressed (ON)   |
| MECHANICAL_ERR   | A mechanical error occurred   |
| AUTOCUTTER_ERR   | An autocutter error occurred  |
| UNRECOVER_ERR    | An unrecoverable error occurred   |
| AUTORECOVER_ERR  | An automatically recoverable error occurred   |
| RECEIPT_NEAR_END | No paper in roll paper near end sensor  |
| RECEIPT_END      | No paper in roll paper end sensor   |

- Battery Status: The following will be displayed.

| String displayed | Description   |
|------------------|---|
| 0xnxxx           | Battery status value<br>For details, refer to <a href="#">Battery Status (p.48)</a> . |

### *Method (API) execution error*

Any of the following will be displayed:

- Error Code: Any of the following statuses will be displayed:

| String displayed | Description   |
|------------------|---|
| ERR_PARAM        | An invalid parameter was passed.                            |
| ERR_OPEN         | The open process failed.                                    |
| ERR_CONNECT      | Failed to connect to the device.                            |
| ERR_TIMEOUT      | All data couldn't be sent during the specified time.        |
| ERR_MEMORY       | Could not secure the memory required for the process.       |
| ERR_ILLEGAL      | Used in an illegal manner.                                  |
| ERR_PROCESSING   | Failed to execute the process.                              |
| ERR_UNSUPPORTED  | An unsupported model or language of use has been specified. |
| ERR_OFF_LINE     | Printer is offline.   |
| ERR_FAILURE      | Another error occurred.                                     |

- Method: The API in which a method execution error occurred is displayed.

## Perform *Bluetooth* Pairing

Use the following procedure:



The *Bluetooth* pairing connection function has restrictions such as models, firmware, and iOS versions. For details, refer to *Bluetooth* Connection API (p. 147).

- 1** Tap (Printer Discovery) on the main screen.
- 2** Tap (*Bluetooth*).
- 3** The devices that have been paired and “Other...” are displayed in (Printer List). Tap (Other...).
- 4** Devices that can be paired are displayed on the (Select An Accessory) screen. Tap the device you want to use.
- 5** The (Error Code) screen is displayed. When “Success” is displayed, tap (OK).

This completes pairing. Tap (Open) on the Main screen to start printing.

To disconnect pairing after printing is complete, tap (Close+Disconnect).

## Acquisition of Printer Model Name



A command transmission/reception API is used for acquisition of printer model name. For details, refer to [Command Transmission/Reception \(p.153\)](#).

Use the following procedure:

- 1** Start the sample program. For details, refer to [Environmental Construction \(p.19\)](#).
- 2** Search for printers. Tap (Printer Discovery) on the main screen.  
When (Device Type) is selected, the printers detected by the search are displayed in list form.
- 3** Tap the printer to use from (Printer List) displayed.
- 4** Tap (Get Printer Name) on the main screen.
- 5** Tap (Get Printer Name).
- 6** The following will be displayed.

| Content displayed | Description  |
|-------------------|--|
| Printer Name      | Displays the model name of the printer.              |
| Language          | Displays the language specifications of the printer. |

## Printer Selection Using QR Code

Use the following procedure:

- 1** Start the sample program. For details, refer to [Environmental Construction \(p.19\)](#).
- 2** Tap (Quick pairing and Easy print by QR code) on the main screen.
- 3** Read the QR code using the camera.  
Put the QR code inside the red frame in the camera preview.
- 4** Tap (Print) to print.

## QR Code Printing

Use the following procedure:

- 1** Start the sample program. For details, refer to [Environmental Construction \(p.19\)](#).
- 2** Tap (Print QR code) on the main screen.
- 3** Tap (Find) in the QR Code Printing window.  
In (Printer List), the detected printers are displayed in list form.
- 4** Select the printer you want to use.
- 5** Tap (Print) to print.

# Programming Guide

This chapter describes how to write programs in the application development using ePOS-Print SDK.



For ways to construct a development environment for iOS applications that use ePOS-Print SDK for iOS, please refer to the "ePOS-Print SDK for iOS Application Development - Setup Guide".

## How to Incorporate the ePOS-Print SDK for iOS

This section explains how to incorporate the ePOS-Print SDK for iOS.  
Incorporate the SDK using following procedures.

- 1** Create a new project in Xcode.
- 2** Drag the following Objective-C headers into any level of the target project in (Project Navigator) for Xcode.
  - ePOS-Print.h
  - ePOSEasySelect.h



ePOSEasySelect.h is required only when Printer Easy Select is used.

- 3** Use the following procedure to integrate ExternalAccessory.framework:
  1. In Project Navigator, select the Project file (the root file).
  2. Select Targets > Build Phases.
  3. Expand "Link Binary With Libraries", and tap the (+) button.
  4. Select ExternalAccessory.framework, and tap the Add button.
- 4** Drag the following static libraries into any level of the target project in (Project Navigator) for Xcode.
  - libeposprint.a
  - libeposeasyselect.a



libeposeasyselect.a is required only when Printer Easy Select is used.

- 5 Write the Objective-C header import declaration in the \*.m source file(s) of the application you would like to use this SDK in as follows:

```
#import "ePOS-Print.h"
```

```
#import "ePOSEasySelect.h"
```



There is no need to define the ePOSEasySelect.h if Printer Easy Select is not used.

- 6 When the *Bluetooth* is used, set the protocol name.  
Set the protocol name according to the following procedure:

| Key                                      | Type    | Value                                   |
|--|---------|---|
| Localization native development region   | String  | en                                      |
| Bundle display name                      | String  | \${PRODUCT_NAME}                        |
| Executable file                          | String  | \${EXECUTABLE_NAME}                     |
| Bundle identifier                        | String  | test.\${PRODUCT_NAME:rfc1034identifier} |
| InfoDictionary version                   | String  | 6.0                                     |
| Bundle name                              | String  | \${PRODUCT_NAME}                        |
| Bundle OS Type code                      | String  | APPL                                    |
| Bundle versions string, short            | String  | 1.0                                     |
| Bundle creator OS Type code              | String  | ????                                    |
| Bundle version                           | String  | 1.0                                     |
| Application requires iPhone environment  | Boolean | YES                                     |
| ► Required device capabilities           | Array   | (1 item)                                |
| ► Supported interface orientations       | Array   | (3 items)                               |
| ▼ Supported external accessory protocols | Array   | (1 item)                                |
| Item 0                                   | String  | com.epson.escpos                        |

1. In Project Navigator, select xxx.plist. (The file name will be Project name-info.)
2. In the pop-up menu, select Add Row.
3. Select "Supported external accessory protocols".
4. Expand the items added in Step 3.
5. Enter com.epson.escpos as the Value for Item 0.



# ePOS-Print SDK

## Print Mode

There are two types of print modes: standard and page modes.

### Standard mode

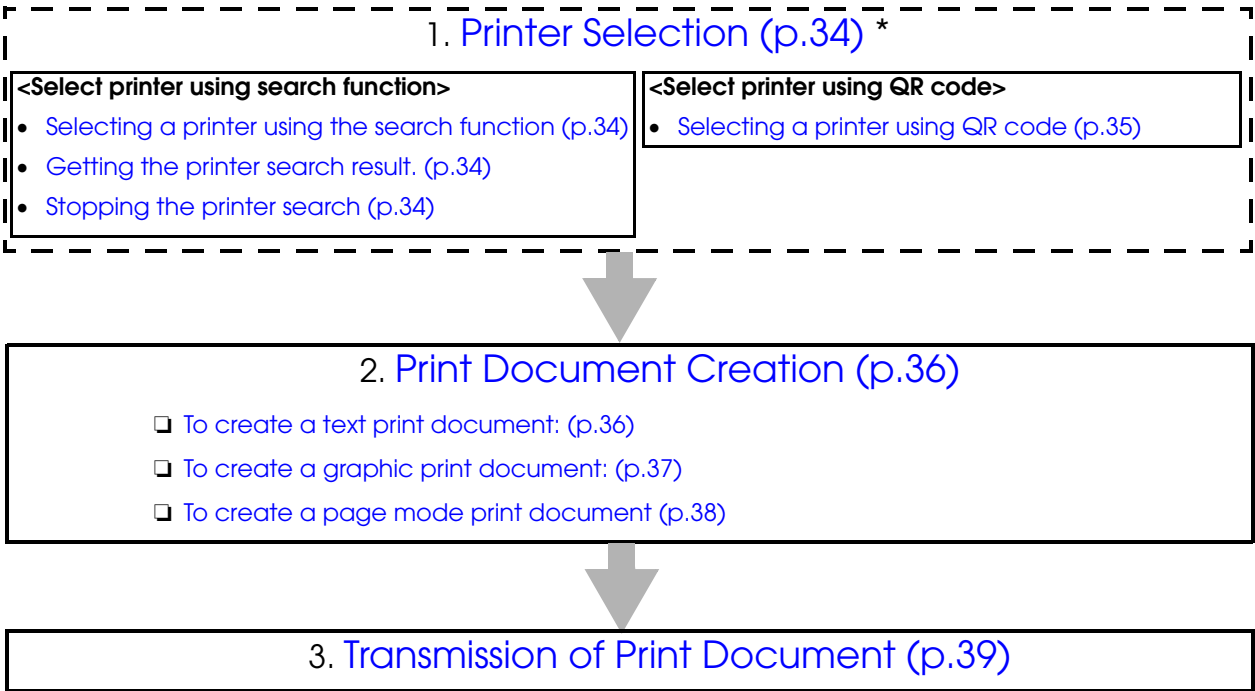
In standard mode, characters are printed line by line. The line feed space is adjusted based on the font size and the height of images, barcodes, etc. This mode is suitable for the type of printing such as printing receipts that requires the paper length to change according to the print space.

### Page mode


In page mode, you set a print area, lay out data in it, and print the data in a batch operation. Characters, images, and barcodes are laid out in the print positions (coordinates).

## Programming Flow

Perform programming following this flow.



\* This is optional.



To ensure successful print operation, write a program in such a way that data is sent after checking the printer status. For the above procedure, refer to [Printing After Checking the Printer Status \(p.41\)](#).

## Printer Selection

### Selecting a printer using the search function

Use the `EpsonIoFinder` class's [start \(p.136\)](#) to start searching for printers. Please refer to the following code.

```
int result = EPSONIO_OC_SUCCESS;
int devicetype = EPSONIO_OC_DEVTYPE_TCP;

//Start Search
switch (devicetype) {
    //Wi-Fi/Ethernet device
    case EPSONIO_OC_DEVTYPE_TCP:
        result = [EpsonIoFinder start:EPSONIO_OC_DEVTYPE_TCP FindOption:option];
        break;
    //Bluetooth device
    case EPSONIO_OC_DEVTYPE_BLUETOOTH:
        result = [EpsonIoFinder start:EPSONIO_OC_DEVTYPE_BLUETOOTH FindOption:option];
        break;
    default:
        result = [EpsonIoFinder start:EPSONIO_OC_DEVTYPE_TCP FindOption:option];
        break;
}
```

### Getting the printer search result.

Use the `EpsonIoFinder` class's [getDeviceInfoList \(p.138\)](#) to get the result of the printer search. Please refer to the following code. Use the obtained results in [openPrinter \(p.110\)](#).

```
int errStatus = EPSONIO_OC_SUCCESS;

//Get device list
NSArray *array = [[NSArray alloc]initWithArray:
    [EpsonIoFinder getDeviceInfoList:&errStatus
    FilterOption:EPSONIO_OC_PARAM_DEFAULT]];
```



Since the printer search takes time to complete, you might not receive any search results if you call the `EpsonIoFinder` class's `getDeviceInfoList` immediately after you call `start`.

### Stopping the printer search

Use the `EpsonIoFinder` class's [stop \(p.137\)](#) to stop searching for printers. Please refer to the following code.

```
//Stop search
int errStatus = [EpsonIoFinder stop];
```

### Selecting a printer using QR code

Use [parseQR \(p.141\)](#) in the EasySelect class to analyze the QR code.

Use the programming example below for your reference. Use the obtained results in [openPrinter \(p.110\)](#).

```
id easySelect = [EposEasySelect alloc];
NSString *data;

//Store the QR code data obtained from the camera image.

//Analyze the QR code
EposEasySelectInfo *easySelectInfo = [easySelect parseQR:data];
if( easySelectInfo == nil){
    //If it is not QR code for EasySelect
    return ;
}

id printer = [[EposPrint alloc] init];

//Open the printer using analyzed data
if ( printer != nil ) {
    errorStatus = [printer openPrinter:easySelectInfo.deviceType
                                DeviceName:easySelectInfo.macAddress];

    //Create an EposBuilder class instance by using the analyzed data.
    id builder = [[EposBuilder alloc]
                  initWithPrinterModel: easySelectInfo.printerName
                  Lang: EPOS_OC_MODEL_ANK];

    errorStatus = [printer closePrinter];
    [printer release];
}
```

### How to Create Printer Easy Select QR code

- ❑ For models that can automatically print Printer Easy Select QR code

Use the QR code for dynamic status sheets.

For details on how to print dynamic status sheets, refer to the Technical Reference Guide of each model.

- ❑ For models that cannot automatically print Printer Easy Select QR code

Create QR code using [createQR \(p.142\)](#).

Refer to QR code creation in the sample program.

## Print Document Creation

Create a print document using the [EposBuilder class \(p.49\)](#).

Create an EposBuilder class using the constructor for it and create a print document using APIs of the EposBuilder class. Use the programming example below for your reference.

```
//Initialize an EposBuilder class instance
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
             Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    //Create a print document
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextLang: EPOS_OC_LANG_EN];
    errorStatus = [builder addTextSmooth: EPOS_OC_TRUE];
    errorStatus = [builder addTextFont: EPOS_OC_FONT_A];
    errorStatus = [builder addTextSize: 3 Height: 3];
    errorStatus = [builder addText: @"Hello,\t"];
    errorStatus = [builder addText: @"World!\n"];
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
    [builder release];
}
```

### To create a text print document:

To create a text print document, using APIs for text, store the font settings in command buffers to create a print document. Use the programming example below for your reference.



Make the language settings based on the language of the characters you are printing.  
For details, please refer to [addTextLang \(p.59\)](#).

For the string "Hello, World!", to create a print document based on the following settings:

- Font:      FontA
- Scale:    x 4 (horizontal) and x 4 (vertical)
- Style:    Bold

```
//Initialize an EposBuilder class instance
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
             Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    //Create a print document
    int errorStatus = EPOS_OC_SUCCESS;
    //<Configure the print character settings>
    errorStatus = [builder addTextLang: EPOS_OC_LANG_EN];
    errorStatus = [builder addTextSmooth: EPOS_OC_TRUE];
    errorStatus = [builder addTextFont: EPOS_OC_FONT_A];
    errorStatus = [builder addTextSize: 4 Height: 4];
    errorStatus = [builder addTextStyle: EPOS_OC_FALSE U1: EPOS_OC_FALSE
                                     Em: EPOS_OC_TRUE Color: EPOS_OC_PARAM_UNSPECIFIED];
    //<Specify the print data>
    errorStatus = [builder addText: @"Hello,\t"];
    errorStatus = [builder addText: @"World!\n"];
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
}
```

---

**To create a graphic print document:**

To create a graphic print document, for graphics, store the UIImage class in the command buffers with [addImage \(p.69\)](#) of the EposBuilder class. Use the programming example below for your reference.

```
UIImage * imageData = [UIImage imageNamed:@"Sample.png"];

//Initialize an EposBuilder class instance
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    //Create a print document
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addImage: imageData X: 0 Y: 0 Width: 256 Height: 256
                          Color: EPOS_OC_PARAM_DEFAULT];
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
}
```



For ways of graphic printing, you can also print the graphics registered in the printer's NV memory. For details, please refer to [addLogo \(p.77\)](#).

## To create a page mode print document

The page mode starts by storing [addPageBegin \(p.89\)](#) of the EposBuilder class into a command buffer. Store the print area ([addPageArea \(p.91\)](#)) and the print start position ([addPagePosition \(p.94\)](#)) in command buffers.

Specify the print start position according to the print data. Then, store APIs in command buffers and create print data. For the page mode end, store [addPageEnd \(p.90\)](#) in a command buffer. Use the programming example below for your reference.



Make the language settings based on the language of the characters you are printing.  
For details, please refer to [addTextLang \(p.59\)](#).

**For the string "Hello, World!", to create a print document based on the following settings:**

- Page mode print area (in dots):  
Origin of horizontal axis: 100, origin of vertical axis: 50, width: 200, height: 100
- Page mode print positions (in dots):  
Horizontal print position: 0, vertical print position: 42
- Font: FontA
- Scale: x 2 (horizontal) and x 2 (vertical)
- Style: Bold

```
//Initialize an EposBuilder class instance
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    //Create a print document
    //<The page mode starts>
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPageArea: 100 Y: 50 Width: 200 Height: 100];
    errorStatus = [builder addPagePosition: 0 Y: 42];
    //<Configure the print character settings>
    errorStatus = [builder addTextLang: EPOS_OC_LANG_EN];
    errorStatus = [builder addTextSmooth: EPOS_OC_TRUE];
    errorStatus = [builder addTextFont: EPOS_OC_FONT_A];
    errorStatus = [builder addTextSize: 2 Height: 2];
    errorStatus = [builder addTextStyle: EPOS_OC_FALSE U1: EPOS_OC_FALSE
                        Em: EPOS_OC_TRUE Color: EPOS_OC_PARAM_UNSPECIFIED];
    //<Specify the print data>
    errorStatus = [builder addText: @"Hello,\t"];
    errorStatus = [builder addText: @"World!\n"];
    //<The page mode ends>
    errorStatus = [builder addPageEnd];
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
}
```

## Transmission of Print Document

Send a print document using the [EposPrint class \(p.51\)](#). Create an EposPrint class using the constructor for it, use sendData to specify the EposBuilder class instance that stores the command buffers for the print document, and send the document.

The command buffers stored in the EposBuilder class will be retained until [clearCommandBuffer \(p.54\)](#) is executed. Execute clearCommandBuffer after the success of [sendData \(p.118\)](#).



If you want to print the same document repeatedly, you don't have to execute clearComanndBuffer.

Use the programming example below for your reference.

```
//Initialize an EposBuilder class instance
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V" Lang: EPOS_OC_MODEL_ANK];
if (builder != nil) {
    int errorStatus = EPOS_OC_SUCCESS;
    //Create a print document
    //The page mode starts>
    errorStatus = [builder addTextLang: EPOS_OC_LANG_EN];
    errorStatus = [builder addTextSmooth: EPOS_OC_TRUE];
    errorStatus = [builder addTextFont: EPOS_OC_FONT_A];
    errorStatus = [builder addTextSize: 4 Height: 4];
    errorStatus = [builder addTextStyle: EPOS_OC_FALSE Ul: EPOS_OC_FALSE
                                Em: EPOS_OC_TRUE Color: EPOS_OC_PARAM_UNSPECIFIED];
    //Specify the print data>
    errorStatus = [builder addText: @"Hello,\t"];
    errorStatus = [builder addText: @"World!\n"];
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
    //Initialize an EposPrint class instance
    id printer = [[EposPrint alloc] init];
    long status;
    //Send a print document
    if (printer != nil) {
        //Start communication with the printer>
        errorStatus = [printer openPrinter:deviceList.deviceType
                                DeviceName:deviceList.deviceName Enabled:EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT Timeout:EPOS_OC_PARAM_DEFAULT];

        //Send data>
        errorStatus = [printer sendData:builder Timeout:10000 Status:&status];
        //Delete the command buffers>
        if ((status & EPOS_OC_ST_PRINT_SUCCESS) == EPOS_OC_ST_PRINT_SUCCESS) {
            errorStatus = [builder clearCommandBuffer];
        }
        //End communication with the printer>
        errorStatus = [printer closePrinter];
        [printer release];
    }
    [builder release];
}
```

---

### ***Effective range of command buffers for setting***

The effective range of addXXX in the EposBuilder class instance used for setting is from the time when addXXX is set until sendData is executed. The set value is initialized each time sendData is executed. Refer to the following:

*Example:*

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V" Lang: EPOS_OC_MODEL_ANK];
errorStatus = [builder addText: @"Hello, World!\n"]; String for which the addTextFont setting is disabled
errorStatus = [builder addTextFont: EPOS_OC_FONT_A];
errorStatus = [builder addText: @"Hello, World!\n"]; String for which the addTextFont setting is enabled (FONT_A)
errorStatus = [printer sendData:builder Timeout:10000 Status:&status];
errorStatus = [builder addText: @"Hello, World!\n"]; String for which the addTextFont setting is disabled
errorStatus = [builder addTextFont: EPOS_OC_FONT_B];
errorStatus = [builder addText: @"Hello, World!\n"]; String for which the addTextFont setting is enabled (FONT_B)
errorStatus = [printer sendData:builder Timeout:10000 Status:&status];
```



## Printing After Checking the Printer Status

To ensure successful print operation, print after checking the printer status.

If the empty print data is transmitted and the printer is online, it will be printed.

Use the programming example below for your reference.

```
//Initialize an EposBuilder class instance
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V" Lang: EPOS_OC_MODEL_ANK];
if (builder != nil) {
    int errorStatus = EPOS_OC_SUCCESS;
    //Create a print document
    errorStatus = [builder addText: @"Hello,\t"];
    errorStatus = [builder addText: @"World!\n"];
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
}
(1)

//Initialize an EposBuilder class instance for confirmation
id conBuilder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
(2)

//Initialize an EposPrint class instance
id printer = [[EposPrint alloc] init];
long status;
if (printer != nil) {
    //<Start communication with the printer>
    errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP DeviceName:@"192.168.192.168"];


    //<Send data for confirmation>
    errorStatus = [printer sendData:conBuilder Timeout:10000 Status:&status];
    (3)

    if (errorStatus == EPOS_OC_SUCCESS && (status & EPOS_OC_ST_OFF_LINE) != EPOS_OC_ST_OFF_LINE) {
        //<Send print data>
        errorStatus = [printer sendData:builder Timeout:10000 Status:&status];
        (4)
    }
    else if ((status & EPOS_OC_ST_OFF_LINE) != EPOS_OC_ST_OFF_LINE) {
        (5)
    }
    else if (errorStatus == EPOS_OC_ERR_CONNECT) {
        //<End communication with the printer>
        errorStatus = [printer closePrinter];
        (6)

        //<Send data for confirmation>
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP DeviceName:@"192.168.192.168"];

        //<Send print data>
        errorStatus = [printer sendData:builder Timeout:10000 Status:&status];
    }
    else {
        ;
    }

    //<End communication with the printer>
    errorStatus = [printer closePrinter];
    [printer release];
}
[conBuilder release];
[builder release];
}
```

- 
- 
- 1** Create print data.
  - 2** To check the printer status, send empty print data.
  - 3** Send the print data created in (2).
  - 4** If the print data created in (2) was properly sent, and the printer is online, then send the print data created in (1).
  - 5** If the print data created in (2) was properly sent, and the printer is offline, eliminate the cause that makes the printer offline.  
(Such as cover open and no paper.)
  - 6** If the print data created in (2) was not properly sent, stop communicating with the printer, restart communicating with the printer again, and then send the print data.

# Automatic Acquisition of Printer Status

In the ePOS-Print SDK, the printer status can be automatically notified to the application by means of callback. Refer to the following.

```
//Implementation of callback method for giving notification of printer status
- (void)onStatusChange:(NSString *)deviceName Status:(NSNumber *)status
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        //Registration of the printer status notification destination callback method
        [printer setStatusChangeEventCallback @selector(onStatusChange:Status:)
                                             Target:self];

        //Start communications with the printer and monitoring of the printer status
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                     Name:@"192.168.192.168"
                                     Enabled: EPOS_OC_TRUE
                                     Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

(1)/(4)

(2)

(3)

- 1 Implement the notification destination callback method when events occur.



In the above description, the callback method, which notifies the printer status at the intervals specified in [openPrinter \(p.110\)](#), is defined.  
ePOS-Print has callback method according to each printer status, for example, events such as cover open and drawer open. Use these according to the desired purpose of use. See the [Event List \(p.44\)](#) for the callback method that can be used with ePOS-Print.

- 2 Register the printer status notification destination.
- 3 Use [openPrinter \(p.110\)](#) to start monitoring of the printer status.
- 4 Notify the printer status to the event implemented in (1).



When printer status notification is ended, it ends on the [closePrinter \(p.117\)](#) of the EposPrint class.

## Event List



For details on the callback method, refer to [API Reference \(p.49\)](#), which explains the callback method registration API.

| Event                       | Callback method registration API                            |
|-----------------------------|---|
| Printer status notification | <a href="#">setStatusChangeEventCallback (p.122)</a>        |
| Online notification         | <a href="#">setOnlineEventCallback (p.123)</a>              |
| Offline notification        | <a href="#">setOfflineEventCallback (p.124)</a>             |
| Power off notification      | <a href="#">setPowerOffEventCallback (p.125)</a>            |
| Cover close notification    | <a href="#">setCoverOkEventCallback (p.126)</a>             |
| Cover open notification     | <a href="#">setCoverOpenEventCallback (p.127)</a>           |
| Paper OK notification       | <a href="#">setPaperOkEventCallback (p.128)</a>             |
| Paper near end notification | <a href="#">setPaperNearEndEventCallback (p.129)</a>        |
| Paper end notification      | <a href="#">setPaperEndEventCallback (p.130)</a>            |
| Drawer close notification   | <a href="#">setDrawerClosedEventCallback (p.131)</a>        |
| Drawer open notification    | <a href="#">setDrawerOpenEventCallback (p.132)</a>          |
| Battery low notification    | <a href="#">setBatteryLowEventCallback (p.133)</a>          |
| Battery OK notification     | <a href="#">setBatteryOkEventCallback (p.134)</a>           |
| Battery status notification | <a href="#">setBatteryStatusChangeEventCallback (p.135)</a> |

# Status

The following statuses are defined in ePOS-Print SDK for iOS.

| Type           | Description  |
|----------------|--|
| Error status   | These are the return values when the API for each class is executed.<br>For details, refer to <a href="#">Error Statuses and Actions to Take (p.45)</a> .  |
| Printer status | Status of the printer when print data was sent.<br>The printer status can be acquired only when <a href="#">sendData (p.118)</a> is executed.<br>For details, refer to <a href="#">Printer Statuses and Actions to Take (p.47)</a> . |
| Battery status | Status of the printer's remaining battery power.<br>For details, refer to <a href="#">Battery Status (p.48)</a> .  |

## Error Statuses and Actions to Take

| Error status        | Cause  | Action to Take  |
|---------------------|--|---|
| EPOS_OC_SUCCESS     | Processing was successful.   | -   |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.<br><Example> <ul style="list-style-type: none"> <li>An invalid parameter such as null was passed.</li> <li>A value outside the supported range was specified.</li> </ul> | The parameter was specified incorrectly.<br>Check the parameter.  |
| EPOS_OC_ERR_OPEN    | Open processing failed.<br><Example><br>Could not connect to the designated printer.   | Check the iOS device and the printer.<br>(The printer's power condition, communication condition, etc.)   |
| EPOS_OC_ERR_CONNECT | Failed to connect to device.<br><Example><br>Failed to send the data to the printer.   | Execute <code>closePrinter</code> and then <code>openPrinter</code> to restore the communication with the device.<br>When <i>Bluetooth</i> is selected for the interface, to execute <code>openPrinter</code> , the iOS device and the printer need to be paired. |
| EPOS_OC_ERR_TIMEOUT | The specified timeout time was exceeded.<br><Example><br>Could not send all the data within the specified time.  | Check the timeout time.<br>Set a value for the timeout time longer than the time required for printing.   |
| EPOS_OC_ERR_MEMORY  | Could not allocate the necessary memory for processing.  | End the unneeded applications.  |
| EPOS_OC_ERR_ILLEGAL | Illegal method used.<br><Example><br>When the printer was not opened, an API for sending a command to the printer was called.  | Use the API in a proper way.<br>Refer to <a href="#">Programming Flow (p.33)</a> .  |

| Error status            | Cause  | Action to Take  |
|-------------------------|--|---|
| EPOS_OC_ERR_PROCESSING  | Could not execute process.<br><Example><br>Could not execute the process because an identical process is being executed in another thread. | Review the application processing timing so that processes do not overlap each other.   |
| EPOS_OC_ERR_UNSUPPORTED | An unsupported model name or language specification was specified.   | Cannot be used for unsupported models.  |
| EPOS_OC_ERR_OFF_LINE    | The printer is offline.  | Eliminate the cause that makes the printer offline.<br>(Such as cover open and no paper.)   |
| EPOS_OC_ERR_FAILURE     | An unspecified error occurred.   | <ul style="list-style-type: none"> <li>• Check the communication settings of the iOS device. (Wi-Fi connection setting, <i>Bluetooth</i> connection setting, etc.)</li> <li>• Check that there is no problem with the execution environment.</li> </ul> |

## Printer Statuses and Actions to Take

| Printer Status   | Cause  | Action to Take  |
|--|--|---|
| EPOS_OC_ST_NO_RESPONSE<br>(0x00000001)                             | <ul style="list-style-type: none"> <li>The power to the printer is not turned ON.</li> <li>Communication is not established.</li> <li>The communication cable is not connected.</li> </ul> | Check the printer status including the power condition and cable, and the communication status.       |
| EPOS_OC_ST_PRINT_SUCCESS<br>(0x00000002)                           | Printing is successfully completed   | -   |
| <Other than TM-P Series><br>EPOS_OC_ST_DRAWER_KICK<br>(0x00000004) | Status of the 3rd pin of the drawer kick-out connector = "H"   | -   |
| <TM-P Series><br>EPOS_OC_ST_BATTERY_OFFLINE<br>(0x00000004)        | Battery offline status   | Charge the battery.   |
| EPOS_OC_ST_OFF_LINE<br>(0x00000008)                                | Offline  | Eliminate the cause that makes the printer offline. (Such as cover open and no paper.)                |
| EPOS_OC_ST_COVER_OPEN<br>(0x00000020)                              | The cover is open  | Close the printer's cover.  |
| EPOS_OC_ST_PAPER_FEED<br>(0x00000040)                              | Paper is being fed by a paper feed switch operation  | -   |
| EPOS_OC_ST_PANEL_SWITCH<br>(0x00000200)                            | The paper feed switch is being pressed (ON)  | -   |
| EPOS_OC_ST_MECHANICAL_ERR<br>(0x00000400)                          | A mechanical error occurred  | Eliminate the cause of the error and turn the printer on again.                                       |
| EPOS_OC_ST_AUTOCUTTER_ERR<br>(0x00000800)                          | An autocutter error occurred   | Turn the printer off immediately.   |
| EPOS_OC_ST_UNRECOVER_ERR<br>(0x00002000)                           | An unrecoverable error occurred  | Turn the printer off immediately.   |
| EPOS_OC_ST_AUTORECOVER_ERR<br>(0x00004000)                         | An automatically recoverable error occurred  | The error status is automatically canceled when the temperature of the head drops as the time passes. |
| EPOS_OC_ST_RECEIPT_NEAR_END<br>(0x00020000)                        | No paper in roll paper near end sensor   | Feed paper into the printer.  |
| EPOS_OC_ST_RECEIPT_END<br>(0x00080000)                             | No paper in roll paper end sensor  | Feed paper into the printer.  |
| EPOS_OC_ST_BUZZER<br>(0x01000000)                                  | A buzzer is on (only for applicable devices)   | -   |
|  | Waiting for label to be removed (only for applicable devices)  | Remove the label.   |

## Battery Status

The battery status consists of the following 16 bits (0x0000).

| Bit          | Description   |
|--------------|---|
| Upper 8 bits | Common battery status<br>For details, refer to <a href="#">Common battery status (upper 8 bits) (p.48)</a> .        |
| Lower 8 bits | Battery status exclusive by model<br>For details, refer to <a href="#">Support Information by Printer (p.164)</a> . |



"0x0000" is returned if the battery status cannot be acquired or if the model does not support the battery status.

### *Common battery status (upper 8 bits)*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |



# API Reference

This chapter describes the APIs provided in the ePOS-Print SDK for iOS.

## ePOS-Print API

The ePOS-Print APIs are APIs for creating and printing print documents. The following classes are available.

- ❑ EposBuilder class ([p. 49](#))
- ❑ EposPrint class ([p. 51](#))



The APIs that you can use and the settings that you can designate vary based on the printer. For details, refer to [List of Supported APIs for Each Printer Model \(p.163\)](#) and [Support Information by Printer \(p.164\)](#).

### ***EposBuilder class***

This class creates print documents for printer control commands such as character strings to print, graphic printing, and paper cutting. The following APIs are available.

| API                      |                    | Description  | Page                   |
|--------------------------|--------------------|--|------------------------|
| initWithPrinterModel     |                    | Initialize an EposBuilder class instance.                | <a href="#">p. 52</a>  |
| Clearing command buffers | clearCommandBuffer | Clears the command buffers added by APIs.                | <a href="#">p. 54</a>  |
| Text                     | addTextAlign       | Adds a tag for the text alignment setting.               | <a href="#">p. 55</a>  |
|                          | addTextLineSpace   | Adds a tag for the line feed space setting.              | <a href="#">p. 56</a>  |
|                          | addTextRotate      | Adds a tag for the text rotation setting.                | <a href="#">p. 57</a>  |
|                          | addText            | Adds a tag for printing text.                            | <a href="#">p. 58</a>  |
|                          | addTextLang        | Adds a tag for the target language setting.              | <a href="#">p. 59</a>  |
|                          | addTextFont        | Adds a tag for the text font setting.                    | <a href="#">p. 60</a>  |
|                          | addTextSmooth      | Adds a tag for the text smoothing setting.               | <a href="#">p. 61</a>  |
|                          | addTextDouble      | Adds a tag for specifying the double-sized text setting. | <a href="#">p. 62</a>  |
|                          | addTextSize        | Adds a tag for the text scale setting.                   | <a href="#">p. 63</a>  |
|                          | addTextStyle       | Adds a tag for the text style setting.                   | <a href="#">p. 64</a>  |
| Paper Feed               | addTextPosition    | Adds a tag for specifying the print position of text.    | <a href="#">p. 66</a>  |
|                          | addFeedUnit        | Adds a tag for paper feeding (in dots).                  | <a href="#">p. 67</a>  |
|                          | addFeedLine        | Adds a tag for paper feeding (in lines).                 | <a href="#">p. 68</a>  |
|                          | addFeedPosition    | Adds a tag for label / black mark paper feeding.         | <a href="#">p. 105</a> |

| API             |                               | Description   | Page                   |
|-----------------|-------------------------------|---|------------------------|
| Graphic         | addImage                      | Adds multiple tone raster image printing to the command buffer.<br>Compresses image data and adds them to the command buffer. ( <i>Bluetooth</i> interface) | <a href="#">p. 69</a>  |
|                 | addImage<br>(Previous format) | Adds multiple tone raster image printing to the command buffer.<br>(Image data compression cannot be used ( <i>Bluetooth</i> interface).)                   | <a href="#">p. 72</a>  |
|                 | addImage<br>(Previous format) | Adds a tag for a raster image to be printed.<br>(Image data compression cannot be used ( <i>Bluetooth</i> interface).<br>Multiple tones cannot be printed.) | <a href="#">p. 75</a>  |
|                 | addLogo                       | Adds a tag for an NV logo to be printed.  | <a href="#">p. 77</a>  |
| Barcode         | addBarcode                    | Adds a tag for a barcode to be printed.   | <a href="#">p. 78</a>  |
|                 | addSymbol                     | Adds a tag for a 2D-Code to be printed.   | <a href="#">p. 84</a>  |
| Pagemode        | addPageBegin                  | Adds a tag for switching to page mode.  | <a href="#">p. 89</a>  |
|                 | addPageEnd                    | Adds a tag for finishing page mode.   | <a href="#">p. 90</a>  |
|                 | addPageArea                   | Adds a tag for specifying the print area in page mode.  | <a href="#">p. 91</a>  |
|                 | addPageDirection              | Adds a tag for specifying the print direction in page mode.   | <a href="#">p. 92</a>  |
|                 | addPagePosition               | Adds a tag for specifying the print position in page mode.  | <a href="#">p. 94</a>  |
|                 | addPageLine                   | Adds a tag for drawing a line in page mode.   | <a href="#">p. 95</a>  |
|                 | addPageRectangle              | Adds a tag for drawing a rectangle in page.   | <a href="#">p. 97</a>  |
| Cut             | addCut                        | Adds a tag for paper cut.   | <a href="#">p. 99</a>  |
| Drawer kick-out | addPulse                      | Adds a tag for the drawer kick-out.   | <a href="#">p. 100</a> |
| Buzzer          | addSound                      | Adds a tag for turning on the buzzer.   | <a href="#">p. 101</a> |
|                 | addSound<br>(Previous format) | Adds a tag for turning on the buzzer.<br>(The buzzer sounding cycle cannot be set.)   | <a href="#">p. 103</a> |
| Paper Layout    | addLayout                     | Adds a tag for paper layout information.  | <a href="#">p. 106</a> |
| Send Command    | addCommand                    | Adds a tag for inserting commands.  | <a href="#">p. 108</a> |

**EposPrint class**

Controls the printer by sending a print document created using the EposBuilder class, and monitors the transmission result and the communication status.

| API                                 | Description  | Page                   |
|-------------------------------------|--|------------------------|
| init                                | Initialize an EposPrint class instance.  | <a href="#">p. 109</a> |
| openPrinter                         | Start communication with the printer.  | <a href="#">p. 112</a> |
| openPrinter(Previous format)        | Start communication with the printer.<br>(Timeout cannot be set.)                                    | <a href="#">p. 112</a> |
| openPrinter(Previous format)        | Start communication with the printer.<br>(The printer status acquisition and timeout cannot be set.) | <a href="#">p. 115</a> |
| closePrinter                        | End communication with the printer.  | <a href="#">p. 117</a> |
| sendData                            | Sends a command to the printer.  | <a href="#">p. 118</a> |
| sendData(Previous format)           | Sends a command to the printer.<br>(The battery status cannot be acquired.)                          | <a href="#">p. 120</a> |
| setStatusChangeEventCallback        | Registers the printer status callback method.  | <a href="#">p. 122</a> |
| setOnlineEventCallback              | Registers the online event callback method.  | <a href="#">p. 123</a> |
| setOfflineEventCallback             | Registers the offline event callback method.   | <a href="#">p. 124</a> |
| setPowerOffEventCallback            | Registers the power off event callback method.   | <a href="#">p. 125</a> |
| setCoverOkEventCallback             | Registers the cover close event callback method.   | <a href="#">p. 126</a> |
| setCoverOpenEventCallback           | Registers the cover open event callback method.  | <a href="#">p. 127</a> |
| setPaperOkEventCallback             | Registers the paper OK event callback method.  | <a href="#">p. 128</a> |
| setPaperNearEndEventCallback        | Registers the paper near end event callback method.  | <a href="#">p. 129</a> |
| setPaperEndEventCallback            | Registers the paper end event callback method.   | <a href="#">p. 130</a> |
| setDrawerClosedEventCallback        | Registers the drawer close event callback method.  | <a href="#">p. 131</a> |
| setDrawerOpenEventCallback          | Registers the drawer open event callback method.   | <a href="#">p. 132</a> |
| setBatteryLowEventCallback          | Registers the battery low event notification destination   | <a href="#">p. 133</a> |
| setBatteryOkEventCallback           | Registers the battery OK event notification destination  | <a href="#">p. 134</a> |
| setBatteryStatusChangeEventCallback | Registers the battery status callback method.  | <a href="#">p. 135</a> |

## initWithPrinterModel

Initializes an EposBuilder class instance.

### Syntax

```
- (id) initWithPrinterModel: (NSString *)printerModel  
    Lang: (int) lang;
```

### Parameter

- printerModel : Specifies the model name for the target printer.

| Set value  | Description   |
|------------|---|
| "TM-P20"   | <ul style="list-style-type: none"><li>• TM-P20 Wi-Fi</li><li>• TM-P20 iOS <i>Bluetooth</i> model</li></ul>  |
| "TM-P60"   | <ul style="list-style-type: none"><li>• TM-P60(Receipt) Wi-Fi</li><li>• TM-P60(Receipt) iOS <i>Bluetooth</i> model</li><li>• TM-P60(Peeler) Wi-Fi</li><li>• TM-P60(Peeler) iOS <i>Bluetooth</i> model</li></ul>         |
| "TM-P60II" | <ul style="list-style-type: none"><li>• TM-P60II(Receipt) Wi-Fi</li><li>• TM-P60II(Receipt) iOS <i>Bluetooth</i> model</li><li>• TM-P60II(Peeler) Wi-Fi</li><li>• TM-P60II(Peeler) iOS <i>Bluetooth</i> model</li></ul> |
| "TM-P80"   | <ul style="list-style-type: none"><li>• TM-P80 Wi-Fi</li><li>• TM-P80 iOS <i>Bluetooth</i> model</li></ul>  |
| "TM-T20"   | TM-T20  |
| "TM-T20II" | <ul style="list-style-type: none"><li>• TM-T20II</li><li>• TM-T20II iOS <i>Bluetooth</i> model</li></ul>  |
| "TM-T70"   | TM-T70  |
| "TM-T70II" | <ul style="list-style-type: none"><li>• TM-T70II</li><li>• TM-T70II iOS <i>Bluetooth</i> model</li></ul>  |
| "TM-T81II" | TM-T81II  |
| "TM-T82"   | TM-T82  |
| "TM-T82II" | TM-T82II  |
| "TM-T88V"  | <ul style="list-style-type: none"><li>• TM-T88V</li><li>• TM-T88V iOS <i>Bluetooth</i> model</li></ul>  |
| "TM-T90II" | TM-T90II  |
| "TM-U220"  | TM-U220   |
| "TM-U330"  | TM-U330   |

- lang : Specifies the language specifications for the printer.

| Set value               | Printer model       | TM printer-separate setting |        |          |        |        |          |        |          |          |        |          |         |          |         |         |
|-------------------------|---------------------|-----------------------------|--------|----------|--------|--------|----------|--------|----------|----------|--------|----------|---------|----------|---------|---------|
|                         |                     | TM-P20                      | TM-P60 | TM-P60II | TM-P80 | TM-T20 | TM-T20II | TM-T70 | TM-T70II | TM-T81II | TM-T82 | TM-T82II | TM-T88V | TM-T90II | TM-U220 | TM-U330 |
| EPOS_OC_MODEL_ANK       | ANK                 | ✓                           | ✓      | ✓        | ✓      | ✓      | ✓        | ✓      | ✓        | -        | -      | ✓        | ✓       | -        | ✓       | -       |
| EPOS_OC_MODEL_JAPANESE  | Japanese            | ✓                           | -      | -        | -      | ✓      | -        | ✓      | ✓        | -        | -      | -        | ✓       | ✓        | ✓       | -       |
| EPOS_OC_MODEL_CHINESE   | Simplified Chinese  | ✓                           | -      | -        | -      | -      | -        | ✓      | ✓        | ✓        | -      | ✓        | ✓       | -        | ✓       | ✓       |
| EPOS_OC_MODEL_TAIWAN    | Traditional Chinese | ✓                           | -      | ✓        | ✓      | -      | -        | ✓      | ✓        | -        | -      | ✓        | ✓       | -        | ✓       | -       |
| EPOS_OC_MODEL_KOREAN    | Korean              | -                           | -      | -        | -      | -      | -        | -      | ✓        | -        | -      | -        | ✓       | -        | ✓       | -       |
| EPOS_OC_MODEL_THAI      | Thai                | -                           | -      | -        | -      | -      | -        | ✓      | ✓        | -        | ✓      | ✓        | ✓       | -        | ✓       | -       |
| EPOS_OC_MODEL_SOUTHASIA | South Asian         | ✓                           | -      | -        | -      | -      | -        | ✓      | ✓        | -        | ✓      | ✓        | ✓       | -        | ✓       | -       |

### Return Value

If processing succeeds, the initialized EposBuilder class instance is returned.

If processing failed, "nil" is returned. The following reasons can cause processing to fail.

- An invalid parameter was specified.
- Could not acquire the necessary memory
- An unsupported model name or language specification was specified.

### Example

If you are initializing the command buffer for the TM-T88V ANK model:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    ///Process///
    [builder release];
}
```

## clearCommandBuffer

Clears command buffers used by APIs of the EposBuilder class.

The command buffers stored in the EposBuilder class will be retained until this API is executed.

---

### Syntax

- (int) **clearCommandBuffer**;

### Return Value

| Error status    | Description                |
|-----------------|----------------------------|
| EPOS_OC_SUCCESS | Processing was successful. |

### Example

If you are clearing the command buffer:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus;
    ///Process///
    errorStatus = [builder clearCommandBuffer];
    ///Process///
    [builder release];
}
```

## addTextAlign

Adds the text alignment setting to the command buffer.



- This API setting also applies to barcodes/2D-Code.
- When the page mode is selected, use [addPagePosition \(p.94\)](#) instead of this API to set the alignment.

### Syntax

- (int) **addTextAlign**: (int)align;

### Parameter

- align : Specifies the text alignment.

| Set value                    | Description             |
|------------------------------|-------------------------|
| EPOS_OC_ALIGN_LEFT (default) | Alignment to the left   |
| EPOS_OC_ALIGN_CENTER         | Alignment to the center |
| EPOS_OC_ALIGN_RIGHT          | Alignment to the right  |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set alignment to the center:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextAlign: EPOS_OC_ALIGN_CENTER];
    ///Process///
}
```

## addTextLineSpace

Adds the line feed space setting to the command buffer.

### Syntax

```
- (int) addTextLineSpace: (long) linespc;
```

### Parameter

- linespc : Specifies the line feed space (in dots). Specifies an integer from 0 to 255. (Default value: Refer to [Support Information by Printer \(p.164\)](#).)

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set the line feed space to 50 dots:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextLineSpace: 50];
    ///Process///
}
```



## addTextRotate

Adds the text rotation setting to the command buffer.



- This API setting also applies to barcodes/two dimensional symbols.
- When the page mode is selected, to set text rotation, use the [addPageDirection \(p.92\)](#) instead of this API function.

### Syntax

```
- (int) addTextRotate: (int) rotate;
```

### Parameter

- rotate : Specifies whether to rotate text.

| Set value               | Description                         |
|-------------------------|-------------------------------------|
| EPOS_OC_TRUE            | Specifies rotated printing of text. |
| EPOS_OC_FALSE (default) | Cancels rotated printing of text.   |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set text rotation:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextRotate: EPOS_OC_TRUE];
    ///Process///
}
```

## addText

Adds the printing of text to the command buffer.



After printing text, to print content other than text, execute line feed or paper feed.  
(Example: After printing text, an attempt was made to perform graphic printing, but nothing was printed.)

### Syntax

```
- (int) addText: (NSString *)data;
```

### Parameter

- data : Specify a character string to be printed.  
For the horizontal tab/line feed, use the following escape sequences:

| String | Description        |
|--------|--------------------|
| \t     | Horizontal tab(HT) |
| \n     | Line feed (LF)     |
| \\     | Carriage return    |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To add character strings:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addText: @"Hello,\t"];
    errorStatus = [builder addText: @"World!\n"];
    ///Process///
}
```

## addTextLang

Adds the language setting to a command buffer. Encodes the string specified by [addText \(p.58\)](#) according to the language information specified by this API. Specify the value according to the language specifications set for [initWithPrinterModel \(p.52\)](#).



This API is an API to be called before calling [addText \(p.58\)](#).

### Syntax

```
- (int) addTextLang: (int) lang;
```

### Parameter

- lang : Specifies the target language.

| Set value                | Language                               |
|--------------------------|--|
| EPOS_OC_LANG_EN(default) | English(ANK)                           |
| EPOS_OC_LANG_JA          | Japanese                               |
| EPOS_OC_LANG_ZH_CN       | Simplified Chinese                     |
| EPOS_OC_LANG_ZH_TW       | Traditional Chinese                    |
| EPOS_OC_LANG_KO          | Korean                                 |
| EPOS_OC_LANG_TH          | Thai (South Asia specifications)       |
| EPOS_OC_LANG_VI          | Vietnamese (South Asia specifications) |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set the language as English:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextLang: EPOS_OC_LANG_EN];
    ///Process///
}
```

## addTextFont

Adds the text font setting to the command buffer.

### Syntax

- (int) **addTextFont**: (int) font;

### Parameter

- font : Specifies the font.

| Set value                   | Description | TM printer-separate setting |                 |        |                 |                 |          |                 |         |          |         |         |
|-----------------------------|-------------|-----------------------------|-----------------|--------|-----------------|-----------------|----------|-----------------|---------|----------|---------|---------|
|                             |             | TM-P20                      | TM-P60/TM-P60II | TM-P80 | TM-T20/TM-T20II | TM-T70/TM-T70II | TM-T81II | TM-T82/TM-T82II | TM-T88V | TM-T90II | TM-U220 | TM-U330 |
| EPOS_OC_FONT_A<br>(default) | Font A      | ✓                           | ✓               | ✓      | ✓               | ✓               | ✓        | ✓               | ✓       | ✓        | ✓       | ✓       |
| EPOS_OC_FONT_B              | Font B      | ✓                           | ✓               | ✓      | ✓               | ✓               | ✓        | ✓               | ✓       | ✓        | ✓       | ✓       |
| EPOS_OC_FONT_C              | Font C      | ✓                           | ✓               | -      | -               | -               | -        | -               | -       | ✓        | -       | -       |
| EPOS_OC_FONT_D              | Font D      | ✓                           | -               | -      | -               | -               | -        | -               | -       | -        | -       | -       |
| EPOS_OC_FONT_E              | Font E      | ✓                           | -               | -      | -               | -               | -        | -               | -       | -        | -       | -       |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set the font B:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextFont: EPOS_OC_FONT_B];
    ///Process///
}
```

## addTextSmooth

Adds the smoothing setting to the command buffer.

### Syntax

```
- (int) addTextSmooth: (int) smooth;
```

### Parameter

- **smooth** : Specifies whether to enable smoothing.

| Set value               | Description          |
|-------------------------|----------------------|
| EPOS_OC_TRUE            | Specifies smoothing. |
| EPOS_OC_FALSE (default) | Cancels smoothing    |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To enable smoothing:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextSmooth: EPOS_OC_TRUE];
    ///Process///
}
```

## addTextDouble

Adds the double-sized text setting to the command buffer.

### Syntax

- (int) **addTextDouble**: (int) dw Dh: (int) dh;

### Parameter

- dw : Specifies the double-sized width.

| Set value                 | Description                       |
|---------------------------|-----------------------------------|
| EPOS_OC_TRUE              | Specifies the double-sized width. |
| EPOS_OC_FALSE (default)   | Cancels the double-sized width    |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.      |

- dh : Specifies the double-sized height.

| Set value                 | Description                       |
|---------------------------|-----------------------------------|
| EPOS_OC_TRUE              | Specifies the double-sized height |
| EPOS_OC_FALSE (default)   | Cancels the double-sized height   |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.      |



When EPOS\_OC\_TRUE or 1 is set for both the dw and dh parameters, double width and height characters are printed.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set the size as double width and height:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextDouble: EPOS_OC_TRUE
                  Dh: EPOS_OC_TRUE];
    ///Process///
}
```

## addTextSize

Adds the text scale setting to the command buffer.

### Syntax

```
- (int) addTextSize: (long)width Height: (long)height;
```

### Parameter

- width : Specifies the horizontal scale of text.

| Set value                 | Description                    |
|---------------------------|--------------------------------|
| Integer from 1 to 8       | Horizontal scale (default : 1) |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.   |

- height : Specifies the vertical scale of text.

| Set value                 | Description                  |
|---------------------------|------------------------------|
| Integer from 1 to 8       | Vertical scale (default : 1) |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting. |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set a horizontal scale of x 4 and a vertical scale of x 4:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextSize: 4 Height: 4];
    ///Process///
}
```

## addTextStyle

Adds the text style setting to the command buffer.

### Syntax

```
- (int) addTextStyle: (int)reverse Ul: (int)ul Em: (int)em  
Color: (int)color;
```

### Parameter

- reverse : Specifies inversion of black and white for text.

| Set value                 | Description   |
|---------------------------|---|
| EPOS_OC_TRUE              | Specifies the inversion of black and white parts of characters. |
| EPOS_OC_FALSE (default)   | Cancels the inversion of black and white parts of characters.   |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.                                    |

- ul : Specifies the underline style.

| Set value                 | Description                  |
|---------------------------|------------------------------|
| EPOS_OC_TRUE              | Specifies underlining.       |
| EPOS_OC_FALSE (default)   | Cancels underlining.         |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting. |

- em : Specifies the bold style.

| Set value                 | Description                                  |
|---------------------------|--|
| EPOS_OC_TRUE              | Specifies emphasized printing of characters. |
| EPOS_OC_FALSE (default)   | Cancels emphasized printing of characters.   |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.                 |

- color : Specifies the color.

| Set value                 | Description                       |
|---------------------------|-----------------------------------|
| EPOS_OC_COLOR_NONE        | Characters are not printed.       |
| EPOS_OC_COLOR_1 (default) | First color                       |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current color setting |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |



*Example*

To set the underline style:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextStyle: EPOS_OC_PARAM_UNSPECIFIED
        U1: EPOS_OC_TRUE Em: EPOS_OC_PARAM_UNSPECIFIED
        Color: EPOS_OC_PARAM_UNSPECIFIED];
    ///Process///
}
```

## addTextPosition

Adds the horizontal print start position of text to the command buffer.



After executing this API, you cannot use [addTextAlign \(p.55\)](#) or [addTextRotate \(p.57\)](#).

### Syntax

- (int) **addTextPosition:** (long) x;

### Parameter

- x: Specifies the horizontal print start position (in dots).  
Specifies an integer from 0 to 65535.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To set the print position at 120 dots from the left end:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addTextPosition: 120];
    ///Process///
}
```

## addFeedUnit

Adds paper feeding in dots to the command buffer.

### Syntax

```
- (int) addFeedUnit: (long)unit;
```

### Parameter

- **unit** : Specifies the paper feed space (in dots). Specifies an integer from 0 to 255.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To feed paper by 30 dots:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addFeedUnit: 30];
    ///Process///
}
```

## addFeedLine

Adds paper feeding in lines to the command buffer.

### Syntax

- (int) **addFeedLine**: (long) line;

### Parameter

- unit : Specifies the paper feed space (in lines). Specifies an integer from 0 to 255.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To feed paper by 3 lines:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addFeedLine: 3];
    ///Process///
}
```

## addImage

Adds raster image printing to the command buffer. Prints the graphic rendered by the UIImage class. Out of the UI Image class graphics, the specified scope is converted to raster image data according to this API setting.

One pixel in an image equals to one printer dot. When an image contains any transparent color, the background color of the image is assumed to be white.



- Set image compression only for a *Bluetooth* interface.
- To print a raster image at high speed, specify EPOS\_OC\_ALIGN\_LEFT for the [addTextAlign \(p.55\)](#), and specify a multiple of 8 not exceeding the printer's paper width for the width parameter of this API.
- Multiple tone printing is not supported in Page Mode. Multiple tone graphic printing is supported in Standard Mode only.
- Image compression is not supported in Page Mode.

### Syntax

```
(int) addImage: (UIImage *)data X:(long)x Y:(long)y
Width:(long)width Height:(long)height
Color:(int)color Mode:(int)mode
Halftone:(int)halftone
Brightness:(double)brightness;
Compress:(int)compress;
```

### Parameter

- data : Specifies an instance of the UIImage class.
- x : Specifies the horizontal start position in the print area. Specifies an integer from 0 to 65534.
- y : Specifies the vertical start position in the print area. Specifies an integer from 0 to 65534.
- width : Specifies the width of the print area. Specifies an integer from 1 to 65535.
- height : Specifies the height of the print area. Specifies an integer from 1 to 65535.



If the area defined in the x/y parameters and the width/height parameters do not fit in the image size defined by the data parameter, EPOS\_OC\_ERR\_PARAM will be returned for the return value.

- color : Specifies the color.

| Set value             | Description                 |
|-----------------------|-----------------------------|
| EPOS_OC_COLOR_NONE    | Characters are not printed. |
| EPOS_OC_COLOR_1       | First color                 |
| EPOS_OC_PARAM_DEFAULT | First color                 |

- mode : Specify the color mode.

| Set value             | Description   | TM printer-separate setting |                  |        |                  |        |          |          |                  |         |          |                 |
|-----------------------|---|-----------------------------|------------------|--------|------------------|--------|----------|----------|------------------|---------|----------|-----------------|
|                       |   | TM-P20                      | TM-P60/ TM-P60II | TM-P80 | TM-T20/ TM-T20II | TM-T70 | TM-T70II | TM-T81II | TM-T82/ TM-T82II | TM-T88V | TM-T90II | TM-U220/TM-U330 |
| EPOS_OC_MODE_MONO     | Monochrome (2 tone)   | ✓                           | ✓                | ✓      | ✓                | ✓      | ✓        | ✓        | ✓                | ✓       | ✓        | ✓               |
| EPOS_OC_MODE_GRAY16   | Multiple tone (16 tone)                                       | -                           | -                | -      | -                | -      | ✓        | -        | -                | ✓       | ✓        | -               |
| EPOS_OC_PARAM_DEFAULT | Specify the half tone treatment method. (Monochrome (2 tone)) | ✓                           | ✓                | ✓      | ✓                | ✓      | ✓        | ✓        | ✓                | ✓       | ✓        | ✓               |

- halftone : Specify the half tone treatment method.

| Set value                        | Description  |
|----------------------------------|--|
| EPOS_OC_HALFTONE_DITHER          | Dither<br>(This is suitable for graphic printing).                                   |
| EPOS_OC_HALFTONE_ERROR_DIFFUSION | Error diffusion<br>(This is suitable for mixed printing or characters and graphics). |
| EPOS_OC_HALFTONE_THRESHOLD       | Threshold value<br>(This is suitable for printing of characters).                    |
| EPOS_OC_PARAM_DEFAULT            | Default value (dither) selection   |



In the case of multiple tone (16 tone), this is disregarded.

- brightness : Specify the correction value for brightness.

| Set value                      | Description                               |
|--------------------------------|---|
| Actual figure from 0.1 to 10.0 | Brightness correction value (gamma value) |
| EPOS_OC_PARAM_DEFAULT          | Select the default value (1.0)            |



If you specify a value other than 1.0, the printing speed will become slower.

- **compress :** Specifies image compression. Specify EPOS\_OC\_COMPRESS\_DEFLATE only when *Bluetooth* is selected for the interface.

| Set value                | Description   | TM printer-separate setting |                  |        |        |          |        |          |           |                  |         |          |                 |
|--------------------------|---|-----------------------------|------------------|--------|--------|----------|--------|----------|-----------|------------------|---------|----------|-----------------|
|                          |   | TM-P20                      | TM-P60/ TM-P60II | TM-P80 | TM-T20 | TM-T20II | TM-T70 | TM-T70II | TM-T81 II | TM-T82/ TM-T82II | TM-T88V | TM-T90II | TM-U220/TM-U330 |
| EPOS_OC_COMPRESS_DEFLATE | Image compression is carried out  | ✓                           | -                | -      | -      | ✓        | -      | ✓        | -         | -                | ✓       | -        | -               |
| EPOS_OC_COMPRESS_NONE    | Image compression is not carried out  | ✓                           | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓         | ✓                | ✓       | ✓        | ✓               |
| EPOS_OC_PARAM_DEFAULT    | Specify the half tone treatment method.<br>(Image compression is not carried out) | ✓                           | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓         | ✓                | ✓       | ✓        | ✓               |



For TCP communication, specify EPOS\_OC\_PARAM\_DEFAULT.

#### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

#### Example

```
UIImage * imageData = Nil;

id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
               Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    ///Process///
    errorStatus = [builder addImage: imageData X: 0 Y: 0 Width: 256
                   Height: 256 Color: EPOS_OC_PARAM_DEFAULT Mode: EPOS_OC_MODE_MONO
                   Halftone: EPOS_OC_HALFTONE_DITHER Brightness: 1.0]
    Compress: EPOS_OC_COMPRESS_NONE;
    ///Process///
}
```

## addImage(Previous format)

Adds raster image printing to the command buffer. When the *Bluetooth* interface is used, white streaks may appear because printing by image data compression is not possible.

Prints the graphic rendered by the *UIImage* class.

Out of the *UI Image* class graphics, the specified scope is converted to raster image data according to this API setting. One pixel in an image equals to one printer dot. When an image contains any transparent color, the background color of the image is assumed to be white.



- To print a raster image at high speed, specify `EPOS_OC_ALIGN_LEFT` for the [addTextAlign \(p.55\)](#), and specify a multiple of 8 not exceeding the printer's paper width for the width parameter of this API.
- Multiple tone printing is not supported in Page Mode. Multiple tone graphic printing is supported in Standard Mode only.

### Syntax

```
(int) addImage: (UIImage *)data X:(long)x Y:(long)y  
Width:(long)width Height:(long)height  
Color:(int)color Mode:(int)mode  
Halftone:(int)halftone  
Brightness:(double)brightness;
```

### Parameter

- data : Specifies an instance of the *UIImage* class.
- x : Specifies the horizontal start position in the print area.  
Specifies an integer from 0 to 65534.
- y : Specifies the vertical start position in the print area.  
Specifies an integer from 0 to 65534.
- width : Specifies the width of the print area. Specifies an integer from 1 to 65535.
- height : Specifies the height of the print area. Specifies an integer from 1 to 65535.



If the area defined in the x/y parameters and the width/height parameters do not fit in the image size defined by the data parameter, `EPOS_OC_ERR_PARAM` will be returned for the return value.

- color : Specifies the color.

| Set value                          | Description                 |
|------------------------------------|-----------------------------|
| <code>EPOS_OC_COLOR_NONE</code>    | Characters are not printed. |
| <code>EPOS_OC_COLOR_1</code>       | First color                 |
| <code>EPOS_OC_PARAM_DEFAULT</code> | First color                 |



- **mode :** Specify the color mode.

| Set value             | Description   | TM printer-separate setting |                  |        |                  |        |          |          |                  |         |          |
|-----------------------|---|-----------------------------|------------------|--------|------------------|--------|----------|----------|------------------|---------|----------|
|                       |   | TM-P20                      | TM-P60/ TM-P60II | TM-P80 | TM-T20/ TM-T20II | TM-T70 | TM-T70II | TM-T81II | TM-T82/ TM-T82II | TM-T88V | TM-T90II |
| EPOS_OC_MODE_MONO     | Monochrome (2 tone)   | ✓                           | ✓                | ✓      | ✓                | ✓      | ✓        | ✓        | ✓                | ✓       | ✓        |
| EPOS_OC_MODE_GRAY16   | Multiple tone (16 tone)                                       | -                           | -                | -      | -                | -      | ✓        | -        | -                | ✓       | ✓        |
| EPOS_OC_PARAM_DEFAULT | Specify the half tone treatment method. (Monochrome (2 tone)) | ✓                           | ✓                | ✓      | ✓                | ✓      | ✓        | ✓        | ✓                | ✓       | ✓        |

- **halftone :** Specify the half tone treatment method.

| Set value                        | Description  |
|----------------------------------|--|
| EPOS_OC_HALFTONE_DITHER          | Dither<br>(This is suitable for graphic printing).                                   |
| EPOS_OC_HALFTONE_ERROR_DIFFUSION | Error diffusion<br>(This is suitable for mixed printing or characters and graphics). |
| EPOS_OC_HALFTONE_THRESHOLD       | Threshold value<br>(This is suitable for printing of characters).                    |
| EPOS_OC_PARAM_DEFAULT            | Default value (dither) selection   |



In the case of multiple tone (16 tone), this is disregarded.

- **brightness :** Specify the correction value for brightness.

| Set value                      | Description                               |
|--------------------------------|---|
| Actual figure from 0.1 to 10.0 | Brightness correction value (gamma value) |
| EPOS_OC_PARAM_DEFAULT          | Select the default value (1.0)            |



If you specify a value other than 1.0, the printing speed will become slower.

#### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

## Example

```
UIImage * imageData = Nil;

id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    ///Process///
    errorStatus = [builder addImage: imageData X: 0 Y: 0 Width: 256
                  Height: 256 Color: EPOS_OC_PARAM_DEFAULT Mode: EPOS_OC_MODE_MONO
                  Halftone: EPOS_OC_HALFTONE_DITHER Brightness: 1.0];
    ///Process///
}
```

To print an image 256 dots wide and 256 dots high in page mode:

```
UIImage * imageData = Nil;

id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    ///Process///
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPagePosition: 0 Y: 255];
    errorStatus = [builder addImage: imageData X: 0 Y: 0 Width: 256
                  Height: 256 Color: EPOS_OC_PARAM_DEFAULT Mode: EPOS_OC_MODE_MONO
                  Halftone: EPOS_OC_HALFTONE_DITHER Brightness: 1.0];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addImage(Previous format)

Adds raster image printing to the command buffer. Multiple tones cannot be printed.

When the *Bluetooth* interface is used, white streaks may appear because printing by image data compression is not possible.

Prints the graphic rendered by the *UIImage* class.

In the *UIImage* class *graphic*, changes the specified range to binary value with dither processing, and converts it to raster image data. One pixel in an image equals to one printer dot. When an image contains any transparent color, the background color of the image is assumed to be white.



- When printing in multiple tone, use [addImage \(p.69\)](#).
- To print a raster image at high speed, specify `EPOS_OC_ALIGN_LEFT` for the [addTextAlign \(p.55\)](#), and specify a multiple of 8 not exceeding the printer's paper width for the width parameter of this API.

### Syntax

```
(int) addImage: (UIImage *)data X:(long)x Y:(long)y
                    Width:(long)width Height:(long)height
                    Color:(int)color;
```

### Parameter

- `data` : Specifies an instance of the *UIImage* class.
- `x` : Specifies the horizontal start position in the print area.  
Specifies an integer from 0 to 65534.
- `y` : Specifies the vertical start position in the print area.  
Specifies an integer from 0 to 65534.
- `width` : Specifies the width of the print area. Specifies an integer from 1 to 65535.
- `height` : Specifies the height of the print area. Specifies an integer from 1 to 65535.



If the area defined in the `x/y` parameters and the `width/height` parameters do not fit in the image size defined by the `data` parameter, `EPOS_OC_ERR_PARAM` will be returned for the return value.

- `color` : Specifies the color.

| Set value                          | Description                 |
|------------------------------------|-----------------------------|
| <code>EPOS_OC_COLOR_NONE</code>    | Characters are not printed. |
| <code>EPOS_OC_COLOR_1</code>       | First color                 |
| <code>EPOS_OC_PARAM_DEFAULT</code> | First color                 |

### Return Value

| Error status                     | Description                    |
|----------------------------------|--------------------------------|
| <code>EPOS_OC_SUCCESS</code>     | Processing was successful.     |
| <code>EPOS_OC_ERR_PARAM</code>   | Invalid parameter was passed.  |
| <code>EPOS_OC_ERR_MEMORY</code>  | Could not allocate memory.     |
| <code>EPOS_OC_ERR_FAILURE</code> | An unspecified error occurred. |

## Example

```
UIImage * imageData = Nil;

id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    ///Process///
    errorStatus = [builder addImage: imageData X: 0 Y: 0 Width: 256
    Height: 256 Color: EPOS_OC_PARAM_DEFAULT];
    ///Process///
}
```

To print an image 256 dots wide and 256 dots high in page mode:

```
UIImage * imageData = Nil;

id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    ///Process///
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPagePosition: 0 Y: 255];
    errorStatus = [builder addImage: imageData X: 0 Y: 0 Width: 256
    Height: 256 Color: EPOS_OC_PARAM_DEFAULT];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addLogo

Adds NV logo printing to the command buffer.

Prints a logo registered in the NV memory of the printer.



- Register a logo in advance into the printer using the following utilities:
  - \* Model-dedicated Utility
  - \* TM Flash Logo Setup Utility
- Multiple tone printing is not supported in Page Mode. Multiple tone graphic printing is supported in Standard Mode only.

### Syntax

– (int) **addLogo**: (long)key1 Key2: (long)key2;

### Parameter

- key1 : Specifies the key code 1 of an NV logo. Specifies an integer from 32 to 126.
- key2 : Specifies the key code 2 of an NV logo. Specifies an integer from 32 to 126.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To print a NV logo with the key code parameters specified as 48, 48:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addLogo: 48 Key2: 48];
    ///Process///
}
```

## addBarcode

Adds barcode printing to the command buffer.

### Syntax

```
- (int) addBarcode: (NSString *)data Type:(int)type  
    Hri:(int)hri Font:(int)font  
    Width:(long)width  
    Height:(long)height;
```

### Parameter

- data : Specifies the barcode data as a string.



Specify a string that follows the barcode standard specified by the type parameter. If the specified string does not conform to the standard, a barcode will not be printed.

| Barcode type | Description  |
|--------------|--|
| UPC-A        | When an 11-digit number is specified, a check digit is automatically added.<br>When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.   |
| UPC-E        | Specify 0 as the first digit.<br>Specify the manufacturer code in the digits 2 to 6.<br>Specify (right-align) the item code in the digits 7 to 11. The number of item code digits varies depending on the manufacturer code. Specify 0s in empty digits. |
| EAN13        | When an 11-digit number is specified, a check digit is automatically added.<br>When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.   |
| JAN13        |  |
| EAN8         | When a 7-digit number is specified, a check digit is automatically added.<br>When an 8-digit number is specified, the 8th digit is processed as a check digit but the check digit is not validated.  |
| JAN8         |  |
| CODE39       | When the first character is *, the character is processed as the start character. In other cases, a start character is automatically added.  |
| ITF          | Start and stop codes are automatically added.<br>Check digits are not added or validated.  |
| CODABAR      | Specify a start character (A to D, a to d).<br>Specify a stop character (A to D, a to d).<br>Check digits are not added or validated.  |
| CODE93       | Start and stop characters are automatically added.<br>A check digit is automatically calculated and added.   |

| Barcode type                | Description   |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
|-----------------------------|---|-------|----|-------|----|-------|----|-------|----|---------|----|---------|----|---------|----|--------|----|----|----|
| CODE128                     | <p>Specify a start character (CODE A, CODE B, CODE C).<br/> A stop character is automatically added.<br/> A check digit is automatically calculated and added.<br/> To encode each of the following characters, specify two characters starting with the character "{":</p> <table> <tr><td>FNC1:</td><td>{1</td></tr> <tr><td>FNC2:</td><td>{2</td></tr> <tr><td>FNC3:</td><td>{3</td></tr> <tr><td>FNC4:</td><td>{4</td></tr> <tr><td>CODE A:</td><td>{A</td></tr> <tr><td>CODE B:</td><td>{B</td></tr> <tr><td>CODE C:</td><td>{C</td></tr> <tr><td>SHIFT:</td><td>{S</td></tr> <tr><td>{:</td><td>{{</td></tr> </table>   | FNC1: | {1 | FNC2: | {2 | FNC3: | {3 | FNC4: | {4 | CODE A: | {A | CODE B: | {B | CODE C: | {C | SHIFT: | {S | {: | {{ |
| FNC1:                       | {1  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| FNC2:                       | {2  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| FNC3:                       | {3  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| FNC4:                       | {4  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| CODE A:                     | {A  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| CODE B:                     | {B  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| CODE C:                     | {C  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| SHIFT:                      | {S  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| {:                          | {{  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| GS1-128                     | <p>A start character, FNC1, a check digit, and a stop character are automatically added.<br/> To automatically calculate and add a check digit for an application identifier (AI) and the subsequent data, specify the character "*" in the position of the check digit.<br/> You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data.<br/> You can insert spaces between an application identifier (AI) and data. The spaces are used as HRI print characters and are not encoded as data.<br/> To encode each of the following characters, specify two characters starting with the character "{":</p> <table> <tr><td>FNC1:</td><td>{1</td></tr> <tr><td>FNC3:</td><td>{3</td></tr> <tr><td>(:</td><td>{{</td></tr> <tr><td>):</td><td>{}</td></tr> <tr><td>*:</td><td>{*</td></tr> <tr><td>{:</td><td>{{</td></tr> </table> | FNC1: | {1 | FNC3: | {3 | (:    | {{ | ):    | {} | *:      | {* | {:      | {{ |         |    |        |    |    |    |
| FNC1:                       | {1  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| FNC3:                       | {3  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| (:                          | {{  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| ):                          | {}  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| *:                          | {*  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| {:                          | {{  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| GS1 DataBar Omnidirectional | Specify a 13-digit global trade item number (GTIN) not including an application identifier (AI) or a check digit.   |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| GS1 DataBar Truncated       |   |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| GS1 DataBar Limited         |   |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| GS1 DataBar Expanded        | <p>You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data.<br/> To encode each of the following characters, specify two characters starting with the character "{":</p> <table> <tr><td>FNC1:</td><td>{1</td></tr> <tr><td>(:</td><td>{{</td></tr> <tr><td>):</td><td>{}</td></tr> </table>  | FNC1: | {1 | (:    | {{ | ):    | {} |       |    |         |    |         |    |         |    |        |    |    |    |
| FNC1:                       | {1  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| (:                          | {{  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |
| ):                          | {}  |       |    |       |    |       |    |       |    |         |    |         |    |         |    |        |    |    |    |

To specify binary data that cannot be represented by character strings, use the following escape sequences.

| String | Description  |
|--------|--------------|
| \xnn   | Control code |
| \\     | Back slash   |

- **type :** Specifies the barcode type.

| Set value                                   | Barcode type                |
|---|-----------------------------|
| EPOS_OC_BARCODE_UPC_A                       | UPC-A                       |
| EPOS_OC_BARCODE_UPC_E                       | UPC-E                       |
| EPOS_OC_BARCODE_EAN13                       | EAN13                       |
| EPOS_OC_BARCODE_JAN13                       | JAN13                       |
| EPOS_OC_BARCODE_EAN8                        | EAN8                        |
| EPOS_OC_BARCODE_JAN8                        | JAN8                        |
| EPOS_OC_BARCODE_CODE39                      | CODE39                      |
| EPOS_OC_BARCODE_ITF                         | ITF                         |
| EPOS_OC_BARCODE_CODABAR                     | CODABAR                     |
| EPOS_OC_BARCODE_CODE93                      | CODE93                      |
| EPOS_OC_BARCODE_CODE128                     | CODE128                     |
| EPOS_OC_BARCODE_GS1_128                     | GS1-128                     |
| EPOS_OC_BARCODE_GS1_DATABAR_OMNIDIRECTIONAL | GS1 DataBar Omnidirectional |
| EPOS_OC_BARCODE_GS1_DATABAR_TRUNCATED       | GS1 DataBar Truncated       |
| EPOS_OC_BARCODE_GS1_DATABAR_LIMITED         | GS1 DataBar Limited         |
| EPOS_OC_BARCODE_GS1_DATABAR_EXPANDED        | GS1 DataBar Expanded        |

- **hri :** Specifies the HRI position.

| Set value                 | Description                      |
|---------------------------|----------------------------------|
| EPOS_OC_HRI_NONE(default) | HRI not printed                  |
| EPOS_OC_HRI_ABOVE         | Above the barcode                |
| EPOS_OC_HRI_BELOW         | Below the barcode                |
| EPOS_OC_HRI_BOTH          | Both above and below the barcode |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.     |

- **font :** Specifies the HRI font.

| Set value                 | Description                  |
|---------------------------|------------------------------|
| EPOS_OC_FONT_A(default)   | Font A                       |
| EPOS_OC_FONT_B            | Font B                       |
| EPOS_OC_FONT_C            | Font C                       |
| EPOS_OC_FONT_D            | Font D                       |
| EPOS_OC_FONT_E            | Font E                       |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting. |



- width : Specifies the width of each module in dots. Specifies an integer from 2 to 6.

| Set value                 | Description                           |
|---------------------------|---------------------------------------|
| Integer from 2 to 6       | The width of each module. (Unit: dot) |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.          |

- height : Specifies the barcode height in dots. Specifies an integer from 1 to 255.

| Set value                 | Description                     |
|---------------------------|---------------------------------|
| Integer from 1 to 255     | The barcode height. (Unit: dot) |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.    |

#### *Return Value*

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

## Example

To print barcodes:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addBarcode: @"01234567890"
                                Type: EPOS_OC_BARCODE_UPC_A Hri: EPOS_OC_HRI_BELOW
                                Font: EPOS_OC_PARAM_UNSPECIFIED Width: 2 Height: 64];
    errorStatus = [builder addBarcode: @"01234500005"
                                Type: EPOS_OC_BARCODE_UPC_E Hri: EPOS_OC_PARAM_UNSPECIFIED
                                Font: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
                                Height: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"201234567890"
                                Type: EPOS_OC_BARCODE_EAN13 Hri: EPOS_OC_PARAM_UNSPECIFIED
                                Font: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
                                Height: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"201234567890"
                                Type: EPOS_OC_BARCODE_JAN13 Hri: EPOS_OC_PARAM_UNSPECIFIED
                                Font: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
                                Height: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"2012345" Type:
                                EPOS_OC_BARCODE_EAN8
                                Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
                                Width: EPOS_OC_PARAM_UNSPECIFIED Height:
                                EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"2012345" Type:
                                EPOS_OC_BARCODE_JAN8
                                Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
                                Width: EPOS_OC_PARAM_UNSPECIFIED Height:
                                EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"ABCDE" Type:
                                EPOS_OC_BARCODE_CODE39
                                Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
                                Width: EPOS_OC_PARAM_UNSPECIFIED Height:
                                EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"012345" Type: EPOS_OC_BARCODE_ITF
                                Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
                                Width: EPOS_OC_PARAM_UNSPECIFIED Height:
                                EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"A012345A"
                                Type: EPOS_OC_BARCODE_CODABAR Hri: EPOS_OC_PARAM_UNSPECIFIED
                                Font: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
                                Height: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"ABCDE" Type:
                                EPOS_OC_BARCODE_CODE93
                                Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
                                Width: EPOS_OC_PARAM_UNSPECIFIED Height:
                                EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addBarcode: @"{Babcde"
                                Type: EPOS_OC_BARCODE_CODE128 Hri: EPOS_OC_PARAM_UNSPECIFIED
                                Font: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
                                Height: EPOS_OC_PARAM_UNSPECIFIED];
}
```

```

errorStatus = [builder addBarcode: @"(01)201234567890*"
    Type: EPOS_OC_BARCODE_GS1_128 Hri: EPOS_OC_PARAM_UNSPECIFIED
    Font: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
    Height: EPOS_OC_PARAM_UNSPECIFIED];
errorStatus = [builder addBarcode: @"0201234567890"
    Type: EPOS_OC_BARCODE_GS1_DATABAR_OMNIDIRECTIONAL
    Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
    Width: EPOS_OC_PARAM_UNSPECIFIED Height:
    EPOS_OC_PARAM_UNSPECIFIED];
errorStatus = [builder addBarcode: @"0201234567890"
    Type: EPOS_OC_BARCODE_GS1_DATABAR_TRUNCATED
    Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
    Width: EPOS_OC_PARAM_UNSPECIFIED Height:
    EPOS_OC_PARAM_UNSPECIFIED];
errorStatus = [builder addBarcode: @"0201234567890"
    Type: EPOS_OC_BARCODE_GS1_DATABAR_LIMITED
    Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
    Width: EPOS_OC_PARAM_UNSPECIFIED Height:
    EPOS_OC_PARAM_UNSPECIFIED];
errorStatus = [builder addBarcode: @"(01)2012345678903"
    Type: EPOS_OC_BARCODE_GS1_DATABAR_EXPANDED
    Hri: EPOS_OC_PARAM_UNSPECIFIED Font: EPOS_OC_PARAM_UNSPECIFIED
    Width: EPOS_OC_PARAM_UNSPECIFIED Height:
    EPOS_OC_PARAM_UNSPECIFIED];

```

```

    ///Process///

```

```

}

```

## addSymbol

Adds 2D-Code printing to the command buffer.

### Syntax

```
- (int) addSymbol:(NSString *)data Type:(int)type  
    Level:(int)level Width:(long)width  
    Height:(long)height Size:(long)size;
```

### Parameter

- data : Specifies 2D-Code data as a character string.

| 2D-Code type     | Description   |
|------------------|---|
| Standard PDF417  | Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string.<br><br>The data area can contain up to 928 code words in a maximum of 90 rows, each of which can contain up to 30 code words.   |
| Truncated PDF417 |   |
| QR Code Model 1  | Convert the character string to the string in Shift-JIS, apply the escape sequence, and then encode the string based on the data type as shown below.<br><br>Number: 0 to 9<br>Alphanumeric character: 0 to 9, A to Z, space, \$, %, *, +, -, ., /, :<br>Kanji character: Shift-JIS value<br>8-bit, byte data: 0x00 to 0xff |
| QR Code Model 2  |   |

| 2D-Code type                        | Description   |
|-------------------------------------|---|
| MaxiCode Mode 2                     | Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string.   |
| MaxiCode Mode 3                     |   |
| MaxiCode Mode 4                     |   |
| MaxiCode Mode 5                     |   |
| MaxiCode Mode 6                     | <p>In Modes 2 and 3, when the first piece of data is ()&gt;\ x1e01\x1dyy (where yy is a two-digit number), this is processed as the message header, and the subsequent data is processed as the primary message. In other cases, from the first piece of data, data is processed as the primary message.</p> <p>In Mode 2, specify the primary message in the following format:</p> <p>Postal code (1- to 9-digit number) GS:(\x1d) ISO country code (1- to 3-digit number) GS:(\x1d) Service class code (1- to 3-digit number)</p> <p>In Mode 3, specify the primary message in the following format:</p> <p>Postal code (1 to 6 pieces of data convertible by Code Set A) GS:(\x1d) ISO country code (1- to 3-digit number) GS:(\x1d) Service class code (1- to 3-digit number)</p> |
| GS1 DataBar Stacked                 | Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string.   |
| GS1 DataBar Stacked Omnidirectional |   |
| GS1 DataBar Expanded Stacked        | <p>Specify a 13-digit global trade item number (GTIN) not including an application identifier (AI) or a check digit.</p> <p>Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string.</p> <p>You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{":</p> <p>FNC1: {1<br/>(: {(<br/>): }</p>  |
| Aztec Code Full-Range mode          | <p>After converting the character string to UTF-8, conduct the escape sequence and encode.</p> <p>Up to 3,067 characters of text, 3,832 numerical figures and 1,914 bytes of binary data can be specified.</p>  |
| Aztec Code Compact mode             | <p>After converting the character string to UTF-8, conduct the escape sequence and encode.</p> <p>Up to 89 characters of text, 110 numerical figures and 53 bytes of binary data can be specified.</p>  |

| 2D-Code type                   | Description  |
|--------------------------------|--|
| DataMatrix square              | After converting the character string to UTF-8, conduct the escape sequence and encode.<br><br>The symbol is either a square ranging in size from 10 lines x 10 rows ~ 144 lines ~ 144 rows, or a rectangle comprising 8 lines, 12 lines or 16 lines. Up to 2,335 alphanumeric, 3,116 numerical figures and 1,556 bytes of binary data can be specified. |
| DataMatrix rectangle, 8 lines  |  |
| DataMatrix rectangle, 12 lines |  |
| DataMatrix rectangle, 16 lines |  |

To specify binary data that cannot be represented by character strings, use the following escape sequences.

| String | Description  |
|--------|--------------|
| \xnn   | Control code |
| \\     | Back slash   |

- type : Specifies the 2D-Code type.

| Set value  | 2D-Code type                        |
|--|-------------------------------------|
| EPOS_OC_SYMBOL_PDF417_STANDARD                     | Standard PDF417                     |
| EPOS_OC_SYMBOL_PDF417_TRUNCATED                    | Truncated PDF417                    |
| EPOS_OC_SYMBOL_QRCODE_MODEL_1                      | QR Code Model 1                     |
| EPOS_OC_SYMBOL_QRCODE_MODEL_2                      | QR Code Model 2                     |
| EPOS_OC_SYMBOL_MAXICODE_MODE_2                     | MaxiCode Mode 2                     |
| EPOS_OC_SYMBOL_MAXICODE_MODE_3                     | MaxiCode Mode 3                     |
| EPOS_OC_SYMBOL_MAXICODE_MODE_4                     | MaxiCode Mode 4                     |
| EPOS_OC_SYMBOL_MAXICODE_MODE_5                     | MaxiCode Mode 5                     |
| EPOS_OC_SYMBOL_MAXICODE_MODE_6                     | MaxiCode Mode 6                     |
| EPOS_OC_SYMBOL_GS1_DATABAR_STACKED                 | GS1 DataBar Stacked                 |
| EPOS_OC_SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL | GS1 DataBar Stacked Omnidirectional |
| EPOS_OC_SYMBOL_GS1_DATABAR_EXPANDED_STACKED        | GS1 DataBar Expanded Stacked        |
| EPOS_OC_SYMBOL_AZTECCODE_FULLRANGE                 | Aztec Code Full-Range mode          |
| EPOS_OC_SYMBOL_AZTECCODE_COMPACT                   | Aztec Code Compact mode             |
| EPOS_OC_SYMBOL_DATAMATRIX_SQUARE                   | DataMatrix square                   |
| EPOS_OC_SYMBOL_DATAMATRIX_RECTANGLE_8              | DataMatrix rectangle, 8 lines       |
| EPOS_OC_SYMBOL_DATAMATRIX_RECTANGLE_12             | DataMatrix rectangle, 12 lines      |
| EPOS_OC_SYMBOL_DATAMATRIX_RECTANGLE_16             | DataMatrix rectangle, 16 lines      |

- level : Specifies the error correction level.

| Set value       | Description                     |
|-----------------|---------------------------------|
| EPOS_OC_LEVEL_0 | PDF417 error correction level 0 |
| EPOS_OC_LEVEL_1 | PDF417 error correction level 1 |
| EPOS_OC_LEVEL_2 | PDF417 error correction level 2 |
| EPOS_OC_LEVEL_3 | PDF417 error correction level 3 |

| Set value                 | Description                                      |
|---------------------------|--|
| EPOS_OC_LEVEL_4           | PDF417 error correction level 4                  |
| EPOS_OC_LEVEL_5           | PDF417 error correction level 5                  |
| EPOS_OC_LEVEL_6           | PDF417 error correction level 6                  |
| EPOS_OC_LEVEL_7           | PDF417 error correction level 7                  |
| EPOS_OC_LEVEL_8           | PDF417 error correction level 8                  |
| EPOS_OC_LEVEL_L           | QR Code error correction level L                 |
| EPOS_OC_LEVEL_M           | QR Code error correction level M                 |
| EPOS_OC_LEVEL_Q           | QR Code error correction level Q                 |
| EPOS_OC_LEVEL_H           | QR Code error correction level H                 |
| Integer from 5 to 95      | Aztec Code error correction level (percent unit) |
| EPOS_OC_LEVEL_DEFAULT     | Default level                                    |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting.                     |



- Select the level according to the 2D-Code type.
- MaxiCode/two-dimensional GS1 DataBar/DataMatrix, select EPOS\_OC\_LEVEL\_DEFAULT.

- width : Specifies the module width.

| Set value                 | Description                  |
|---------------------------|------------------------------|
| Integer from 1 to 255     | Module width                 |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting. |



MaxiCode is ignored.

- height : Specifies the module height.

| Set value                 | Description                  |
|---------------------------|------------------------------|
| Integer from 1 to 255     | Module height                |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting. |



QR Code/MaxiCode/two-dimensional GS1 DataBar/Aztec Code/DataMatrix are ignored.

- size : Specifies the 2D-Code maximum size.

| Set value                 | Description                  |
|---------------------------|------------------------------|
| Integer from 0 to 65535   | 2D-Code maximum size         |
| EPOS_OC_PARAM_UNSPECIFIED | Retains the current setting. |



QR Code/MaxiCode/Aztec Code/DataMatrix are ignored.

## Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

## Example

### To print 2D-Code:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addSymbol: @"ABCDE"
        Type: EPOS_OC_SYMBOL_PDF417_STANDARD Level:
        EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
        Height: EPOS_OC_PARAM_UNSPECIFIED Size: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addSymbol: @"ABCDE"
        Type: EPOS_OC_SYMBOL_QRCODE_MODEL_2 Level: EPOS_OC_LEVEL_Q
        Width: EPOS_OC_PARAM_UNSPECIFIED Height: EPOS_OC_PARAM_UNSPECIFIED
        Size: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addSymbol: @"908063840\\x1d850\\x1d001\\x1d\\x04"
        Type: EPOS_OC_SYMBOL_MAXICODE_MODE_2 Level: EPOS_OC_PARAM_UNSPECIFIED
        Width: EPOS_OC_PARAM_UNSPECIFIED Height: EPOS_OC_PARAM_UNSPECIFIED
        Size: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addSymbol: @"0201234567890"
        Type: EPOS_OC_SYMBOL_GS1_DATABAR_STACKED
        Level: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
        Height: EPOS_OC_PARAM_UNSPECIFIED Size: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addSymbol: @"0201234567890"
        Type: EPOS_OC_SYMBOL_GS1_DATABAR_STACKED_OMNIDIRECTIONAL
        Level: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
        Height: EPOS_OC_PARAM_UNSPECIFIED Size: EPOS_OC_PARAM_UNSPECIFIED];
    errorStatus = [builder addSymbol: @"(01)02012345678903"
        Type: EPOS_OC_SYMBOL_GS1_DATABAR_EXPANDED_STACKED
        Level: EPOS_OC_PARAM_UNSPECIFIED Width: EPOS_OC_PARAM_UNSPECIFIED
        Height: EPOS_OC_PARAM_UNSPECIFIED Size: EPOS_OC_PARAM_UNSPECIFIED];
    ///Process///
}
```



## addPageBegin

Adds the switching to page mode to the command buffer. The page mode process starts.



Use this API function with [addPageEnd \(p.90\)](#).

### Syntax

- (int) **addPageBegin**;

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To print the characters "ABCDE" in page mode:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addText: @"ABCDE"];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addPageEnd

Adds the end of page mode to the command buffer. The page mode process ends.



Use this API function with [addPageBegin](#) (p.89).

### Syntax

- (int) **addPageEnd**;

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To print the characters "ABCDE" in page mode:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addText: @"ABCDE"];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addPageArea

Adds the print area in page mode to the command buffer.

Specifies the print area in page mode (coordinates). After this API function, specify a print data API function such as the addText method.



- Specify a print area to cover the content to be printed. If the print data extends beyond the print area, the print result will be such that the print data has been printed incompletely.
- Use this API function by inserting it between [addPageBegin \(p.89\)](#) and [addPageEnd \(p.90\)](#).

### Syntax

```
(int) addPageArea: (long)x Y: (long)y Width: (long)width  
Height: (long)height;
```

### Parameter

- **x**: Specifies the origin of the horizontal axis (in dots). Specifies an integer from 0 to 65535. 0 is the left end of the printer's printable area.
- **y**: Specifies the origin of the vertical axis (in dots). Specifies an integer from 0 to 65535. 0 is the position in which no paper feed has been performed.
- **width**: Specifies the width of the print area (in dots). Specifies an integer from 1 to 65535.
- **height**: Specifies the height of the print area (in dots). Specifies an integer from 1 to 65535.



Determine the width and height of the print area according to the print direction setting. Otherwise, the print data might not be printed completely.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To specify the print area with the origin (100, 50), a width of 200 dots, and a height of 30 dots and print the characters "ABCDE":

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"  
             Lang: EPOS_OC_MODEL_ANK];  
if ( builder != nil ) {  
    int errorStatus = EPOS_OC_SUCCESS;  
    errorStatus = [builder addPageBegin];  
    errorStatus = [builder addPageArea: 100 Y: 50 Width: 200 Height: 30];  
    errorStatus = [builder addText: @"ABCDE"];  
    errorStatus = [builder addPageEnd];  
    ///Process///  
}
```

## addPageDirection

Adds the page mode print direction setting to the command buffer. Specifies the print direction in page mode. This function can be omitted if rotation is not required.



Use this API function by inserting it between [addPageBegin \(p.89\)](#) and [addPageEnd \(p.90\)](#).

### Syntax

```
- (int) addPageDirection: (int) dir;
```

### Parameter

- dir : Specifies the print direction in page mode.

| Set value                                | Description  |
|--|--|
| EPOS_OC_DIRECTION_LEFT_TO_RIGHT(default) | Left to right<br>(No rotation.Data is printed from the top left corner to the right.)                                  |
| EPOS_OC_DIRECTION_BOTTOM_TO_TOP          | Bottom to top<br>(Counterclockwise rotation by 90 degrees.<br>Data is printed from the bottom left corner to the top.) |
| EPOS_OC_DIRECTION_RIGHT_TO_LEFT          | Right to left<br>(Rotation by 180 degrees.Data is printed from the bottom right corner to the left.)                   |
| EPOS_OC_DIRECTION_TOP_TO_BOTTOM          | Top to bottom<br>(Clockwise rotation by 90 degrees.<br>Data is printed from the top right corner to the bottom.)       |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

*Example*

To print the characters "ABCDE" by rotating them 90 degrees clockwise:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPageArea: 100 Y: 50 Width: 30 Height: 200];
    errorStatus = [builder addPageDirection:
        EPOS_OC_DIRECTION_TOP_TO_BOTTOM];
    errorStatus = [builder addText: @"ABCDE"];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addPagePosition

Adds the page mode print-position-set area to the command buffer.

Specifies the print start position (coordinates) in the area specified by the addPageArea method.



Use this API function by inserting it between [addPageBegin \(p.89\)](#) and [addPageEnd \(p.90\)](#).

### Syntax

- (int) **addPagePosition**: (long)x Y: (long)y;

### Parameter

- x : Specifies the horizontal print position (in dots). Specifies an integer from 0 to 65535.
- y : Specifies the vertical print position (in dots). Specifies an integer from 0 to 65535.



Specify the print start position (coordinates) according to the content to be printed. Refer to the following.

- \* To print a character string:  
Specify the left end of the baseline for the first character. This can be omitted for left-aligned printing of standard-sized characters. To print double-sized height characters, specify a value equal to or greater than 42 for y.
- \* To print a barcode:  
Specify the bottom left of the symbol. And specify the barcode height for y.
- \* To print a graphic/logo:  
Specify the bottom left of the graphic data. And specify the graphic data height for y.
- \* To print a 2D-Code:  
Specify the top left of the symbol. This can be omitted when printing from the top left.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To specify (50,30) for the print start position in the area specified by the addPageArea method and print the characters "ABCDE":

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPageArea: 100 Y: 50 Width: 200 Height: 100];
    errorStatus = [builder addPagePosition: 50 Y: 30];
    errorStatus = [builder addText: @"ABCDE"];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addPageLine

Adds line drawing in page mode to the command buffer. Draws a line in page mode.



- Diagonal lines cannot be drawn.
- Use this API function by inserting it between [addPageBegin \(p.89\)](#) and [addPageEnd \(p.90\)](#).

### Syntax

```
(int) addPageLine: (long)x1 Y1:(long)y1 X2:(long)x2  
Y2:(long)y2 Style:(int)style;
```

### Parameter

- x1 : Specifies the horizontal start position of the line (in dots).  
Specifies an integer from 0 to 65535.
- y1 : Specifies the vertical start position of the line (in dots).  
Specifies an integer from 0 to 65535.
- x2 : Specifies the horizontal end position of the line (in dots).  
Specifies an integer from 0 to 65535.
- y2 : Specifies the vertical end position of the line (in dots).  
Specifies an integer from 0 to 65535.
- style : Specifies the line type.

| Set value                  | Description         |
|----------------------------|---------------------|
| EPOS_OC_LINE_THIN          | Solid line: Thin    |
| EPOS_OC_LINE_MEDIUM        | Solid line: Medium  |
| EPOS_OC_LINE_THICK         | Solid line: Thick   |
| EPOS_OC_LINE_THIN_DOUBLE   | Double line: Thin   |
| EPOS_OC_LINE_MEDIUM_DOUBLE | Double line: Medium |
| EPOS_OC_LINE_THICK_DOUBLE  | Double line: Thick  |
| EPOS_OC_PARAM_DEFAULT      | Solid line: Thin    |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To draw a thin solid line between the start position(100, 0) and the end position(500, 0):

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-P60"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPageLine: 100 Y1: 0 X2: 500 Y2: 0
        Style: EPOS_OC_LINE_THIN];
    errorStatus = [builder addPageEnd];
    ///Process///
}
```



## addPageRectangle

Adds rectangle drawing in page mode to the command buffer. Draws a rectangle in page mode.



Use this API function by inserting it between [addPageBegin \(p.89\)](#) and [addPageEnd \(p.90\)](#).

### Syntax

```
- (int) addPageRectangle: (long)x1 Y1:(long)y1
                        X2:(long)x2 Y2:(long)y2
                        Style:(int)style;
```

### Parameter

- **x1** : Specifies the horizontal start position of the line (in dots).  
Specifies an integer from 0 to 65535.
- **y1** : Specifies the vertical start position of the line (in dots).  
Specifies an integer from 0 to 65535.
- **x2** : Specifies the horizontal end position of the line (in dots).  
Specifies an integer from 0 to 65535.
- **y2** : Specifies the vertical end position of the line (in dots).  
Specifies an integer from 0 to 65535.
- **style** : Specifies the line type.

| Set value                  | Description         |
|----------------------------|---------------------|
| EPOS_OC_LINE_THIN          | Solid line: Thin    |
| EPOS_OC_LINE_MEDIUM        | Solid line: Medium  |
| EPOS_OC_LINE_THICK         | Solid line: Thick   |
| EPOS_OC_LINE_THIN_DOUBLE   | Double line: Thin   |
| EPOS_OC_LINE_MEDIUM_DOUBLE | Double line: Medium |
| EPOS_OC_LINE_THICK_DOUBLE  | Double line: Thick  |
| EPOS_OC_PARAM_DEFAULT      | Solid line: Thin    |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To draw a rectangle with a thin line, with the start position(100, 0) and the end position(500, 200) as its vertexes:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-P60"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPageBegin];
    errorStatus = [builder addPageRectangle: 100 Y1: 0 X2: 500 Y2: 200
        Style: EPOS_OC_LINE_THIN);
    errorStatus = [builder addPageEnd];
    ///Process///
}
```

## addCut

Adds paper cut to the command buffer. Sets paper cut.



Not available in page mode.

### Syntax

```
- (int) addCut: (int) type;
```

### Parameter

- type : Specifies the paper cut type.

| Set value             | Description   |
|-----------------------|---|
| EPOS_OC_CUT_NO_FEED   | Cut without feeding<br>(The paper is cut without being fed.)  |
| EPOS_OC_CUT_FEED      | Feed cut<br>(The paper is fed to the cut position and then is cut.)                                   |
| EPOS_OC_CUT_RESERVE   | Cut reservation<br>(Printing continues until the cut position is reached, at which the paper is cut.) |
| EPOS_OC_PARAM_DEFAULT | Feed cut<br>(The paper is fed to the cut position and then is cut.)                                   |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To perform feed cut operation:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addCut: EPOS_OC_CUT_FEED];
    ///Process///
}
```

## addPulse

Adds the drawer kick to the command buffer. Sets the drawer kick.



- Not available in page mode.
- The drawer and the buzzer cannot be used together.

### Syntax

- (int) **addPulse**: (int)drawer Time: (int)time;

### Parameter

- drawer : Specifies the drawer kick connector.

| Set value             | Description                            |
|-----------------------|--|
| EPOS_OC_DRAWER_1      | Pin 2 of the drawer kick-out connector |
| EPOS_OC_DRAWER_2      | Pin 5 of the drawer kick-out connector |
| EPOS_OC_PARAM_DEFAULT | Pin 2 of the drawer kick-out connector |

- time : Specifies the ON time of the drawer kick signal.

| Set value             | Description |
|-----------------------|-------------|
| EPOS_OC_PULSE_100     | 100 ms      |
| EPOS_OC_PULSE_200     | 200 ms      |
| EPOS_OC_PULSE_300     | 300 ms      |
| EPOS_OC_PULSE_400     | 400 ms      |
| EPOS_OC_PULSE_500     | 500 ms      |
| EPOS_OC_PARAM_DEFAULT | 100 ms      |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To send a 100 msec pulse signal to the pin 2 of the drawer kick connector:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addPulse: EPOS_OC_DRAWER_1
        Time: EPOS_OC_PULSE_100];
    ///Process///
}
```

## addSound

Adds the turning on of the buzzer to the command buffer. Sets the buzzer.



- Not available in page mode.
- The buzzer function and the drawer cannot be used together.
- This API function cannot be used if the printer is not provided with the buzzer.

### Syntax

```
(int) addSound: (int)pattern Repeat: (long)repeat  
Cycle: (long)cycle;
```

### Parameter

- pattern : Specifies the buzzer pattern.

| Set value                 | Description                                      |
|---------------------------|--|
| EPOS_OC_PATTERN_A         | Pattern A (Optional Buzzer)                      |
| EPOS_OC_PATTERN_B         | Pattern B (Optional Buzzer)                      |
| EPOS_OC_PATTERN_C         | Pattern C (Optional Buzzer)                      |
| EPOS_OC_PATTERN_D         | Pattern D (Optional Buzzer)                      |
| EPOS_OC_PATTERN_E         | Pattern E (Optional Buzzer)                      |
| EPOS_OC_PATTERN_ERROR     | Error sound pattern (Optional Buzzer)            |
| EPOS_OC_PATTERN_PAPER_END | Pattern when there is no paper (Optional Buzzer) |
| EPOS_OC_PATTERN_1         | Pattern 1 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_2         | Pattern 2 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_3         | Pattern 3 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_4         | Pattern 4 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_5         | Pattern 5 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_6         | Pattern 6 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_7         | Pattern 7 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_8         | Pattern 8 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_9         | Pattern 9 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_10        | Pattern 10 (Internal Buzzer)                     |
| EPOS_OC_PARAM_DEFAULT     | Pattern A  |

- repeat : Specifies the number of repeats.

| Set value             | Description       |
|-----------------------|-------------------|
| 1 to 255              | Number of repeats |
| EPOS_OC_PARAM_DEFAULT | One time          |

- cycle : This specifies the buzzer sounding cycle (in units of milliseconds)

| Set value             | Description                |
|-----------------------|----------------------------|
| 1000 to 25500         | 1000 to 25500 milliseconds |
| EPOS_OC_PARAM_DEFAULT | 1000 milliseconds          |



"Pattern A to E"/ "Error sound pattern"/"Pattern when there is no paper" is disregarded.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

#### When sounding pattern 1 three times at 1,000 millisecond cycles

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addSound: EPOS_OC_PATTERN_1 Repeat: 3 Cycle: 1000];
    ///Process///
}
```

## addSound(Previous format)

Adds the turning on of the buzzer to the command buffer. Sets the buzzer.



- You cannot set the buzzer sounding cycle. If you want to optionally set the buzzer sounding cycle (milliseconds), use [addSound \(p.101\)](#).
- Not available in page mode.
- The buzzer function and the drawer cannot be used together.
- This API function cannot be used if the printer is not provided with the buzzer.

### Syntax

```
(int) addSound:(int)pattern Repeat:(long)repeat;
```

### Parameter

- pattern : Specifies the buzzer pattern.

| Set value                 | Description                                      |
|---------------------------|--|
| EPOS_OC_PATTERN_A         | Pattern A (Optional Buzzer)                      |
| EPOS_OC_PATTERN_B         | Pattern B (Optional Buzzer)                      |
| EPOS_OC_PATTERN_C         | Pattern C (Optional Buzzer)                      |
| EPOS_OC_PATTERN_D         | Pattern D (Optional Buzzer)                      |
| EPOS_OC_PATTERN_E         | Pattern E (Optional Buzzer)                      |
| EPOS_OC_PATTERN_ERROR     | Error sound pattern (Optional Buzzer)            |
| EPOS_OC_PATTERN_PAPER_END | Pattern when there is no paper (Optional Buzzer) |
| EPOS_OC_PATTERN_1         | Pattern 1 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_2         | Pattern 2 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_3         | Pattern 3 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_4         | Pattern 4 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_5         | Pattern 5 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_6         | Pattern 6 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_7         | Pattern 7 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_8         | Pattern 8 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_9         | Pattern 9 (Internal Buzzer)                      |
| EPOS_OC_PATTERN_10        | Pattern 10 (Internal Buzzer)                     |
| EPOS_OC_PARAM_DEFAULT     | Pattern A  |

- repeat : Specifies the number of repeats.

| Set value             | Description       |
|-----------------------|-------------------|
| 1 to 255              | Number of repeats |
| EPOS_OC_PARAM_DEFAULT | One time          |

## Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

## Example

To repeat the sound pattern A three times:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addSound: EPOS_OC_PATTERN_A Repeat: 3];
    ///Process///
}
```



## addFeedPosition

Adds label / black mark paper feeding to the command buffer.

### Syntax

- (int) **addFeedPosition**: (int)position;

### Parameter

- position : Specifies the feed position.

| Set value                | Description                            |
|--------------------------|--|
| EPOS_OC_FEED_PEELING     | Feeds to the peeling position.         |
| EPOS_OC_FEED_CUTTING     | Feeds to the cutting position.         |
| EPOS_OC_FEED_CURRENT_TOF | Feeds to the top of the current label. |
| EPOS_OC_FEED_NEXT_TOF    | Feeds to the top of the next label.    |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To feed a label paper to the peeling position:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-P60II"
              Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addFeedPosition: EPOS_OC_FEED_PEELING];
    ///Process///
}
```

## addLayout

Adds label / black mark paper layout information to the command buffer.

### Syntax

```
(int) addLayout:(int)type  
Width:(long)width Height:(long)height  
MarginTop:(long)marginTop  
MarginBottom:(long)marginBottom  
OffsetCut:(long)offsetCut  
OffsetLabel:(long)offsetLabel;
```

### Parameter

- type : Specifies the paper type.

| Set value                 | Description                     |
|---------------------------|---------------------------------|
| EPOS_OC_LAYOUT_RECEIPT    | Receipt paper (no black mark)   |
| EPOS_OC_LAYOUT_LABEL      | Label paper (no black mark)     |
| EPOS_OC_LAYOUT_LABEL_BM   | Label paper (with black mark)   |
| EPOS_OC_LAYOUT_RECEIPT_BM | Receipt paper (with black mark) |

- width : Specifies paper width (in units of 0.1 mm). Specifies an integer from 1 to 10000.
- height : Specifies the distance (in units of 0.1 mm) from the standard printing position to the next standard printing position. Specifies an integer from 0 to 10000.  
If "0" is specified, the distance from the standard printing position to the next standard printing position is detected automatically.
- marginTop : Specifies the distance (in units of 0.1 mm) from the standard printing position to the top position. Specifies an integer from -9999 to 10000.
- marginBottom : Specifies the distance (in units of 0.1 mm) from the standard eject position to the bottom edge of the printable area. Specifies an integer from -9999 to 10000.
- offsetCut : Specifies the distance (in units of 0.1 mm) from the standard eject position to the cutting position. Specifies an integer from -9999 to 10000.
- offsetLabel : Specifies the distance (in units of 0.1 mm) from the standard eject position to the bottom edge of the label. Specifies an integer from 0 to 10000.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

*Example*

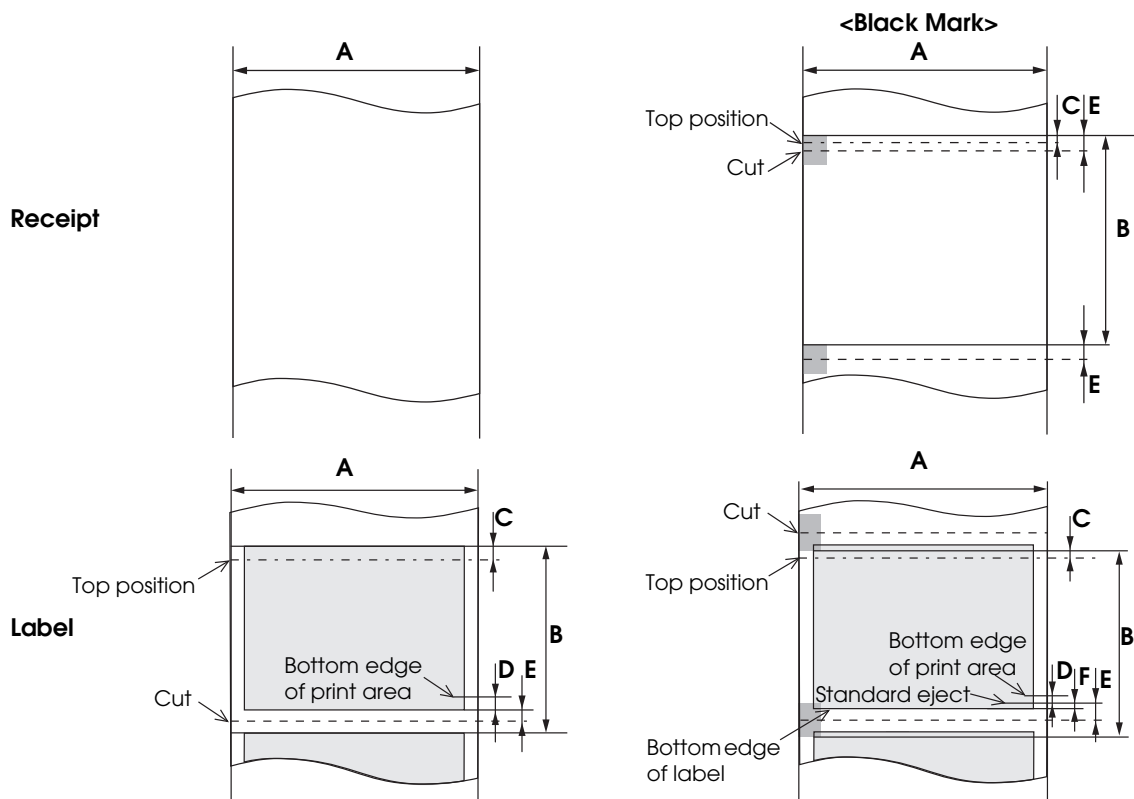
To set 60 mm label paper (black mark):

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-P60II"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [builder addLayout: EPOS_OC_PAPER_TYPE_LABEL_BM
        Width:600 Height:0 MarginTop:15 MarginBottom:-15
        OffsetCut:15 OffsetLabel:0];

    ///Process///
}
```

**Detailed description**

- See below for the parameters that can be specified for each type of paper, and the positions for those parameters.



| Mark | Parameter    | Set value  |                      |            |                    |
|------|--------------|------------|----------------------|------------|--------------------|
|      |              | Receipt    | Receipt (Black mark) | Label      | Label (Black mark) |
| A    | width        | 1 to 10000 | 1 to 10000           | 1 to 10000 | 1 to 10000         |
| B    | height       | 0          | 0 to 10000           | 0 to 10000 | 0 to 10000         |
| C    | marginTop    | 0          | -9999 to 10000       | 0 to 10000 | -9999 to 10000     |
| D    | marginBottom | 0          | 0                    | -9999 to 0 | -9999 to 10000     |
| E    | offsetCut    | 0          | -9999 to 10000       | 0 to 10000 | 0 to 10000         |
| F    | offsetLabel  | 0          | 0                    | 0          | 0 to 10000         |

## addCommand

Adds commands to the command buffer. Sends ESC/POS commands.



For information about ESC/POS commands, see the ESC/POS Command Reference.  
You can view the ESC/POS Command Reference from <https://reference.epson-biz.com/>.

### Syntax

```
- (int) addCommand: (NSData *)data;
```

### Parameter

- data : Specifies ESC/POS command as a binary data.

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_MEMORY  | Could not allocate memory.     |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"  
             Lang: EPOS_OC_MODEL_ANK];  
if ( builder != nil ) {  
    int errorStatus = EPOS_OC_SUCCESS;  
    NSData* data = Nil;  
    ///Process///  
    errorStatus = [builder addCommand: data];  
}
```

## init

Initializes an EposPrint class instance.

---

### Syntax

– (id) **init**;

### Return Value

The initialized EposPrint class instance is returned.

### Example

```
id printer = [[EposPrint alloc] init];
if ( printer != nil) {
    ///Process///
    [printer release];
}
```

## openPrinter

This starts communications with the printer and monitoring of printer status.



if communication with the printer is not required anymore, be sure to call [closePrinter \(p.117\)](#), closePrinter API, to end communication with the printer.



- Printer status is notified by the callback method that was registered in the EposPrint class. For details, see [Automatic Acquisition of Printer Status \(p.43\)](#).
- If you want to stop monitoring of printer status, call [closePrinter \(p.117\)](#).
- If you use the printer from multiple mobile terminals, see the [Cautions \(p.196\)](#).

### Syntax

```
- (int) openPrinter: (int) deviceType  
                        DeviceName: (NSString *) deviceName  
                        Enabled: (int) enabled  
                        Interval: (long) interval  
                        Timeout: (long) timeout;
```

### Parameter

- deviceType : Specifies the type for the device to start communication.

| Set value                 | Description           |
|---------------------------|-----------------------|
| EPOS_OC_DEVTYPE_TCP       | Wi-Fi/Ethernet device |
| EPOS_OC_DEVTYPE_BLUETOOTH | Bluetooth device      |

- deviceName : Specifies the identifier used for identification of the target device.  
Specifies the following for each device type:

| deviceType                | Specified Value   |
|---------------------------|---|
| EPOS_OC_DEVTYPE_TCP       | One of the following can be specified. <ul style="list-style-type: none"><li>• IPv4 IP address (Example: "192.168.192.168")</li><li>• MAC address (Example: "01:23:45:67:89:AB")</li><li>• Printer host name (Arbitrary string)</li></ul> |
| EPOS_OC_DEVTYPE_BLUETOOTH | BD address (Example: "01:23:45:67:89:AB")   |



- When a printer's IP address is set as DHCP, specify a MAC address or printer host name for deviceName.
- When Print.EPOS\_OC\_DEVTYPE\_TCP is selected for deviceType, and a printer host name is specified for deviceName, use in an environment in which it is possible to search for a printer host name from the DNS server.

- enabled : This specifies whether printer status monitoring is enabled or disabled.

| Set value             | Specified Value                 |
|-----------------------|---------------------------------|
| EPOS_OC_TRUE          | Enabled                         |
| EPOS_OC_FALSE         | Disabled                        |
| EPOS_OC_PARAM_DEFAULT | Select default value (disabled) |

- interval : This specifies the interval (in units of milliseconds) for updating printer status.

| Set value              | Specified Value   |
|------------------------|---|
| 1000 to 300000 integer | Interval for updating printer status (in units of milliseconds) |
| EPOS_OC_PARAM_DEFAULT  | Specify the default value (1000)                                |

- timeout : This specifies the maximum waiting time (in milliseconds) for establishing communication with the printer.

| Set value             | Specified Value   |
|-----------------------|---|
| 1000 to 60000 integer | Interval for updating printer status (in units of milliseconds) |
| EPOS_OC_PARAM_DEFAULT | Specify the default value (15000)                               |



- If the specified device does not exist, an error is returned immediately.
- When EPOS\_OC\_DEVTYPE\_TCP is specified for deviceType, if the specified device is already used, an attempt is made to execute this API until the timeout time.
- For Bluetooth communication, specify EPOS\_OC\_PARAM\_DEFAULT.

### Return Value

| Error status           | Description   |
|------------------------|---|
| EPOS_OC_SUCCESS        | Processing was successful.  |
| EPOS_OC_ERR_PARAM      | Invalid parameter was passed.   |
| EPOS_OC_ERR_OPEN       | The port open process failed.   |
| EPOS_OC_ERR_TIMEOUT    | The device specified was already being used, and communication with the printer could not be established within the timeout time. |
| EPOS_OC_ERR_ILLEGAL    | An attempt was made to start communicating with the device with which communication had already started.                          |
| EPOS_OC_ERR_PROCESSING | Could not execute process.  |
| EPOS_OC_ERR_MEMORY     | Could not allocate memory.  |
| EPOS_OC_ERR_FAILURE    | An unspecified error occurred.  |

### Example

Case where printer status monitoring is enabled and communications are commenced using Wi-Fi/Ethernet and a printer with an IP address of 192.168.192.168

```
id printer = [[EposPrint alloc] init];
if ( printer != nil) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
        Name:@"192.168.192.168" Enabled: EPOS_OC_TRUE
        Interval:EPOS_OC_PARAM_DEFAULT Timeout:EPOS_OC_PARAM_DEFAULT];
    ///Process///
}
```

## openPrinter(Previous format)

This starts communications with the printer and monitoring of printer status.



if communication with the printer is not required anymore, be sure to call [closePrinter \(p.117\)](#), closePrinter API, to end communication with the printer.



- The timeout time for this API cannot be set. If you want to set the timeout time for this API, use [openPrinter \(p.110\)](#).
- Printer status is notified by the callback method that was registered in the EposPrint class. For details, see [Automatic Acquisition of Printer Status \(p.43\)](#).
- If you want to stop monitoring of printer status, call [closePrinter \(p.117\)](#).
- If another application opened the printer, depending on the connection method, care should be taken about the following:
  - \* TCP connection:  
Retry this API for 15 seconds. After 15 seconds, EPOS\_OC\_ERR\_OPEN will be returned.
  - \* *Bluetooth* connection:  
When an attempt is made to start communication using this API, its result may not be returned.
- If you use the printer from multiple mobile terminals, see the [Cautions \(p.196\)](#).

### Syntax

```
- (int) openPrinter: (int)deviceType  
                    DeviceName: (NSString *)deviceName  
                    Enabled: (int)enabled  
                    Interval: (long)interval;
```

### Parameter

- deviceType : Specifies the type for the device to start communication.

| Set value                 | Description             |
|---------------------------|-------------------------|
| EPOS_OC_DEVTYPE_TCP       | Wi-Fi/Ethernet device   |
| EPOS_OC_DEVTYPE_BLUETOOTH | <i>Bluetooth</i> device |



- **deviceName** : Specifies the identifier used for identification of the target device.  
Specifies the following for each device type:

| deviceType                | Specified Value   |
|---------------------------|---|
| EPOS_OC_DEVTYPE_TCP       | One of the following can be specified. <ul style="list-style-type: none"> <li>• IPv4 IP address (Example: "192.168.192.168")</li> <li>• MAC address (Example: "01:23:45:67:89:AB")</li> <li>• Printer host name (Arbitrary string)</li> </ul> |
| EPOS_OC_DEVTYPE_BLUETOOTH | BD address (Example: "01:23:45:67:89:AB")   |



- When a printer's IP address is set as DHCP, specify a MAC address or printer host name for deviceName.
- When Print.EPOS\_OC\_DEVTYPE\_TCP is selected for deviceType, and a printer host name is specified for deviceName, use in an environment in which it is possible to search for a printer host name from the DNS server.

- **enabled** : This specifies whether printer status monitoring is enabled or disabled.

| Set value             | Specified Value                 |
|-----------------------|---------------------------------|
| EPOS_OC_TRUE          | Enabled                         |
| EPOS_OC_FALSE         | Disabled                        |
| EPOS_OC_PARAM_DEFAULT | Select default value (disabled) |

- **interval** : This specifies the interval (in units of milliseconds) for updating printer status.

| Set value             | Specified Value  |
|-----------------------|--|
| 1000 to 60000 integer | Interval for updating printer status<br>(in units of milliseconds) |
| EPOS_OC_PARAM_DEFAULT | Specify the default value (1000)                                   |

### Return Value

| Error status           | Description  |
|------------------------|--|
| EPOS_OC_SUCCESS        | Processing was successful.   |
| EPOS_OC_ERR_PARAM      | Invalid parameter was passed.  |
| EPOS_OC_ERR_OPEN       | <ul style="list-style-type: none"> <li>• The port open process failed.</li> <li>• The printer was already in use.</li> </ul> |
| EPOS_OC_ERR_ILLEGAL    | An attempt was made to start communicating with the device with which communication had already started.                     |
| EPOS_OC_ERR_PROCESSING | Could not execute process.   |
| EPOS_OC_ERR_MEMORY     | Could not allocate memory.   |
| EPOS_OC_ERR_FAILURE    | An unspecified error occurred.   |

### Example

Case where printer status monitoring is enabled and communications are commenced using Wi-Fi/Ethernet and a printer with an IP address of 192.168.192.168

```
id printer = [[EposPrint alloc] init];
if ( printer != nil) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
        Name:@"192.168.192.168" Enabled: EPOS_OC_TRUE
        Interval:EPOS_OC_PARAM_DEFAULT];
    ///Process///
}
```

## openPrinter(Previous format)

Starts communication with the printer.



if communication with the printer is not required anymore, be sure to call [closePrinter \(p.117\)](#), closePrinter API, to end communication with the printer.



- The timeout time for this API cannot be set. If you want to set the timeout time for this API, use [openPrinter \(p.110\)](#).
- If you want to automatically acquire the printer status, use [openPrinter \(p.110\)](#).
- If another application opened the printer, depending on the connection method, care should be taken about the following:
  - \* TCP connection:  
Retry this API for 15 seconds. After 15 seconds, EPOS\_OC\_ERR\_OPEN will be returned.
  - \* *Bluetooth* connection:  
When an attempt is made to start communication using this API, its result may not be returned.
- If you use the printer from multiple mobile terminals, see the [Cautions \(p.196\)](#).

### Syntax

```
(int) openPrinter: (int) deviceType  
                        DeviceName: (NSString *) deviceName;
```

### Parameter

- deviceType : Specifies the type for the device to start communication.

| Set value                 | Description             |
|---------------------------|-------------------------|
| EPOS_OC_DEVTYPE_TCP       | Wi-Fi/Ethernet device   |
| EPOS_OC_DEVTYPE_BLUETOOTH | <i>Bluetooth</i> device |

- deviceName : Specifies the identifier used for identification of the target device.  
Specifies the following for each device type:

| deviceType                | Specified Value   |
|---------------------------|---|
| EPOS_OC_DEVTYPE_TCP       | One of the following can be specified. <ul style="list-style-type: none"> <li>• IPv4 IP address (Example: "192.168.192.168")</li> <li>• MAC address (Example: "01:23:45:67:89:AB")</li> <li>• Printer host name (Arbitrary string)</li> </ul> |
| EPOS_OC_DEVTYPE_BLUETOOTH | BD address (Example: "01:23:45:67:89:AB")   |



- When a printer's IP address is set as DHCP, specify a MAC address or printer host name for deviceName.
- When Print.EPOS\_OC\_DEVTYPE\_TCP is selected for deviceType, and a printer host name is specified for deviceName, use in an environment in which it is possible to search for a printer host name from the DNS server.

## Return Value

| Error status           | Description   |
|------------------------|---|
| EPOS_OC_SUCCESS        | Processing was successful.  |
| EPOS_OC_ERR_PARAM      | Invalid parameter was passed.   |
| EPOS_OC_ERR_OPEN       | <ul style="list-style-type: none"><li>• The port open process failed.</li><li>• The printer was already in use.</li></ul> |
| EPOS_OC_ERR_ILLEGAL    | An attempt was made to start communicating with the device with which communication had already started.                  |
| EPOS_OC_ERR_PROCESSING | Could not execute process.  |
| EPOS_OC_ERR_MEMORY     | Could not allocate memory.  |
| EPOS_OC_ERR_FAILURE    | An unspecified error occurred.  |

## Example

To start communication via Wi-Fi/Ethernet with the printer whose IP address is "192.168.192.168":

```
id printer = [[EposPrint alloc] init];
if ( printer != nil) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
        DeviceName:@"192.168.192.168"];
    ///Process///
}
```

## closePrinter

This ends communications with the printer and monitoring of printer status.

### Syntax

– (int) **closePrinter**;

### Return Value

| Error status           | Description   |
|------------------------|---|
| EPOS_OC_SUCCESS        | Processing was successful.                                  |
| EPOS_OC_ERR_ILLEGAL    | This API was called when communication had not started yet. |
| EPOS_OC_ERR_PROCESSING | Could not execute process.                                  |
| EPOS_OC_ERR_FAILURE    | An unspecified error occurred.                              |

### Example

```
id printer = [[EposPrint alloc] init];
if ( printer != nil) {
    int errorStatus = EPOS_OC_SUCCESS;
    errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
        DeviceName:@"192.168.192.168"];
    ///Process///
    errorStatus = [printer closePrinter];
}
```

## sendData

Sends a print document created using the EposBuilder class.



- If you are using a *Bluetooth* connection, it may not be able to detect the offline status, and timeout errors may occur.
- If you use the printer from multiple mobile terminals, see the [Cautions \(p.196\)](#).

### Syntax

```
- (int) sendData: (EposBuilder *)builder  
                Timeout: (long) timeout  
                Status: (unsigned long *) status  
                Battery: (unsigned long *) battery;
```

### Parameter

- builder : Specifies an EposBuilder class instance. For details on the EposBuilder class, refer to [EposBuilder class \(p.49\)](#).
- timeout : Specifies the transmission/reception waiting timeout time. Adjust the timeout time according to the specifications for the model, communication interface, and transmission data size.  
Specifies an integer in the range 0-600000 (in milliseconds).
- status : The printer status when command transmission ended is set.  
A combination of printer status settings is set. For details, refer to [Printer Statuses and Actions to Take \(p.47\)](#).
- battery : The battery status when command transmission ended is set.  
For details, refer to [Support Information by Printer \(p.164\)](#).

### Return Value

| Error status           | Description   |
|------------------------|---|
| EPOS_OC_SUCCESS        | Processing was successful.                                  |
| EPOS_OC_ERR_PARAM      | Invalid parameter was passed.                               |
| EPOS_OC_ERR_ILLEGAL    | This API was called when communication had not started yet. |
| EPOS_OC_ERR_PROCESSING | Could not execute process.                                  |
| EPOS_OC_ERR_TIMEOUT    | Could not send all the data within the specified time.      |
| EPOS_OC_ERR_CONNECT    | Connection error occurred                                   |
| EPOS_OC_ERR_MEMORY     | Could not allocate memory.                                  |
| EPOS_OC_ERR_OFF_LINE   | The printer was offline.                                    |
| EPOS_OC_ERR_FAILURE    | An unspecified error occurred.                              |

*Example*

To send a command to the printer by specifying 10 seconds for its timeout parameter:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-P60II"
             Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    unsigned long status = 0;
    unsigned long battery = 0;

    errorStatus = [builder addText:@"ABCDE"];

    id printer = [[EposPrint alloc] init];

    if ( printer != nil ) {
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                         DeviceName:@"192.168.192.168"];
        errorStatus = [printer sendData:builder Timeout:10000
                        Status:&status Battery:&battery];
        errorStatus = [printer closePrinter];
        [printer release];
    }
    [builder release];
}
```

## sendData(Previous format)

Sends a print document created using the EposBuilder class. The battery status cannot be acquired.



- If you are using a *Bluetooth* connection, it may not be able to detect the offline status, and timeout errors may occur.
- The battery status cannot be acquired. If you want to acquire the battery status when sending a print document, use [sendData \(p.118\)](#).
- If you use the printer from multiple mobile terminals, see the [Cautions \(p.196\)](#).

### Syntax

```
- (int) sendData: (EposBuilder *)builder  
                Timeout: (long) timeout  
                Status: (unsigned long *) status;
```

### Parameter

- builder : Specifies an EposBuilder class instance. For details on the EposBuilder class, refer to [EposBuilder class \(p.49\)](#).
- timeout : Specifies the transmission/reception waiting timeout time. Adjust the timeout time according to the specifications for the model, communication interface, and transmission data size.  
Specifies an integer in the range 0-600000 (in milliseconds).
- status : The printer status when command transmission ended is set.  
A combination of printer status settings is set. For details, refer to [Printer Statuses and Actions to Take \(p.47\)](#).

### Return Value

| Error status           | Description   |
|------------------------|---|
| EPOS_OC_SUCCESS        | Processing was successful.                                  |
| EPOS_OC_ERR_PARAM      | Invalid parameter was passed.                               |
| EPOS_OC_ERR_ILLEGAL    | This API was called when communication had not started yet. |
| EPOS_OC_ERR_PROCESSING | Could not execute process.                                  |
| EPOS_OC_ERR_TIMEOUT    | Could not send all the data within the specified time.      |
| EPOS_OC_ERR_CONNECT    | Connection error occurred                                   |
| EPOS_OC_ERR_MEMORY     | Could not allocate memory.                                  |
| EPOS_OC_ERR_OFF_LINE   | The printer was offline.                                    |
| EPOS_OC_ERR_FAILURE    | An unspecified error occurred.                              |



*Example*

To send a command to the printer by specifying 10 seconds for its timeout parameter:

```
id builder = [[EposBuilder alloc] initWithPrinterModel: @"TM-T88V"
    Lang: EPOS_OC_MODEL_ANK];
if ( builder != nil ) {
    int errorStatus = EPOS_OC_SUCCESS;
    unsigned long status = 0;

    errorStatus = [builder addText:@"ABCDE"];

    id printer = [[EposPrint alloc] init];

    if ( printer != nil ) {
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
            DeviceName:@"192.168.192.168"];
        errorStatus = [printer sendData:builder Timeout:10000
            Status:&status];
        errorStatus = [printer closePrinter];
        [printer release];
    }
    [builder release];
}
```

## setStatusChangeEventCallback

This registers the callback method for printer status events.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setStatusChangeEventCallback: (SEL) method  
      Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName  
      Status:(NSNumber *)status;
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of printer status is set.
- status : Printer status is set.

### Example

```
- (void)onStatusChange:(NSString *)deviceName Status:(NSNumber *)status  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setStatusChangeEventCallback @selector(onStatusChange:Status:)  
                                             Target:self];  
  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                             Name:@"192.168.192.168"  
                             Enabled: EPOS_OC_TRUE  
                             Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```

## setOnlineEventCallback

This registers the callback method for online events. This is the notification method when printer status is online.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setOnlineEventCallback: (SEL) method
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of online event is set.

### Example

```
- (void)onOnline:(NSString *)deviceName
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setOnlineEventCallback @selector(onOnline:) Target:self];

        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                Name:@"192.168.192.168"
                                Enabled: EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

## setOfflineEventCallback

This registers the callback method for offline events. This is the notification method when printer is offline.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setOfflineEventCallback: (SEL) method  
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of offline event is set.

### Example

```
- (void)onOffline:(NSString *)deviceName  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setOfflineEventCallback @selector(onOffline:) Target:self];  
  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                        Name:@"192.168.192.168"  
                        Enabled: EPOS_OC_TRUE  
                        Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```

## setPowerOffEventCallback

This registers the callback method for power off events. This is the notification method when there is no response concerning printer status.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setPowerOffEventCallback: (SEL) method
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of power off event is set.

### Example

```
- (void)onPowerOff:(NSString *)deviceName
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setPowerOffEventCallback @selector(onPowerOff:) Target:self];

        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                Name:@"192.168.192.168"
                                Enabled: EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

## setCoverOkEventCallback

This registers the callback method for cover close events. This is the notification method when printer status indicates cover close.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setCoverOkEventCallback: (SEL) method  
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of cover ok event is set.

### Example

```
- (void)onCoverOk:(NSString *)deviceName  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setCoverOkEventCallback @selector(onCoverOk:) Target:self];  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                        Name:@"192.168.192.168"  
                        Enabled: EPOS_OC_TRUE  
                        Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```

## setCoverOpenEventCallback

This registers the callback method for cover open events. This is the notification method when printer status indicates cover open.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setCoverOpenEventCallback: (SEL) method
                                Target: (NSObject*) target;
```

### Parameter

- **method :** This specifies the callback method selector.
- **target :** This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- **deviceName :** The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of cover open event is set.

### Example

```
- (void)onCoverOpen:(NSString *)deviceName
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setCoverOpenEventCallback @selector(onCoverOpen:) Target:self];

        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                Name:@"192.168.192.168"
                                Enabled: EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

## setPaperOkEventCallback

This registers the callback method for paper OK events. This is the notification method when printer status indicates paper ok.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setPaperOkEventCallback: (SEL) method  
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of paper ok event is set.

### Example

```
- (void)onPaperOk:(NSString *)deviceName  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setPaperOkEventCallback @selector(onPaperOk:) Target:self];  
  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                        Name:@"192.168.192.168"  
                        Enabled: EPOS_OC_TRUE  
                        Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```



## setPaperNearEndEventCallback

This registers the callback method for paper near end events. This is the notification method when printer status indicates paper is near the end.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setPaperNearEndEventCallback: (SEL) method
                                Target: (NSObject*) target;
```

### Parameter

- **method :** This specifies the callback method selector.
- **target :** This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- **deviceName :** The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of paper near end event is set.

### Example

```
- (void)onPaperNearEnd:(NSString *)deviceName
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setPaperNearEndEventCallback @selector(onPaperNearEnd:) Target:self];
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                Name:@"192.168.192.168"
                                Enabled: EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

## setPaperEndEventCallback

This registers the callback method for paper end events. This is the notification method when printer status indicates there is no paper.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setPaperEndEventCallback: (SEL) method  
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of paper end event is set.

### Example

```
- (void)onPaperEnd:(NSString *)deviceName  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setPaperEndEventCallback @selector(onPaperEnd:) Target:self];  
  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                        Name:@"192.168.192.168"  
                        Enabled: EPOS_OC_TRUE  
                        Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```

## setDrawerClosedEventCallback

This registers the callback method for drawer closed events. This is the notification method when printer status indicates drawer closed.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setDrawerClosedEventCallback: (SEL) method
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of drawer closed event is set.

### Example

```
- (void)onDrawerClosed:(NSString *)deviceName
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setDrawerClosedEventCallback @selector(onDrawerClosed:) Target:self];

        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                Name:@"192.168.192.168"
                                Enabled: EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

## setDrawerOpenEventCallback

This registers the callback method for drawer open events. This is the notification method when drawer is open concerning printer status.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setDrawerOpenEventCallback: (SEL) method  
Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of drawer open is set.

### Example

```
- (void)onDrawerOpen:(NSString *)deviceName  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setDrawerOpenEventCallback @selector(onDrawerOpen:) Target:self];  
  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                        Name:@"192.168.192.168"  
                        Enabled: EPOS_OC_TRUE  
                        Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```

## setBatteryLowEventCallback

This registers the callback method for a battery low event. This is the notification method when printer status is offline due to battery.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setBatteryLowEventCallback: (SEL) method
                                Target: (NSObject*) target;
```

### Parameter

- **method :** This specifies the callback method selector.
- **target :** This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- **deviceName :** The identifier (IPv4 format IP address / BD address/ Printer host name) of the device that performed the battery low notification is set.

### Example

```
- (void)onBatteryLow:(NSString *)deviceName
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setBatteryLowEventCallback @selector(onBatteryLow:) Target:self];

        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                Name:@"192.168.192.168"
                                Enabled: EPOS_OC_TRUE
                                Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

## setBatteryOkEventCallback

This registers the callback method for a battery OK event. This is the notification method when the printer status recovers from offline due to remaining battery power.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setBatteryOkEventCallback: (SEL) method  
                                Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
```

### Parameter

- deviceName : The identifier (IPv4 format IP address / BD address/ Printer host name) of the device that performed the battery OK event notification is set.

### Example

```
- (void)onBatteryOk:(NSString *)deviceName  
{  
    ///Process///  
}  
  
- (void)openPrinter  
{  
    id printer = [[EposPrint alloc] init];  
  
    if ( printer != nil) {  
        int errorStatus = EPOS_OC_SUCCESS;  
  
        [printer setBatteryOkEventCallback @selector(onBatteryOk:) Target:self];  
  
        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP  
                        Name:@"192.168.192.168"  
                        Enabled: EPOS_OC_TRUE  
                        Interval:EPOS_OC_PARAM_DEFAULT];  
  
        ///Process///  
    }  
}
```

## setBatteryStatusChangeEventCallback

This registers the callback method for battery status events.



- This API can be executed following execution of [openPrinter \(p.110\)](#).
- When this API is executed on multiple occasions, the callback method that is specified afterwards is overwritten.

### Syntax

```
- (void) setBatteryStatusChangeEventCallback:
      (SEL) method Target: (NSObject*) target;
```

### Parameter

- method : This specifies the callback method selector.
- target : This specifies the object that has the callback method.



If null is specified for either the method or target, the callback method is nullified.

### Definition of Callback Method

```
- (void) Method name: (NSString *)deviceName
                        Battery:(NSNumber *)battery;
```

### Parameter

- deviceName : The identifier (IPv4 type IP address/ BD address/ Printer host name) of the device that is notified of battery status is set.
- battery : Battery status is set.

### Example

```
- (void)onBatteryStatusChange:(NSString *)deviceName Battery:(NSNumber *)battery
{
    ///Process///
}

- (void)openPrinter
{
    id printer = [[EposPrint alloc] init];

    if ( printer != nil) {
        int errorStatus = EPOS_OC_SUCCESS;

        [printer setBatteryStatusChangeEventCallback @selector
                                                    (onBatteryStatusChange:Battery:) Target:self];

        errorStatus = [printer openPrinter:EPOS_OC_DEVTYPE_TCP
                                     Name:@"192.168.192.168"
                                     Enabled: EPOS_OC_TRUE
                                     Interval:EPOS_OC_PARAM_DEFAULT];

        ///Process///
    }
}
```

# Printer Search API

API to search for printers. The following classes are available.

- EpsonIoFinder class ([p. 136](#))

## EpsonIoFinder class

Class to search for printers. The following APIs are available.

| API                         | Description                         | Page                   |
|-----------------------------|-------------------------------------|------------------------|
| start                       | Starts searching for printers.      | <a href="#">p. 136</a> |
| stop                        | End communication with the printer. | <a href="#">p. 137</a> |
| getDeviceInfoList           | Getting the printer search result.  | <a href="#">p. 138</a> |
| getResult (Previous format) |                                     | <a href="#">p. 140</a> |

## start

Starts a search for printers of the specified device type.



- If you use this API, be sure to use [stop \(p.137\)](#) to stop the search.
- You cannot call this API when a printer search is already in progress.

## Syntax

```
+ (int) start: (int)deviceType  
                FindOption: (NSString *)findOption;
```

## Parameter

- deviceType : Specifies the device type to search for. The following values can be specified.

| deviceType                   | Description  |
|------------------------------|--|
| EPSONIO_OC_DEVTYPE_TCP       | Searches for TM devices connected to the network   |
| EPSONIO_OC_DEVTYPE_BLUETOOTH | Searches for printers that support ESC/POS and that can be connected to iOS via <i>Bluetooth</i> |

- findOption : Specifies the setting value when searching for a specific target device.

| deviceType                   | Setting Value                       |
|------------------------------|-------------------------------------|
| EPSONIO_OC_DEVTYPE_TCP       | The broadcast address to search for |
| EPSONIO_OC_DEVTYPE_BLUETOOTH | nil                                 |

## Return value

| Return value              | Description   |
|---------------------------|---|
| EPSONIO_OC_SUCCESS        | Processing was successful                                 |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when a search was already in progress |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                                |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed                              |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate memory.                                |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                            |



## stop

Stops the printer search.

---

### Syntax

```
+ (int) stop;
```

*Return value*

| Return value              | Description  |
|---------------------------|--|
| EPSONIO_OC_SUCCESS        | Processing was successful                              |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when a search was not in progress. |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                             |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                         |

## getDeviceInfoList

Acquires the results of device search until this API is called.



This API cannot acquire *Bluetooth* devices that are already open.

### Syntax

```
+ (NSArray *) getDeviceInfoList: (int *)errorStatus  
    FilterOption: (int)filterOption;
```

### Parameter

- `errorStatus` : Returns the error status.

| Return value              | Description  |
|---------------------------|--|
| EPSONIO_OC_SUCCESS        | Processing was successful                              |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when a search was not in progress. |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                             |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed                           |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate memory.                             |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                         |

- `filterOption` : This specifies the filtering method for Epson printers. Specify one of the following values:

| Set value                | Description                 |
|--------------------------|-----------------------------|
| EPSONIO_OC_FILTER_NONE   | Do not filter               |
| EPSONIO_OC_FILTER_NAME   | Filter in the printer name. |
| EPSONIO_OC_PARAM_DEFAULT | Filter in the printer name. |



For iOS device, only Epson printers are searched for regardless of the FilterOption setting.

*Return value*

The device information list (NSArray()) of devices found during search is returned.

Device information is stored in the list as a EpsonIoDeviceInfo-type array.

Information to be stored varies depending on the device type.

| deviceType                   | EpsonIoDeviceInfo | Information to be obtained   |
|------------------------------|-------------------|--|
| EPSONIO_OC_DEVTYPE_TCP       | deviceType        | EPSONIO_OC_DEVTYPE_TCP(Fixed)  |
|                              | printerName       | Printer model name   |
|                              | deviceName        | <ul style="list-style-type: none"> <li>• DHCP disabled: IP address</li> <li>• DHCP enabled: MAC address</li> </ul> |
|                              | ipAddress         | IP Address   |
|                              | macAddress        | MAC Address  |
| EPSONIO_OC_DEVTYPE_BLUETOOTH | deviceType        | EPSONIO_OC_DEVTYPE_BLUETOOTH (Fixed)   |
|                              | printerName       | Bluetooth device name  |
|                              | deviceName        | BD Address<br>(the same format as the MAC address format)  |
|                              | ipAddress         | "" (Empty character)   |
|                              | macAddress        | "" (Empty character)   |

## getResult(Previous format)

Gets the printer search result until the time when this API was called.



This API cannot acquire *Bluetooth* devices that are already open.

### Syntax

```
+ (NSArray *) getResult:(int *)errorStatus;
```

### Parameter

- errorStatus : Returns the error status.

| Return value              | Description  |
|---------------------------|--|
| EPSONIO_OC_SUCCESS        | Processing was successful                              |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when a search was not in progress. |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                             |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed                           |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate memory.                             |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                         |

### Return value

Returns a list of devices found during the search.

Identification information of the found devices is stored as a character string (String type) in the list.

The stored results differ depending on the type of device (deviceType).

| deviceType                   | List to Acquire                                  |
|------------------------------|--|
| EPSONIO_OC_DEVTYPE_TCP       | List of IP addresses of printers                 |
| EPSONIO_OC_DEVTYPE_BLUETOOTH | List of BD addresses of <i>Bluetooth</i> devices |

## Printer Easy Select API

The Printer Easy Select APIs are APIs for selecting a printer using QR Code. They convert data obtained from QR code to a format that can be passed to `openPrinter`.

The following classes are available.

- ❑ `EposEasySelect` class ([p. 141](#))
- ❑ `EposEasySelectInfo` class ([p. 141](#))

---

### ***EposEasySelect* class**

Analyzes QR code data. The following APIs are available.

| API                   | Description                                 | Page                   |
|-----------------------|---|------------------------|
| <code>parseQR</code>  | Analyzes QR code data.                      | <a href="#">p. 141</a> |
| <code>createQR</code> | Creates QR code print data for Easy Select. | <a href="#">p. 142</a> |

---

### ***EposEasySelectInfo* class**

This class stores data analyzed by an `EposEasySelect` class instance and converts it into a variable to be passed to `openPrinter`. The following member variables are available.

| Member Variable          | Description                                      | Page                   |
|--------------------------|--|------------------------|
| <code>deviceType</code>  | Device type in the analysis result               | <a href="#">p. 143</a> |
| <code>printerName</code> | Printer name in the analysis result              | <a href="#">p. 143</a> |
| <code>macAddress</code>  | MAC address or BD address in the analysis result | <a href="#">p. 143</a> |

## **parseQR**

Analyzes QR code string data.

---

### **Syntax**

```
- EposEasySelectInfo* parseQR: (NSString *)data
```

#### *Parameter*

- `data` : Specifies QR code string data.

#### *Return value*

Returns the result of QR code string data analysis. Stores it into an `EposEasySelectInfo` class instance. If analysis fails, returns nil.

## createQR

Creates QR code print data for Easy Select.

### Syntax

```
- (NSString *) createQR: (NSString *)printerName  
                        DeviceType: (int)deviceType  
                        MacAddress (NSString*) :macAddress
```

### Parameter

- printerName : Specifies the printer name.
- deviceType : Specifies the device type. Set either of the following:

| Set value                 | Description            |
|---------------------------|------------------------|
| EPOS_OC_DEVTYPE_TCP       | Wi-Fi/ Ethernet device |
| EPOS_OC_DEVTYPE_BLUETOOTH | Bluetooth device       |

- macAddress : Specifies the BD address.  
BD addresses support the following formats:

| Format            | Description                |
|-------------------|----------------------------|
| 00:11:22:33:44:55 | Separated by a colon ":".  |
| 00-11-22-33-44-55 | Separated by a hyphen "-". |
| 001122334455      | Not separated.             |

### Return value

Returns QR code print data for Easy Select. If print data creation fails, returns nil.

## deviceType

Stores the device type in the analysis result.

| Stored data               | Description  |
|---------------------------|--|
| EPOS_OC_DEVTYPE_TCP       | Wi-Fi/ Ethernet device                             |
| EPOS_OC_DEVTYPE_BLUETOOTH | NFC standard specification (for <i>Bluetooth</i> ) |

### Format

```
int deviceType;
```

## printerName

Stores the printer name in the analysis result.

### Format

```
NSString *printerName;
```

## macAddress

Stores the BD address in the analysis result.

### Format

```
String macAddress;
```

# Log Setting API

Sets the log output. The following class is available.

- EposLog class ([p. 144](#))

## EposLog class

Sets the log output function.

| API            | Description                   | Page                   |
|----------------|-------------------------------|------------------------|
| setLogSettings | Sets the log output function. | <a href="#">p. 144</a> |

## setLogSettings

Sets the log output function.

### Syntax

```
+ (int) setLogSettings: (int) period
                        Enabled: (int) enabled
                        IPAddress: (NSString *) ipAddress
                        Port: (int) port LogSize: (int) logSize
                        LogLevel: (int) logLevel;
```

### Parameter

- period : Specifies the method of setting the log output function.

| Set value             | Description   |
|-----------------------|---|
| EPOS_OC_LOG_TEMPORARY | The settings of this API are disabled when the application is ended.      |
| EPOS_OC_LOG_PERMANENT | The settings of this API are enabled even after the application is ended. |

- enabled : Specifies whether to enable the log output function and the log output destination.

| Set value           | Description                               |
|---------------------|---|
| EPOS_OC_LOG_DISABLE | Disables the log output function.         |
| EPOS_OC_LOG_STORAGE | Outputs log data to the device's storage. |
| EPOS_OC_LOG_TCP     | Outputs log data over TCP.                |



When "enabled" is set to "EPOS\_OC\_LOG\_STORAGE", enable iTunes file sharing. Follow the steps below.

1. Add "UIFileSharingEnabled" to info.plist of the application.  
It is automatically changed to "Application supports iTunes file sharing".
2. Set "Value" of "Application supports iTunes file sharing" to "YES".



- **ipAddress :** Specifies the IPv4 IP address for TCP communication.



If either of the following values is specified for enabled, "nil" can be specified for this parameter.

- \* EPOS\_OC\_LOG\_DISABLE
- \* EPOS\_OC\_LOG\_STORAGE

- **port :** Specifies the port number for TCP communication. Specifies an integer from 0 to 65535.



Even if either of the following values is specified for enabled, specify an integer within the range.

- \* EPOS\_OC\_LOG\_DISABLE
- \* EPOS\_OC\_LOG\_STORAGE

- **logSize :** Specifies the maximum size of log data that is saved on the device's storage. Specifies an integer from 1 to 50 (Unit: MB).



Even if either of the following values is specified for enabled, specify an integer within the range.

- \* EPOS\_OC\_LOG\_DISABLE
- \* EPOS\_OC\_LOG\_TCP

- **logLevel :** Specifies the level of log data to be output.

| Set value       | Description |
|-----------------|-------------|
| EPOS_OC_LOG_LOW | Low level   |

### Return Value

| Error status        | Description                    |
|---------------------|--------------------------------|
| EPOS_OC_SUCCESS     | Processing was successful.     |
| EPOS_OC_ERR_PARAM   | Invalid parameter was passed.  |
| EPOS_OC_ERR_FAILURE | An unspecified error occurred. |

### Example

To output log data to port 8080 (IP address: 192.168.192.168) over TCP:

```
errorStatus = [EposLog setLogSettings: EPOS_OC_LOG_PERMANENT
                Enabled:EPOS_OC_LOG_TCP IPAddress:@"192.168.192.168"
                Port:8080 LogSize:10 LogLevel: EPOS_OC_LOG_LOW ];
///Process///
}
```

To output log data to the device's storage:

```
errorStatus = [EposLog setLogSettings: EPOS_OC_LOG_PERMANENT
                Enabled: EPOS_OC_LOG_STORAGE IPAddress:nil
                Port:0 LogSize:10 LogLevel: EPOS_OC_LOG_LOW ];
///Process///
}
```

To disable the log output function:

```
errorStatus = [EposLog setLogSettings: EPOS_OC_LOG_PERMANENT
                Enabled: EPOS_OC_LOG_DISABLE IPAddress:nil
                Port:0 LogSize:10 LogLevel: EPOS_OC_LOG_LOW ];
///Process///
}
```

---

## How to Extract a Log File

### Save destination

- ❑ Use iTunes and obtain a log file using file sharing.

[http://support.apple.com/kb/HT4094?viewlocale=en\\_US](http://support.apple.com/kb/HT4094?viewlocale=en_US)

### File name

- ❑ EposLog.xx

---

## How to read a log

### Log format

A log record is configured in the following format:

**<< date and time, process ID: thread ID, input and output layer, input and output direction, input and output data >>**

| Item                       | Description   |
|----------------------------|---|
| Date and time              | In yyyy/mm/dd,h:mm:ss.000 format.   |
| Process ID: thread ID      | ID of each process  |
| Input and output layer     | Layer at which data is input and output <ul style="list-style-type: none"><li>• APIIO: Interface layer called by the application</li><li>• IOCM/DEVIO: Layer for communication with devices</li></ul> |
| Input and output direction | Direction in which data is input and output <ul style="list-style-type: none"><li>• -&gt;: Input from a layer</li><li>• &lt;=: Output from a layer</li></ul>  |
| Input and output data      | Called API, parameter, and communication data   |



Each item is separated by a comma (,).

### Output example

To call the addcut method from the application:

```
2014/07/28,20:12:35.836,00002ae9:00006008,APIIO,->,0x687bc5d8,,addCut,1
2014/07/28,20:12:35.836,00002ae9:00006008,APIIO,<-,0x687bc5d8,0,addCut}
```

## Bluetooth Connection API

API for pairing to *Bluetooth* devices.

The following class is available.

- ❑ `EposBluetoothConnection` class ([p. 149](#))

---

### Restrictions for using `EposBluetoothConnection` Class

- ❑ You can use this with iOS Ver.7.0 or later.
- ❑ You can use this with the following printers.
  - TM-P Series

| Model  | Firmware Version      |
|--|-----------------------|
| TM-P60II(Receipt) iOS <i>Bluetooth</i> model | 5.14 ESC/POS or later |
| TM-P60II(Peeler) iOS <i>Bluetooth</i> model  | 6.14 ESC/POS or later |
| TM-P80 iOS <i>Bluetooth</i> model            | 1.02 ESC/POS or later |

The firmware version is printed on the self-test receipt. For details on how to perform self-test printing, refer to the *iOS Bluetooth* TM Printer Technical Reference Guide.

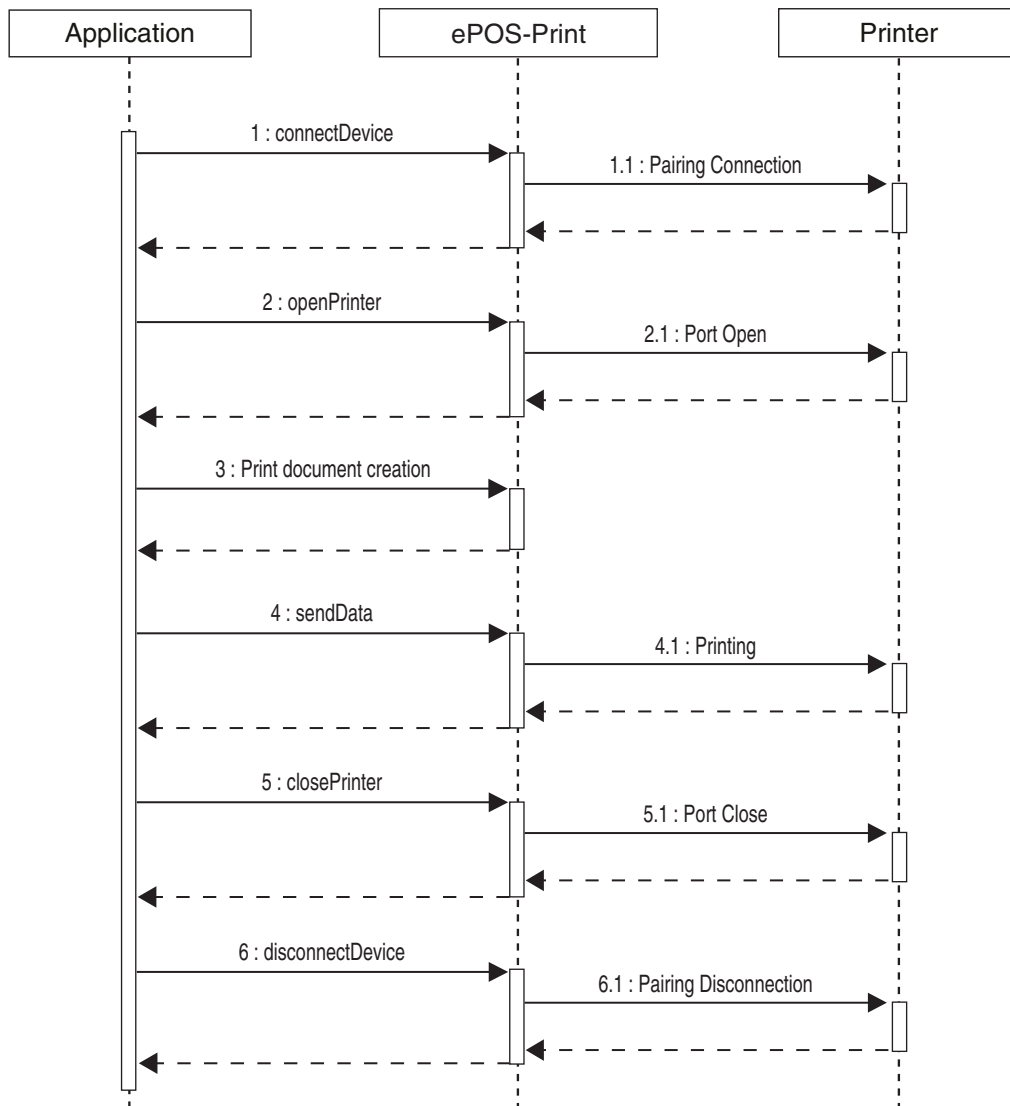
- TM-T Series

| Model                               | Soft Ver      |
|-------------------------------------|---------------|
| TM-T88V iOS <i>Bluetooth</i> model  | 1.05 or later |
| TM-T20II iOS <i>Bluetooth</i> model | 1.05 or later |
| TM-T70II iOS <i>Bluetooth</i> model | 1.05 or later |

The soft ver is printed on the status sheet. For details on how to print the status sheet, refer to the *iOS Bluetooth* TM Printer Technical Reference Guide.

### Using the EposBluetoothConnection Class

Before using openPrinter for the EposPrint class, use the connectDevice to start connection. To disconnect, use disconnectDevice after closePrinter. For the processing flow, refer to the following sequence diagram.



---

**EposBluetoothConnection class**

Connect or disconnect *Bluetooth* pairing. The following APIs are available.

| API              | Description   | Page                   |
|------------------|---|------------------------|
| init             | Initialize an EposBluetoothConnection class instance. | <a href="#">p. 149</a> |
| connectDevice    | Connect <i>Bluetooth</i> pairing.                     | <a href="#">p. 150</a> |
| disconnectDevice | Disconnect <i>Bluetooth</i> pairing.                  | <a href="#">p. 151</a> |

**init**

Initializes an EposBluetoothConnection class instance.

---

**Syntax**

- (id) **init**;

**Return Value**

The initialized EposBluetoothConnection class instance is returned.

## connectDevice

Perform pairing connection.

When this API is performed, *Bluetooth* devices that can be connected using pairing are displayed. When you select a device, pairing connection is performed.



When using with an iOS terminal that has the arm64 architecture, add arm64 to the architecture used for building the application.



Use this API before calling [openPrinter \(p.110\)](#).

### Syntax

```
- (int) connectDevice: (NSMutableString *) macAddress;
```

### Parameter

- macAddress : BD address for the printer that is connected using pairing is set.  
(Ex: 00:11:22:33:44:55)

### Return Value

| Error status                | Description  |
|-----------------------------|--|
| EPOS_BT_SUCCESS             | Processing was successful.                                     |
| EPOS_BT_ERR_PARAM           | Invalid parameter was passed.                                  |
| EPOS_BT_ERR_UNSUPPORTED     | API is performed using the iOS version that is not supported.  |
| EPOS_BT_ERR_CANCEL          | Pairing connection has been canceled.                          |
| EPOS_BT_ERR_ALREADY_CONNECT | The printer that is connected using pairing has been selected. |
| EPOS_BT_ERR_ILLEGAL_DEVICE  | Selected in an illegal device.                                 |
| EPOS_BT_ERR_FAILURE         | An error has occurred due to other factors.                    |

## disconnectDevice

Disconnect pairing connection.



Use this API after calling [closePrinter \(p.117\)](#).

### Syntax

```
- (int) disconnectDevice: (NSString *) macAddress;
```

### Parameter

- `macAddress` : Specifies the printer's BD address for which you want to disconnect pairing. Specify in the following formats.

| Format            | Description                |
|-------------------|----------------------------|
| 00:11:22:33:44:55 | Separated by a colon ":".  |
| 00-11-22-33-44-55 | Separated by a hyphen "-". |
| 001122334455      | Not separated.             |

### Return Value

| Error status            | Description  |
|-------------------------|--|
| EPOS_BT_SUCCESS         | Processing was successful.                           |
| EPOS_BT_ERR_PARAM       | Invalid parameter was passed.                        |
| EPOS_BT_ERR_CONNECT     | Connection error occurred                            |
| EPOS_BT_ERR_MEMORY      | There was not enough memory capacity for processing. |
| EPOS_BT_ERR_ILLEGAL     | API was used in an inappropriate method.             |
| EPOS_BT_ERR_UNSUPPORTED | API was used for the printer that is not supported.  |
| EPOS_BT_ERR_FAILURE     | An error has occurred due to other factors.          |





# Command Transmission/Reception

This chapter describes APIs for transmission and reception of commands (ESC/POS commands, etc.).



The APIs for command transmission and reception described in this chapter are intended for customers who understand ESC/POS commands very well.

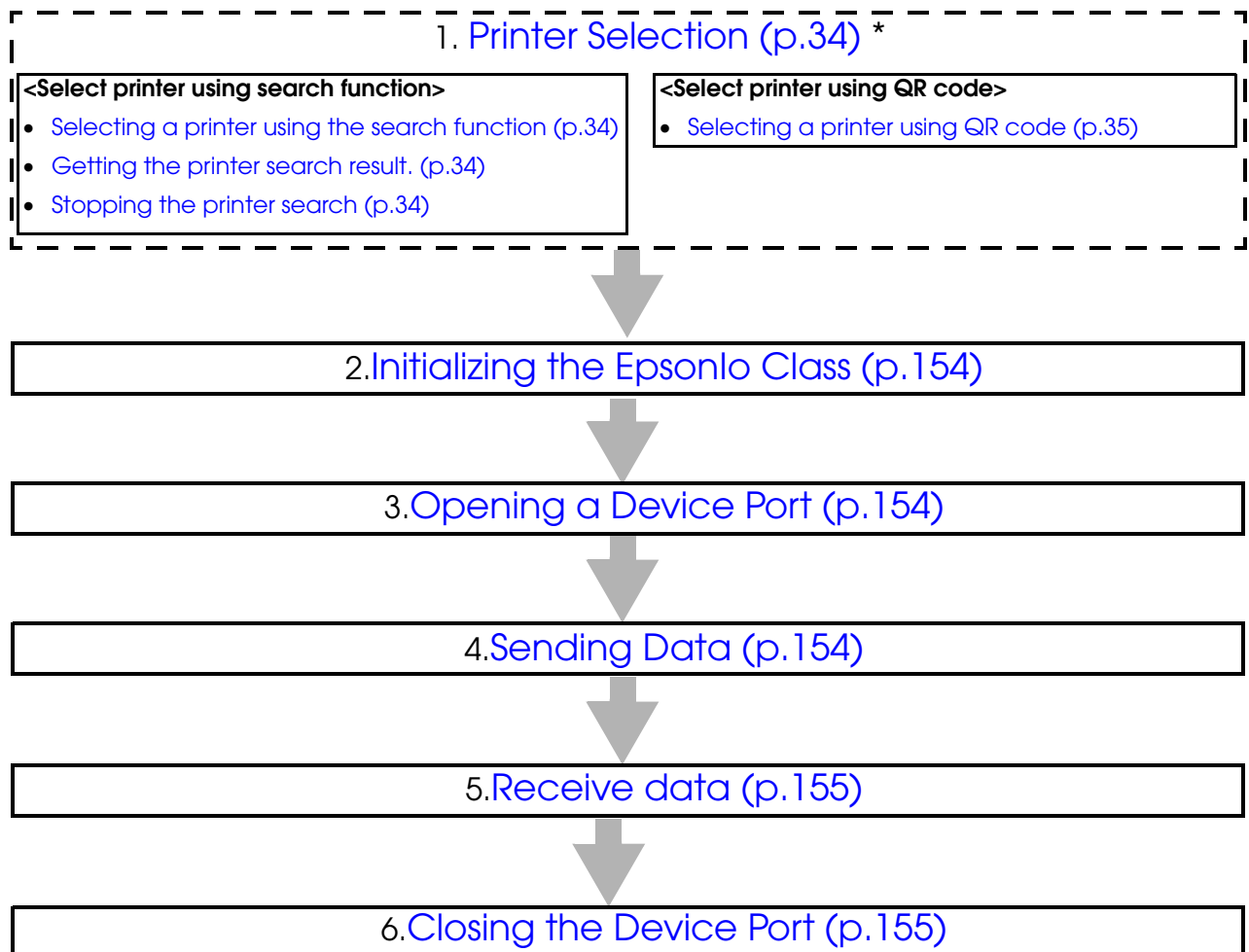


A command transmission/reception API cannot be used with the [EposPrint class \(p.51\)](#) of ePOS-Print API.

## Programming

### Programming Flow

Perform programming following this flow.



\* This is optional.

## Initializing the EpsonIo Class

Use [init \(p.157\)](#) to initialize the EpsonIo class. Please refer to the following code.

```
//Initializing the EpsonIo Class
id port = [[EpsonIo alloc] init];
if ( port != nil ) {
    -Processing-
    [port release];
}
```

## Opening a Device Port

Use the EpsonIo class's [open \(p.158\)](#) to open a device port. Please refer to the following code.

```
//Initializing the EpsonIo Class
id port = [[EpsonIo alloc] init];
if ( port != nil ) {
    int errorStatus = EPSONIO_OC_SUCCESS;
    //Opening a Device Port
    errorStatus = [port open:EPSONIO_OC_DEVTYPE_TCP DeviceName:
        @"192.168.192.168" DeviceSettings:nil];
    if (EPSONIO_OC_SUCCESS == errorStatus ) {
        -Processing-
    }
}
```

## Sending Data

Use the EpsonIo class's [write \(p.160\)](#) to send data to the printer. Please refer to the following code.

Printing out "Hello, World!"

```
//Settings for sending
long sizeWritten;
int errStatus;
NSString *str = @"Hello, World!\r\n";
NSData *data = [str dataUsingEncoding:NSUTF8StringEncoding];

//Sending Data
errStatus = [port write:data Offset:0 Size:[data length]
    Timeout:100 SizeWritten:& sizeWritten];
```

## Receive data

Use the EpsonIo class's [close \(p.159\)](#) to receive data from the printer. Please refer to the following code.

```
//Settings for receiving
NSMutableData *data;
long sizeRead;
int errStatus;
data = [[NSMutableData alloc] initWithLength:256];

//Receive data
errStatus =
[port read:data Offset:0 Size:256 Timeout:100 SizeRead:& sizeRead];
```

## Closing the Device Port

Use the EpsonIo class's [close \(p.159\)](#) to close the device port. Please refer to the following code.

```
//Initializing the EpsonIo Class
id port = [[EpsonIo alloc] init];
if ( port != nil ) {
    int errorStatus = EPSONIO_OC_SUCCESS;
    //Opening a Device Port
    errorStatus = [port open:EPSONIO_OC_DEVTYPE_TCP DeviceName:
@"192.168.192.168" DeviceSettings:nil];
    if (EPSONIO_OC_SUCCESS == errorStatus ) {
        -Processing-
    }
    //Closing the Device Port
    errorStatus = [port close];
}
```

## List of Error Values

Error values are defined in the EpsonIo class.

| Error Value               | Cause   |
|---------------------------|---|
| EPSONIO_OC_SUCCESS        | Processing was successful   |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed.<br><Example> <ul style="list-style-type: none"><li>• An invalid parameter such as Nil was passed.</li><li>• A value outside the supported range was specified.</li></ul>  |
| EPSONIO_OC_ERR_OPEN,      | Open processing failed.<br><Example><br>Failed to create a socket for TCP communication..   |
| EPSONIO_OC_ERR_CONNECT    | Failed to connect to device.<br><Example> <ul style="list-style-type: none"><li>• Failed to send data to the target device for a reason other than a timeout.</li><li>• Failed to receive data from the target device for a reason other than a timeout.</li></ul>        |
| EPSONIO_OC_ERR_TIMEOUT    | Exceeded the specified timeout period.<br><Example> <ul style="list-style-type: none"><li>• Could not send the specified size of data within the specified period.</li><li>• Could not receive even a single byte of data within the specified period.</li></ul>          |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate the necessary memory for processing.   |
| EPSONIO_OC_ERR_ILLEGAL    | Illegal method used.<br><Example> <ul style="list-style-type: none"><li>• The API for sending and receiving data was called when the device port was not open.</li><li>• The printer search API was called again when a printer search was already in progress.</li></ul> |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.<br><Example><br>Could not get lock rights to the shared resource because the same process is currently being executed by another thread.   |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.  |

# Command Transmission/Reception API Reference

The following classes are available for command transmission/reception APIs:

---

## ***Epsonlo class***

Class to transmit and receive data. The following APIs are available.

| API   | Description                           | Page                   |
|-------|---------------------------------------|------------------------|
| init  | Initialize an Epsonlo class instance. | <a href="#">p. 157</a> |
| open  | Opens the device port.                | <a href="#">p. 158</a> |
| close | Closes the device port.               | <a href="#">p. 159</a> |
| write | Send data.                            | <a href="#">p. 160</a> |
| read  | Receive data.                         | <a href="#">p. 162</a> |

### init

Initializes an instance of the Epsonlo class that was created.

---

#### ***Syntax***

- (id) ***init***;

#### ***Return value***

Returns an initialized Epsonlo class instance.

## open

Opens the specified device port.



You can open up to 16 devices ports simultaneously within a single application.

### Syntax

```
- (int) open: (int)deviceType  
                DeviceName: (NSString *)deviceName  
                DeviceSettings: (NSString *)deviceSettings;
```

### Parameter

- deviceType : Specifies the device type to open. The following values can be specified.

| Set value                    | Description   |
|------------------------------|---|
| EPSONIO_OC_DEVTYPE_TCP       | Specify this when the printer to be opened will connect with Wi-Fi/Ethernet.    |
| EPSONIO_OC_DEVTYPE_BLUETOOTH | Specify this when the printer to be opened will connect with <i>Bluetooth</i> . |

- deviceName : Specifies the identifier to locate the target device. The following values can be specified.

| deviceType                   | Setting Value   |
|------------------------------|---|
| EPSONIO_OC_DEVTYPE_TCP       | One of the following can be specified. <ul style="list-style-type: none"><li>IPv4 IP address (Example: "192.168.192.168")</li><li>MAC address (Example: "01:23:45:67:89:AB")</li><li>Printer host name (Arbitrary string)</li></ul> |
| EPSONIO_OC_DEVTYPE_BLUETOOTH | BD address (Example: "01:23:45:67:89:AB")   |

- deviceSettings (Reserved) :  
Specify "nil".

### Return value

| Return value              | Description   |
|---------------------------|---|
| EPSONIO_OC_SUCCESS        | Processing was successful                             |
| EPSONIO_OC_ERR_OPEN       | Open processing failed.                               |
| EPSONIO_OC_ERR_ILLEGAL    | User attempted to open a device that is already open. |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                            |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed.                         |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate memory.                            |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                        |

## close

Closes the specified device port.

---

### Syntax

– (int) **close**;

*Return value*

| Return value              | Description                                       |
|---------------------------|---|
| EPSONIO_OC_SUCCESS        | Processing was successful                         |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when no device port was open. |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                        |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                    |

## write



If *Bluetooth* is used to send data with this API, do not use the [close \(p.159\)](#) until the printer has finished printing. Otherwise, the data sending process will be interrupted.

Sends data to a device port.

### Syntax

```
- (int) write: (NSData *)data  
             Offset: (size_t)offset  
             Size: (size_t)size  
             Timeout: (long)timeout  
             SizeWritten: (size_t *)sizeWritten;
```

### Parameter

- data : The sending data buffer. It stores data to be sent.
- offset : Specifies the start position for sending data.  
Please specify the offset value from the top of the sending data buffer.
- size : Specifies the number of bytes to send.



If "0" is specified for size, no data will be sent.  
In such a case, "0" is returned for sizeWritten.

- timeout : Specifies the time in milliseconds to wait for sending to complete.  
The maximum value that can be specified is 600000 (which equates to 10 minutes).



- Take the transmission speed and volume of data to be sent into account when specifying the timeout value.
- When the timeout value is short, the sending process will still continue until all the data has been sent, while normal data sending is occurring, even if the timeout value is exceeded.

- sizeWritten : Stores the number of bytes of data that were sent.



- The printer did not necessarily receive the amount of data that sizeWritten returns.
- If the amount of time specified in timeout is exceeded, the number of bytes that were sent up to that point is stored in sizeWritten.



*Return value*

| Return value              | Description                                       |
|---------------------------|---|
| EPSONIO_OC_SUCCESS        | Processing was successful                         |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when no device port was open. |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                        |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed.                     |
| EPSONIO_OC_ERR_TIMEOUT    | Could not send all data within specified period   |
| EPSONIO_OC_ERR_CONNECT    | Connection error occurred                         |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate memory.                        |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                    |

## read

Receives data from a device port.



This API continues receiving until a receiving error occurs.  
However, if not even a single byte of data is received during the period specified in timeout, the process ends.

### Syntax

```
- (int) read: (NSMutableData *)data  
             Offset: (size_t)offset  
             Size: (size_t)size  
             Timeout: (long)timeout  
             SizeRead: (size_t *)sizeRead;
```

### Parameter

- data : The receiving data buffer for storing received data.
- offset : Specifies the point to start storing data in the receiving data buffer.  
Please specify the offset value from the top of the receiving data buffer.
- size : Specifies the number of bytes that can be received.



If "0" is specified for size, no data will be received.  
In such a case, "0" is returned for sizeRead.

- timeout : Specifies the time in milliseconds to receive data.  
The maximum value that can be specified is 600000 (which equates to 10 minutes).
- sizeRead : Returns the number of bytes that were received.

### Return value

| Return value              | Description  |
|---------------------------|--|
| EPSONIO_OC_SUCCESS        | Processing was successful                          |
| EPSONIO_OC_ERR_ILLEGAL    | This API was called when no device port was open.  |
| EPSONIO_OC_ERR_PROCESSING | Could not execute process.                         |
| EPSONIO_OC_ERR_PARAM      | Invalid parameter was passed.                      |
| EPSONIO_OC_ERR_TIMEOUT    | Could not receive any data within specified period |
| EPSONIO_OC_ERR_CONNECT    | Connection error occurred                          |
| EPSONIO_OC_ERR_MEMORY     | Could not allocate memory.                         |
| EPSONIO_OC_ERR_FAILURE    | Unspecified error encountered.                     |

# Appendix

## List of Supported APIs for Each Printer Model

| API   | TM-P20 | TM-P60 | TM-P60(Peeler) | TM-P60II | TM-P60II(Peeler) | TM-P80 | TM-T20 | TM-T20II | TM-T70 | TM-T70II | TM-T81II | TM-T82 | TM-T82II | TM-T88V | TM-T90II | TM-U220 | TM-U330 |
|---|--------|--------|----------------|----------|------------------|--------|--------|----------|--------|----------|----------|--------|----------|---------|----------|---------|---------|
| <a href="#">addTextAlign (p.55)</a>               | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addTextLineSpace (p.56)</a>           | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addTextRotate (p.57)</a>              | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addText (p.58)</a>                    | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addTextLang (p.59)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addTextFont (p.60)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addTextSmooth (p.61)</a>              | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addTextDouble (p.62)</a>              | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addTextSize (p.63)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addTextStyle (p.64)</a>               | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addTextPosition (p.66)</a>            | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | ✓       |
| <a href="#">addFeedUnit (p.67)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addFeedLine (p.68)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addImage (p.69)</a>                   | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addImage(Previous format) (p.72)</a>  | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addImage(Previous format) (p.75)</a>  | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addLogo (p.77)</a>                    | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addBarcode (p.78)</a>                 | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addSymbol (p.84)</a>                  | ✓      | -      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addPageBegin (p.89)</a>               | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addPageEnd (p.90)</a>                 | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addPageArea (p.91)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addPageDirection (p.92)</a>           | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addPagePosition (p.94)</a>            | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | -       | -       |
| <a href="#">addPageLine (p.95)</a>                | ✓      | -      | ✓              | ✓        | ✓                | ✓      | -      | -        | -      | -        | -        | -      | -        | -       | -        | -       | -       |
| <a href="#">addPageRectangle (p.97)</a>           | ✓      | -      | ✓              | ✓        | ✓                | ✓      | -      | -        | -      | -        | -        | -      | -        | -       | -        | -       | -       |
| <a href="#">addCut (p.99)</a>                     | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addPulse (p.100)</a>                  | -      | -      | -              | -        | -                | -      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |
| <a href="#">addSound (p.101)</a>                  | ✓      | -      | -              | ✓        | ✓                | ✓      | -      | ✓        | -      | ✓        | -        | ✓      | ✓        | ✓       | -        | -       | ✓       |
| <a href="#">addSound(Previous format) (p.103)</a> | ✓      | -      | -              | ✓        | ✓                | ✓      | -      | ✓        | -      | ✓        | -        | ✓      | ✓        | ✓       | -        | -       | ✓       |
| <a href="#">addFeedPosition (p.105)</a>           | ✓      | -      | ✓              | -        | ✓                | ✓      | -      | -        | -      | -        | -        | -      | -        | -       | -        | -       | ✓       |
| <a href="#">addLayout (p.106)</a>                 | ✓      | -      | ✓              | -        | ✓                | ✓      | -      | -        | -      | -        | -        | -      | -        | -       | -        | -       | -       |
| <a href="#">addCommand (p.108)</a>                | ✓      | ✓      | ✓              | ✓        | ✓                | ✓      | ✓      | ✓        | ✓      | ✓        | ✓        | ✓      | ✓        | ✓       | ✓        | ✓       | ✓       |

## Support Information by Printer

### TM-P20

|                         |        | 58 mm   |
|-------------------------|--------|---|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• South Asian model</li> </ul> |
| Print Width             |        | 384 dots  |
| Characters in a Line    | Font A | ANK: 32 characters  |
|                         | Font B | ANK: 42 characters  |
|                         | Font C | ANK: 42 characters  |
|                         | Font D | ANK: 38 characters  |
|                         | Font E | ANK: 48 characters  |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |
|                         | Font B | ANK: 9 dots x 24 dots (W x H)   |
|                         | Font C | ANK: 9 dots x 17 dots (W x H)   |
|                         | Font D | ANK: 10 dots x 24 dots (W x H)  |
|                         | Font E | ANK: 8 dots x 16 dots (W x H)   |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |
|                         | Font B | At the 21st dot from the top of the character   |
|                         | Font C | At the 16th dot from the top of the character   |
|                         | Font D | At the 21st dot from the top of the character   |
|                         | Font E | At the 15th dot from the top of the character   |
| Default Line Feed Space |        | 30 dots   |
| Color Specification     |        | First color   |
| Page Mode Default Area  |        | 384 dots x 2400 dots (W x H)  |
| Page Mode Maximum Area  |        | 384 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         |

|                      | 58 mm  |
|----------------------|--|
| Two-Dimensional Code | PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology |
| Paper Cut            | Feeds paper to cutting position  |
| Drawer Kick-Out      | Not supported  |
| Buzzer               | Optional   |
| Battery              | Supported  |

### **Battery Status**

#### *Upper 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | The AC adapter is connected |
| 0x31           | The AC adapter is connected |

#### *Lower 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | Battery amount 0 (real end) |
| 0x31           | Battery amount 1 (near end) |
| 0x32           | Battery amount 2            |
| 0x33           | Battery amount 3            |
| 0x34           | Battery amount 4            |
| 0x35           | Battery amount 5            |
| 0x36           | Battery amount 6            |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P20 iOS *Bluetooth* model

|                         |        | 58 mm   |
|-------------------------|--------|---|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• South Asian model</li> </ul> |
| Print Width             |        | 384 dots  |
| Characters in a Line    | Font A | ANK: 32 characters  |
|                         | Font B | ANK: 42 characters  |
|                         | Font C | ANK: 42 characters  |
|                         | Font D | ANK: 38 characters  |
|                         | Font E | ANK: 48 characters  |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |
|                         | Font B | ANK: 9 dots x 24 dots (W x H)   |
|                         | Font C | ANK: 9 dots x 17 dots (W x H)   |
|                         | Font D | ANK: 10 dots x 24 dots (W x H)  |
|                         | Font E | ANK: 8 dots x 16 dots (W x H)   |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |
|                         | Font B | At the 21st dot from the top of the character   |
|                         | Font C | At the 16th dot from the top of the character   |
|                         | Font D | At the 21st dot from the top of the character   |
|                         | Font E | At the 15th dot from the top of the character   |
| Default Line Feed Space |        | 30 dots   |
| Color Specification     |        | First color   |
| Page Mode Default Area  |        | 384 dots x 2400 dots (W x H)  |
| Page Mode Maximum Area  |        | 384 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, Composite Symbology  |

|                 | 58 mm                           |
|-----------------|---------------------------------|
| Paper Cut       | Feeds paper to cutting position |
| Drawer Kick-Out | Not supported                   |
| Buzzer          | Optional                        |
| Battery         | Supported                       |

### **Battery Status**

*Upper 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | The AC adapter is connected |
| 0x31           | The AC adapter is connected |

*Lower 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | Battery amount 0 (real end) |
| 0x31           | Battery amount 1 (near end) |
| 0x32           | Battery amount 2            |
| 0x33           | Battery amount 3            |
| 0x34           | Battery amount 4            |
| 0x35           | Battery amount 5            |
| 0x36           | Battery amount 6            |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P60

|                         |        | 58 mm   | 60 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Language                |        | ANK model   |                              |
| Print Width             |        | 420 dots  | 432 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 36 characters           |
|                         | Font B | ANK: 42 characters  | ANK: 43 characters           |
|                         | Font C | ANK: 52 characters  | ANK: 54 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 10 dots x 24 dots (W x H)  |                              |
|                         | Font C | ANK: 8 dots x 16 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st dot from the top of the character                                 |                              |
|                         | Font B | At the 21st dot from the top of the character                                 |                              |
|                         | Font C | At the 15th dot from the top of the character                                 |                              |
| Default Line Feed Space |        | 30 dots   |                              |
| Color Specification     |        | First color   |                              |
| Page Mode Default Area  |        | 420 dots x 1200 dots (W x H)  | 432 dots x 1200 dots (W x H) |
| Page Mode Maximum Area  |        | 420 dots x 1200 dots (W x H)  | 432 dots x 1200 dots (W x H) |
| Barcode                 |        | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128 |                              |
| Two-Dimensional Code    |        | Not supported   |                              |
| Paper Cut               |        | Cut, No cut   |                              |
| Drawer Kick-Out         |        | Not supported   |                              |
| Buzzer                  |        | Supported   |                              |
| Battery                 |        | Supported   |                              |



**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                 |
|----------------|-----------------------|
| 0x30           | H level               |
| 0x31           | M level               |
| 0x32           | L level               |
| 0x33           | S level               |
| 0x34           | Battery not installed |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P60[Peeler] iOS *Bluetooth* model

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Print Width             |        | 420 dots  | 432 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 36 characters           |
|                         | Font B | ANK: 42 characters  | ANK: 43 characters           |
|                         | Font C | ANK: 52 characters  | ANK: 54 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 10 dots x 24 dots (W x H)  |                              |
|                         | Font C | ANK: 8 dots x 16 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |                              |
|                         | Font B | At the 21st dot from the top of the character   |                              |
|                         | Font C | At the 15th dot from the top of the character   |                              |
| Default Line Feed Space |        | 30 dots   |                              |
| Color Specification     |        | First color   |                              |
| Page Mode Default Area  |        | 420 dots x 1200 dots (W x H)  | 432 dots x 1200 dots (W x H) |
| Page Mode Maximum Area  |        | 420 dots x 1200 dots (W x H)  | 432 dots x 1200 dots (W x H) |
| Barcode                 |        | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128   |                              |
| Two-Dimensional Code    |        | PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composite Symbology not supported) |                              |
| Paper Cut               |        | Cut, No cut   |                              |
| Drawer Kick-Out         |        | Not supported   |                              |
| Buzzer                  |        | Supported   |                              |
| Battery                 |        | Supported   |                              |

**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                 |
|----------------|-----------------------|
| 0x30           | H level               |
| 0x31           | M level               |
| 0x32           | L level               |
| 0x33           | S level               |
| 0x34           | Battery not installed |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P60[Receipt] iOS *Bluetooth* model

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Print Width             |        | 420 dots  | 432 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 36 characters           |
|                         | Font B | ANK: 42 characters  | ANK: 43 characters           |
|                         | Font C | ANK: 52 characters  | ANK: 54 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 10 dots x 24 dots (W x H)  |                              |
|                         | Font C | ANK: 8 dots x 16 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st dot from the top of the character                                 |                              |
|                         | Font B | At the 21st dot from the top of the character                                 |                              |
|                         | Font C | At the 15th dot from the top of the character                                 |                              |
| Default Line Feed Space |        | 30 dots   |                              |
| Color Specification     |        | First color   |                              |
| Page Mode Default Area  |        | 420 dots x 1200 dots (W x H)  | 432 dots x 1200 dots (W x H) |
| Page Mode Maximum Area  |        | 420 dots x 1200 dots (W x H)  | 432 dots x 1200 dots (W x H) |
| Barcode                 |        | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128 |                              |
| Two-Dimensional Code    |        | Not supported   |                              |
| Paper Cut               |        | Cut, No cut   |                              |
| Drawer Kick-Out         |        | Not supported   |                              |
| Buzzer                  |        | Supported   |                              |
| Battery                 |        | Supported   |                              |

**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                 |
|----------------|-----------------------|
| 0x30           | H level               |
| 0x31           | M level               |
| 0x32           | L level               |
| 0x33           | S level               |
| 0x34           | Battery not installed |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P60II

|                         |             | 58 mm  | 60 mm                        |
|-------------------------|-------------|--|------------------------------|
| Resolution              |             | 203 dpi x 203 dpi (W x H)  |                              |
| Language                |             | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Traditional Chinese model</li> </ul>   |                              |
| Print Width             |             | 420 dots   | 432 dots                     |
| Characters in a Line    | Font A      | ANK: 35 characters   | ANK: 36 characters           |
|                         | Font B      | ANK: 42 characters   | ANK: 43 characters           |
|                         | Font C      | ANK: 52 characters   | ANK: 54 characters           |
|                         | Kanji font* | ANK: 17 characters   | ANK: 18 characters           |
| Character Size          | Font A      | ANK: 12 dots x 24 dots (W x H)   |                              |
|                         | Font B      | ANK: 10 dots x 24 dots (W x H)   |                              |
|                         | Font C      | ANK: 8 dots x 16 dots (W x H)  |                              |
|                         | Kanji font* | ANK: 24 dots x 24 dots (W x H)   |                              |
| Character Baseline      | Font A      | At the 21st dot from the top of the character  |                              |
|                         | Font B      | At the 21st dot from the top of the character  |                              |
|                         | Font C      | At the 15th dot from the top of the character  |                              |
|                         | Kanji font* | At the 21st dot from the top of the character  |                              |
| Default Line Feed Space |             | 30 dots  |                              |
| Color Specification     |             | First color  |                              |
| Page Mode Default Area  |             | 420 dots x 1624 dots (W x H)   | 432 dots x 1624 dots (W x H) |
| Page Mode Maximum Area  |             | 420 dots x 1624 dots (W x H)   | 432 dots x 1624 dots (W x H) |
| Barcode                 |             | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39,<br>ITF, CODABAR, CODE93, CODE128, GS1-128,<br>GS1 DataBar Omnidirectional, GS1 DataBar Truncated,<br>GS1 DataBar Limited, GS1 DataBar Expanded |                              |
| Two-Dimensional Code    |             | PDF417, QR Code, MaxiCode, GS1 DataBar Stacked,<br>GS1 DataBar Stacked Omnidirectional,<br>GS1 DataBar Expanded Stacked, Composite Symbology   |                              |
| Paper Cut               |             | Cut, Feed cut  |                              |
| Drawer Kick-Out         |             | Not supported  |                              |
| Buzzer                  |             | Optional   |                              |
| Battery                 |             | Supported  |                              |

\* Only for Traditional Chinese model

**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | Battery amount 0 (real end) |
| 0x31           | Battery amount 1 (near end) |
| 0x32           | Battery amount 2            |
| 0x33           | Battery amount 3            |
| 0x34           | Battery amount 4            |
| 0x35           | Battery amount 5            |
| 0x36           | Battery amount 6            |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P60II iOS *Bluetooth* model

|                         |             | 58 mm  | 60 mm                        |
|-------------------------|-------------|--|------------------------------|
| Resolution              |             | 203 dpi x 203 dpi (W x H)  |                              |
| Language                |             | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Traditional Chinese model</li> </ul>   |                              |
| Print Width             |             | 420 dots   | 432 dots                     |
| Characters in a Line    | Font A      | ANK: 35 characters   | ANK: 36 characters           |
|                         | Font B      | ANK: 42 characters   | ANK: 43 characters           |
|                         | Font C      | ANK: 52 characters   | ANK: 54 characters           |
|                         | Kanji font* | ANK: 17 characters   | ANK: 18 characters           |
| Character Size          | Font A      | ANK: 12 dots x 24 dots (W x H)   |                              |
|                         | Font B      | ANK: 10 dots x 24 dots (W x H)   |                              |
|                         | Font C      | ANK: 8 dots x 16 dots (W x H)  |                              |
|                         | Kanji font* | ANK: 24 dots x 24 dots (W x H)   |                              |
| Character Baseline      | Font A      | At the 21st dot from the top of the character  |                              |
|                         | Font B      | At the 21st dot from the top of the character  |                              |
|                         | Font C      | At the 15th dot from the top of the character  |                              |
|                         | Kanji font* | At the 21st dot from the top of the character  |                              |
| Default Line Feed Space |             | 30 dots  |                              |
| Color Specification     |             | First color  |                              |
| Page Mode Default Area  |             | 420 dots x 1624 dots (W x H)   | 432 dots x 1624 dots (W x H) |
| Page Mode Maximum Area  |             | 420 dots x 1624 dots (W x H)   | 432 dots x 1624 dots (W x H) |
| Barcode                 |             | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39,<br>ITF, CODABAR, CODE93, CODE128, GS1-128,<br>GS1 DataBar Omnidirectional, GS1 DataBar Truncated,<br>GS1 DataBar Limited, GS1 DataBar Expanded |                              |
| Two-Dimensional Code    |             | PDF417, QR Code, MaxiCode, GS1 DataBar Stacked,<br>GS1 DataBar Stacked Omnidirectional,<br>GS1 DataBar Expanded Stacked, Composite Symbology   |                              |
| Paper Cut               |             | Cut, Feed cut  |                              |
| Drawer Kick-Out         |             | Not supported  |                              |
| Buzzer                  |             | Optional   |                              |
| Battery                 |             | Supported  |                              |

\* Only for Traditional Chinese model



**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | Battery amount 0 (real end) |
| 0x31           | Battery amount 1 (near end) |
| 0x32           | Battery amount 2            |
| 0x33           | Battery amount 3            |
| 0x34           | Battery amount 4            |
| 0x35           | Battery amount 5            |
| 0x36           | Battery amount 6            |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P80

|                         |             | 80 mm   |
|-------------------------|-------------|---|
| Resolution              |             | 203 dpi x 203 dpi (W x H)   |
| Language                |             | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Traditional Chinese model</li> </ul>  |
| Print Width             |             | 576 dots  |
| Characters in a Line    | Font A      | ANK: 48 characters  |
|                         | Font B      | ANK: 64 characters  |
|                         | Kanji font* | ANK: 21 characters  |
| Character Size          | Font A      | ANK: 12 dots x 24 dots (W x H)  |
|                         | Font B      | ANK: 9 dots x 17 dots (W x H)   |
|                         | Kanji font* | ANK: 24 dots x 24 dots (W x H)  |
| Character Baseline      | Font A      | At the 21st dot from the top of the character   |
|                         | Font B      | At the 16th dot from the top of the character   |
|                         | Kanji font* | At the 21st dot from the top of the character   |
| Default Line Feed Space |             | 30 dots   |
| Color Specification     |             | First color   |
| Page Mode Default Area  |             | 576 dots x 1662 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, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded |
| Two-Dimensional Code    |             | PDF417, QR Code, MaxiCode, Data Matrix, Aztec Code, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology                       |
| Paper Cut               |             | Feeds paper to cutting position   |
| Drawer Kick-Out         |             | Not supported   |
| Buzzer                  |             | Supported   |
| Battery                 |             | Supported   |

\* Only for Traditional Chinese model

**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | Battery amount 0 (real end) |
| 0x31           | Battery amount 1 (near end) |
| 0x32           | Battery amount 2            |
| 0x33           | Battery amount 3            |
| 0x34           | Battery amount 4            |
| 0x35           | Battery amount 5            |
| 0x36           | Battery amount 6            |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-P80 iOS *Bluetooth* model

|                         |             | 80 mm   |
|-------------------------|-------------|---|
| Resolution              |             | 203 dpi x 203 dpi (W x H)   |
| Language                |             | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Traditional Chinese model</li> </ul>  |
| Print Width             |             | 576 dots  |
| Characters in a Line    | Font A      | ANK: 48 characters  |
|                         | Font B      | ANK: 64 characters  |
|                         | Kanji font* | ANK: 21 characters  |
| Character Size          | Font A      | ANK: 12 dots x 24 dots (W x H)  |
|                         | Font B      | ANK: 9 dots x 17 dots (W x H)   |
|                         | Kanji font* | ANK: 24 dots x 24 dots (W x H)  |
| Character Baseline      | Font A      | At the 21st dot from the top of the character   |
|                         | Font B      | At the 16th dot from the top of the character   |
|                         | Kanji font* | At the 21st dot from the top of the character   |
| Default Line Feed Space |             | 30 dots   |
| Color Specification     |             | First color   |
| Page Mode Default Area  |             | 576 dots x 1662 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, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded |
| Two-Dimensional Code    |             | PDF417, QR Code, MaxiCode, Data Matrix, Aztec Code, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked, Composite Symbology                       |
| Paper Cut               |             | Feeds paper to cutting position   |
| Drawer Kick-Out         |             | Not supported   |
| Buzzer                  |             | Supported   |
| Battery                 |             | Supported   |

\* Only for Traditional Chinese model

**Battery Status***Upper 8 bits*

| Battery Status | Cause                           |
|----------------|---------------------------------|
| 0x30           | The AC adapter is connected     |
| 0x31           | The AC adapter is not connected |

*Lower 8 bits*

| Battery Status | Cause                       |
|----------------|-----------------------------|
| 0x30           | Battery amount 0 (real end) |
| 0x31           | Battery amount 1 (near end) |
| 0x32           | Battery amount 2            |
| 0x33           | Battery amount 3            |
| 0x34           | Battery amount 4            |
| 0x35           | Battery amount 5            |
| 0x36           | Battery amount 6            |



If 0x0000 is returned, the battery status cannot be acquired.

## TM-T20

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> </ul>   |                              |
| Print Width             |        | 420 dots  | 576 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 48 characters           |
|                         | Font B | ANK: 46 characters  | ANK: 64 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 420 dots x 831 dots (W x H)   | 576 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 420 dots x 1662 dots (W x H)  | 576 dots x 1662 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, Composite Symbology  |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Not supported   |                              |
| Battery                 |        | Not supported   |                              |

**TM-T20II**

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Language                |        | ANK model   |                              |
| Print Width             |        | 420 dots  | 576 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 48 characters           |
|                         | Font B | ANK: 46 characters  | ANK: 64 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 420 dots x 831 dots (W x H)   | 576 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 420 dots x 1662 dots (W x H)  | 576 dots x 1662 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, Composite Symbolology  |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Optional  |                              |
| Battery                 |        | Not supported   |                              |

## TM-T20II iOS *Bluetooth* model

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Language                |        | ANK model   |                              |
| Print Width             |        | 420 dots  | 576 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 48 characters           |
|                         | Font B | ANK: 46 characters  | ANK: 64 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 420 dots x 831 dots (W x H)   | 576 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 420 dots x 1662 dots (W x H)  | 576 dots x 1662 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, Composite Symbolology  |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Optional  |                              |
| Battery                 |        | Not supported   |                              |



**TM-T70**

|                         |        | 80 mm   |
|-------------------------|--------|---|
| Resolution              |        | 180 dpi x 180 dpi (W x H)   |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• South Asian model</li> </ul> |
| Print Width             |        | 512 dots  |
| Characters in a Line    | Font A | ANK: 42 characters  |
|                         | Font B | ANK: 56 characters  |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |
|                         | Font B | At the 15th dot from the top of the character   |
| Default Line Feed Space |        | 30 dots   |
| Color Specification     |        | First color   |
| Page Mode Default Area  |        | 512 dots x 1662 dots (W x H)  |
| Page Mode Maximum Area  |        | 512 dots x 1662 dots (W x H)  |
| Barcode                 |        | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128   |
| Two-Dimensional Code    |        | QR Code   |
| Paper Cut               |        | Cut, Feed cut   |
| Drawer Kick-Out         |        | Supported   |
| Buzzer                  |        | Not supported   |
| Battery                 |        | Not supported   |

## TM-T70II

|                         |        | 58 mm   | 80 mm  |
|-------------------------|--------|---|--|
| Resolution              |        | ANK: 180 dpi x 180 dpi (W x H)<br>Other: 203 dpi x 203 dpi (W x H)  |  |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• Korean model</li> <li>• South Asian model</li> </ul> |  |
| Print Width             |        | ANK: 360 dots<br>Other: 416 dots  | ANK: 512 dots<br>Other: 576 dots   |
| Characters in a Line    | Font A | ANK: 30 characters<br>Other: 34 characters  | ANK: 42 characters<br>Other: 48 characters                               |
|                         | Font B | ANK: 40 characters<br>Other: 52 characters  | ANK: 56 characters<br>Other: 72 characters                               |
| Character Size          | Font A | 12 dots x 24 dots (W x H)   |  |
|                         | Font B | 9 dots x 17 dots (W x H)  |  |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |  |
|                         | Font B | At the 15th dot from the top of the character   |  |
| Default Line Feed Space |        | 30 dots   |  |
| Color Specification     |        | First color   |  |
| Page Mode Default Area  |        | ANK: 360 dots x 1662 dots (W x H)<br>Other: 416 dots x 1662 dots (W x H)  | ANK: 512 dots x 1662 dots (W x H)<br>Other: 576 dots x 1662 dots (W x H) |
| Page Mode Maximum Area  |        | ANK: 360 dots x 1662 dots (W x H)<br>Other: 416 dots x 1662 dots (W x H)  | ANK: 512 dots x 1662 dots (W x H)<br>Other: 576 dots x 1662 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   |  |
| Paper Cut               |        | Cut, Feed cut   |  |
| Drawer Kick-Out         |        | Supported   |  |
| Buzzer                  |        | Supported   |  |
| Battery                 |        | Not supported   |  |

## TM-T70II iOS *Bluetooth* model

|                         |        | 58 mm   | 80 mm  |
|-------------------------|--------|---|--|
| Resolution              |        | ANK: 180 dpi x 180 dpi (W x H)<br>Other: 203 dpi x 203 dpi (W x H)  |  |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• Korean model</li> <li>• South Asian model</li> </ul> |  |
| Print Width             |        | ANK: 360 dots<br>Other: 416 dots  | ANK: 512 dots<br>Other: 576 dots   |
| Characters in a Line    | Font A | ANK: 30 characters<br>Other: 34 characters  | ANK: 42 characters<br>Other: 48 characters                               |
|                         | Font B | ANK: 40 characters<br>Other: 52 characters  | ANK: 56 characters<br>Other: 72 characters                               |
| Character Size          | Font A | 12 dots x 24 dots (W x H)   |  |
|                         | Font B | 9 dots x 17 dots (W x H)  |  |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |  |
|                         | Font B | At the 15th dot from the top of the character   |  |
| Default Line Feed Space |        | 30 dots   |  |
| Color Specification     |        | First color   |  |
| Page Mode Default Area  |        | ANK: 360 dots x 1662 dots (W x H)<br>Other: 416 dots x 1662 dots (W x H)  | ANK: 512 dots x 1662 dots (W x H)<br>Other: 576 dots x 1662 dots (W x H) |
| Page Mode Maximum Area  |        | ANK: 360 dots x 1662 dots (W x H)<br>Other: 416 dots x 1662 dots (W x H)  | ANK: 512 dots x 1662 dots (W x H)<br>Other: 576 dots x 1662 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   |  |
| Paper Cut               |        | Cut, Feed cut   |  |
| Drawer Kick-Out         |        | Supported   |  |
| Buzzer                  |        | Supported   |  |
| Battery                 |        | Not supported   |  |

## TM-T81II

|                         |        | 80 mm   |
|-------------------------|--------|---|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |
| Language                |        | Simplified Chinese model  |
| Print Width             |        | 576 dots  |
| Characters in a Line    | Font A | ANK: 48 characters  |
|                         | Font B | ANK: 64 characters  |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |
| Character Baseline      | Font A | At the 21st 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   |

## TM-T82

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Language                |        | <ul style="list-style-type: none"> <li>• Simplified Chinese model</li> <li>• South Asian model</li> </ul>   |                              |
| Print Width             |        | 420 dots  | 576 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters  | ANK: 48 characters           |
|                         | Font B | ANK: 46 characters  | ANK: 64 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 420 dots x 831 dots (W x H)   | 576 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 420 dots x 1662 dots (W x H)  | 576 dots x 1662 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, Composite Symbology  |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Optional  |                              |
| Battery                 |        | Not supported   |                              |

## TM-T82II

|                         |        | 58 mm  | 80 mm                        |
|-------------------------|--------|--|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)  |                              |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• South Asian model</li> </ul>                            |                              |
| Print Width             |        | 420 dots   | 576 dots                     |
| Characters in a Line    | Font A | ANK: 35 characters   | ANK: 48 characters           |
|                         | Font B | ANK: 46 characters   | ANK: 64 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)   |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)  |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 420 dots x 831 dots (W x H)  | 576 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 420 dots x 1662 dots (W x H)   | 576 dots x 1662 dots (W x H) |
| Barcode                 |        | UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39,<br>ITF, CODABAR, CODE93, CODE128, GS1-128,<br>GS1 DataBar Omnidirectional, GS1 DataBar Truncated,<br>GS1 DataBar Limited, GS1 DataBar Expanded |                              |
| Two-Dimensional Code    |        | PDF417, QR Code, MaxiCode, GS1 DataBar Stacked,<br>GS1 DataBar Stacked Omnidirectional,<br>GS1 DataBar Expanded Stacked, Composite Symbolology   |                              |
| Paper Cut               |        | Cut, Feed cut  |                              |
| Drawer Kick-Out         |        | Supported  |                              |
| Buzzer                  |        | Optional   |                              |
| Battery                 |        | Not supported  |                              |

## TM-T88V

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 180 dpi x 180 dpi (W x H)   |                              |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• Korean model</li> <li>• South Asian model</li> </ul> |                              |
| Print Width             |        | 360 dots  | 512 dots                     |
| Characters in a Line    | Font A | ANK: 30 characters  | ANK: 42 characters           |
|                         | Font B | ANK: 40 characters  | ANK: 56 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 360 dots x 831 dots (W x H)   | 512 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 360 dots x 1662 dots (W x H)  | 512 dots x 1662 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 (Composite Symbology not supported)   |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Optional  |                              |
| Battery                 |        | Not supported   |                              |

## TM-T88V iOS *Bluetooth* model

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 180 dpi x 180 dpi (W x H)   |                              |
| Language                |        | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• Korean model</li> <li>• South Asian model</li> </ul> |                              |
| Print Width             |        | 360 dots  | 512 dots                     |
| Characters in a Line    | Font A | ANK: 30 characters  | ANK: 42 characters           |
|                         | Font B | ANK: 40 characters  | ANK: 56 characters           |
| Character Size          | Font A | ANK: 12 dots x 24 dots (W x H)  |                              |
|                         | Font B | ANK: 9 dots x 17 dots (W x H)   |                              |
| Character Baseline      | Font A | At the 21st 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  |        | 360 dots x 831 dots (W x H)   | 512 dots x 831 dots (W x H)  |
| Page Mode Maximum Area  |        | 360 dots x 1662 dots (W x H)  | 512 dots x 1662 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 (Composite Symbology not supported)   |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Optional  |                              |
| Battery                 |        | Not supported   |                              |



**TM-T90II**

|                         |        | 58 mm   | 80 mm                        |
|-------------------------|--------|---|------------------------------|
| Resolution              |        | 203 dpi x 203 dpi (W x H)   |                              |
| Language                |        | Japanese model  |                              |
| Print Width             |        | 420 dots  | 576 dots                     |
| Characters in a Line    | Font A | 35 characters   | 48 characters                |
|                         | Font B | 42 characters   | 57 characters                |
|                         | Font C | 52 characters   | 72 characters                |
| Character Size          | Font A | 12 dots x 24 dots (W x H)   |                              |
|                         | Font B | 10 dots x 24 dots (W x H)   |                              |
|                         | Font C | 8 dots x 16 dots (W x H)  |                              |
| Character Baseline      | Font A | At the 21st dot from the top of the character   |                              |
|                         | Font B | At the 21st dot from the top of the character   |                              |
|                         | Font C | At the 15th dot from the top of the character   |                              |
| Default Line Feed Space |        | 30 dots   |                              |
| Color Specification     |        | First color   |                              |
| Page Mode Default Area  |        | 420 dots x 1662 dots (W x H)  | 576 dots x 1662 dots (W x H) |
| Page Mode Maximum Area  |        | 420 dots x 1662 dots (W x H)  | 576 dots x 1662 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 Omni-directional, GS1 DataBar Expanded Stacked  |                              |
| Paper Cut               |        | Cut, Feed cut   |                              |
| Drawer Kick-Out         |        | Supported   |                              |
| Buzzer                  |        | Supported   |                              |
| Battery                 |        | Not supported   |                              |

## TM-U220

|                         |                | 76 mm   | 69.5 mm            | 57.5 mm               |
|-------------------------|----------------|---|--------------------|-----------------------|
| Resolution              | Single-density | 80 dpi x 72 dpi (W x H)   |                    |                       |
|                         | Double-density | 160 dpi x 72 dpi (W x H)  |                    |                       |
| Language                |                | <ul style="list-style-type: none"> <li>• ANK model</li> <li>• Japanese model</li> <li>• Simplified Chinese model</li> <li>• Traditional Chinese model</li> <li>• Korean model</li> <li>• Thai model</li> <li>• South Asian model</li> </ul> |                    |                       |
| Print Width             | Single-density | 200 dots  | 180 dots           | 150 dots              |
|                         | Double-density | 400 or 385* half dots   | 360 half dots      | 300 or 297* half dots |
| Characters in a Line    | Font A         | ANK: 33 characters  | ANK: 30 characters | ANK: 25 characters    |
|                         | Font B         | ANK: 40 characters  | ANK: 36 characters | ANK: 30 characters    |
| Character Size          | Font A         | ANK: 9 dots x 9 dots (W x H)  |                    |                       |
|                         | Font B         | ANK: 7 dots x 9 dots (W x H)  |                    |                       |
| Character Baseline      | Font A         | -   |                    |                       |
|                         | Font B         | -   |                    |                       |
| Default Line Feed Space |                | 12 dots   |                    |                       |
| Color Specification     |                | First color   |                    |                       |
| Page Mode Default Area  |                | -   |                    |                       |
| Page Mode Maximum Area  |                | -   |                    |                       |
| Barcode                 |                | Not supported   |                    |                       |
| Two-Dimensional Code    |                | Not supported   |                    |                       |
| Paper Cut               |                | Cut, No cut   |                    |                       |
| Drawer Kick-Out         |                | Supported   |                    |                       |
| Buzzer                  |                | Not supported   |                    |                       |
| Battery                 |                | Not supported   |                    |                       |

\*: DipSW2-1 = ON

## TM-U330

|                         |                      | 76 mm                            | 69.5 mm            | 57.5 mm            |
|-------------------------|----------------------|----------------------------------|--------------------|--------------------|
| Resolution              | Single-density       | 80 dpi x 72 dpi (W x H)          |                    |                    |
|                         | Double-density       | 160 dpi x 72 dpi (W x H)         |                    |                    |
| Language                |                      | Simplified Chinese model         |                    |                    |
| Print Width             | 120 dpi base         | 300 dots                         | 270 dots           | 225 dots           |
|                         | 240 dpi base         | 600 dots                         | 540 dots           | 450 dots           |
|                         | 180 dpi base         | 450 dots                         | 405 dots           | 337 dots           |
| Characters in a Line    | Font A               | ANK: 42 characters               | ANK: 38 characters | ANK: 32 characters |
|                         | Font B               | ANK: 33 characters               | ANK: 30 characters | ANK: 25 characters |
|                         | Chinese (180/90 dpi) | ANK: 16 characters               | ANK: 15 characters | ANK: 12 characters |
|                         | Chinese (80 dpi)     | ANK: 22 characters               | ANK: 20 characters | ANK: 16 characters |
| Character Size          | Font A               | ANK: 9 dots x 24 dots (W x H)    |                    |                    |
|                         | Font B               | ANK: 7 dots x 24 dots (W x H)    |                    |                    |
|                         | Chinese              | Kanji: 24 dots x 24 dots (W x H) |                    |                    |
| Character Baseline      | Font A               | -                                |                    |                    |
|                         | Font B               | -                                |                    |                    |
|                         | Chinese              | -                                |                    |                    |
| Default Line Feed Space |                      | 12 dots                          |                    |                    |
| Color Specification     |                      | First color                      |                    |                    |
| Page Mode Default Area  |                      | -                                |                    |                    |
| Page Mode Maximum Area  |                      | -                                |                    |                    |
| Barcode                 |                      | Not supported                    |                    |                    |
| Two-Dimensional Code    |                      | Not supported                    |                    |                    |
| Paper Cut               |                      | Cut, No cut                      |                    |                    |
| Drawer Kick-Out         |                      | Supported                        |                    |                    |
| Buzzer                  |                      | Not supported                    |                    |                    |
| Battery                 |                      | Not supported                    |                    |                    |

## Cautions

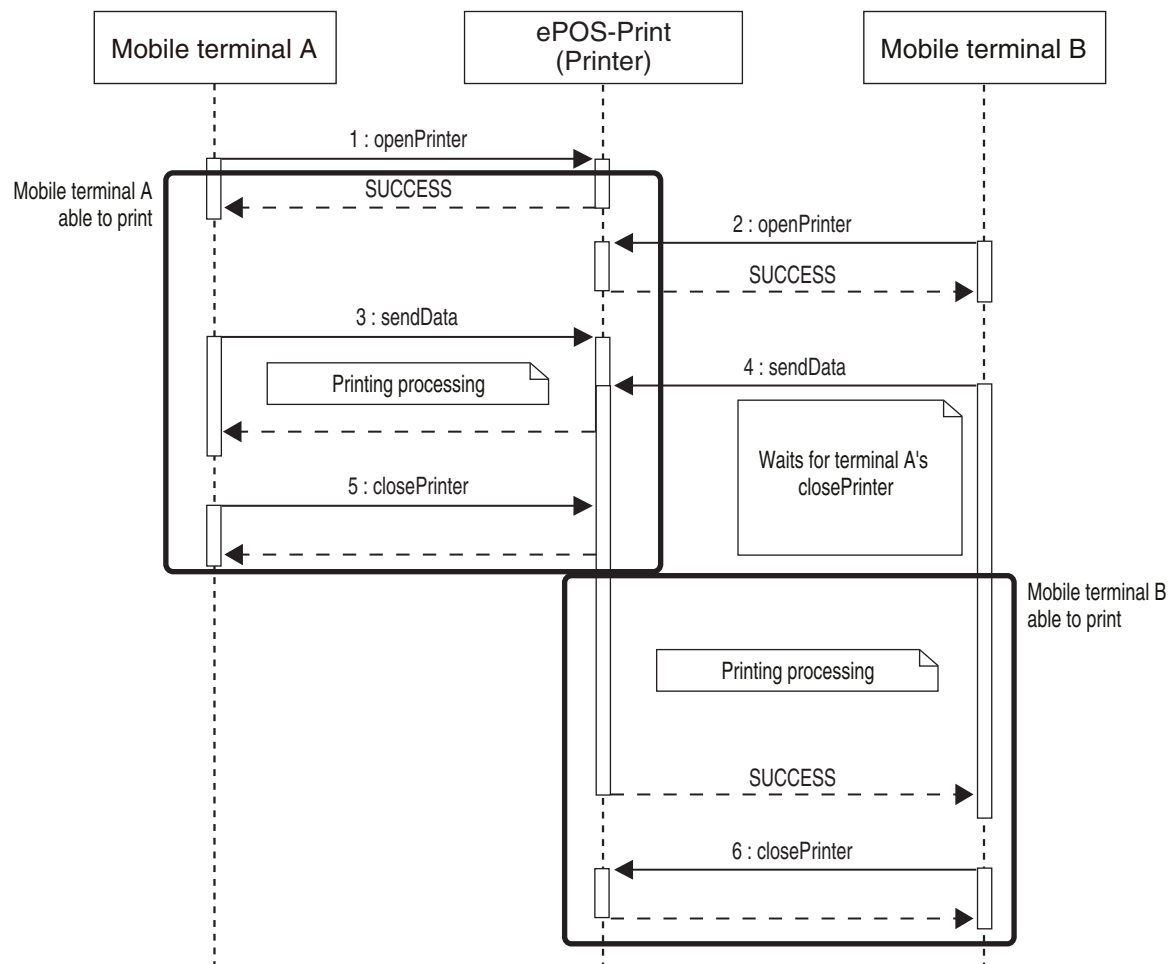
### If you Use the Printer from Multiple Mobile Terminals

If you use the printer from multiple mobile terminals, while you are using a particular terminal it will not be possible to print from the other ones. With Version 1.6.0 and later, if openPrinter processing has been initiated on one terminal when the printer is being used by another terminal, the openPrinter processing will wait for the other terminal's processing to end.

The chart below shows the flow of processing when a single printer is used from mobile terminal A and mobile terminal B.

#### Version 1.5.0 and earlier

With Version 1.5.0 and earlier, mobile terminal B will wait for mobile terminal A's closePrinter processing to end before executing sendData processing.



**Version 1.6.0 and later**

With Version 1.6.0 and later, mobile terminal B will wait for mobile terminal A's closePrinter processing to end before executing openPrinter processing.

