MP-A40 Series
SDK for Android™
Application Programmer's Guide

U00136204708

Seiko Instruments Inc.

U00136204700	August 2016
U00136204701	November 2016
U00136204702	January 2018
U00136204703	February 2019
U00136204704	October 2019
U00136204705	May 2022
U00136204706	July 2024
U00136204707	December 2024
U00136204708	June 2026

Copyright © 2016-2026 by Seiko Instruments Inc. All rights reserved.

Microsoft®, Windows®, and Microsoft Store are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.
Android™ is a trademark of Google LLC.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.
Oracle and Java® are registered trademarks of Oracle Corporation and/or its affiliates.
The company names and product names appearing on our manual are trademarks or registered trademarks of their respective owners.

Seiko Instruments Inc. (hereinafter referred to as "SII") has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

SII is a trademark of Seiko Instruments Inc.

Introduction

This document describes the MP-A40 Series SDK for Android (hereinafter referred to as "the SDK") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Printers

This section lists the printers supported by the SDK.

Printer	Communication Interface
MP-A40 (Bluetooth Model)	Bluetooth
	USB
MP-A40 (Wireless LAN Model)	Wireless LAN
	USB

Table of Contents

Chapter 1 Product Overview 1-1

1.1 Function 1-1
1.2 Configuration 1-1
1.2.1 SII Print Class Library 1-2
1.2.2 Sample Program 1-2

Chapter 2 Product Specification 2-1

2.1 Product Specification 2-1
2.1.1 Operating Environment 2-1
2.1.2 Operating Conditions 2-2
2.1.3 Precaution 2-2

Chapter 3 How to Use Library 3-1

3.1 Developmental Environment for Android Application 3-1
3.2 Use Developed Android Application on Android Device 3-2
3.3 Provided Files 3-3
3.4 Build Library into Android Studio Projects 3-4
3.5 Precautions 3-7

Chapter 4 Function of Library 4-1

4.1 Overview of Library 4-1
4.2 Structure of Library 4-1
4.3 Receive Data Process and Limitations 4-1
4.4 API Reference 4-2
4.4.1 Interface 4-4
(1) CallbackFunctionListener 4-4
onStatusChanged Register process on printer status change 4-4
(2) PrinterListener 4-5
onDiscoveryFinished Finish event of printer search 4-5
4.4.2 Class 4-6
(1) PrinterManager 4-6
PrinterManager Constructor 4-8
open Start using printer 4-8
close Finish using printer 4-9
isOpened Retrieve printer using status 4-9
setWriteTimeout Set send timeout period 4-10
getWriteTimeout Retrieve send timeout period 4-10
setResponseTimeout Set receive timeout period 4-10
getResponseTimeout Retrieve receive timeout period 4-11
write Send Binary data 4-11

read Retrieve receive data 4-11
getReadSize Retrieve available receive data size 4-12
writeAndWaitResponse Send and receive binary data 4-12
reset Reset printer 4-13
getStatus Retrieve printer status 4-13
startCallbackFunction Start callback of printer status change 4-14
stopCallbackFunction Finish callback of printer status change 4-14
registerStyleSheet Register style sheet to printer 4-14
deleteStyleSheet Delete style sheet in printer 4-15
registerLogo Register logo to printer 4-15
deleteLogo Delete logo in printer 4-16
getPrinterInformation
getPrinterInformationNumber
getPrinterInformationString Retrieve printer information 4-16
getVersion Retrieve SDK version 4-18
controlTransaction Start/finish print data batch sending 4-18
selectStandardMode Select standard mode 4-19
selectPageMode Start page mode 4-19
printPageModeData Print page mode data 4-20
setPageModeVerticalPosition Specify vertical absolute position in page mode 4-20
setStandardModeArea Specify print area in standard mode 4-21
setStandardModeAlignment Alignment 4-22
setHorizontalPosition Specify absolute position 4-22
setStandardModeBarcodeDirection
Select print direction for barcode or 2-dimensional barcode 4-23
setLineSpacing Specify line space amount 4-23
setCharacterRightSpace Specify character right space amount 4-23
selectCharacterSet Select character set 4-24
getCharacterSet Retrieve specified character set 4-24
selectInternationalCharacterSet Select international character set 4-24
getInternationalCharacter Retrieve specified international character set 4-25
setCharacterFormatting Format character 4-25
printText Send text data 4-26
printLogo Print logo registered in printer 4-26
sendDataFile Send file data 4-26
sendImage Send image data 4-28
printPDF Print PDF page 4-28
printBarcode Print barcode 4-29
print2DCode Print 2-dimensional barcode 4-32
printPageModeRectangle Draw rectangular 4-34
feedLine Feed paper by line 4-35
feedDotLine Feed paper by dot 4-35
feedCutPosition Feed paper to cut position 4-35
feedMarkPosition Marked paper form feed 4-36
setLog Specify log output 4-36
(2) PrinterStatus 4-37
getErrOffline Retrieve offline error status 4-37

getErrHardware Retrieve hardware error status 4-37
getErrVoltage Retrieve Vp Voltage error status 4-38
getErrHeadTemperature Retrieve head temperature error status 4-38
getErrOutOfPaper Retrieve out-of-paper error status 4-38
getErrMarkPaperJam Retrieve paper jam error status while detecting marked paper 4-38
getErrCoverOpen Retrieve paper cover open error status 4-38
getErrBattery Retrieve battery error status 4-38
getStateFeedSwitch Retrieve feed switch status 4-39
getStatePaperFeed Retrieve paper feed status 4-39
getStateReturnWaiting Retrieve return waiting status 4-39
getStateFlashMemoryRewriting Retrieve FLASH memory rewrite status 4-39
getStateBattery Retrieve battery voltage status 4-39
(3) PrinterDiscovery 4-40
PrinterDiscovery Constructert 4-40
startDiscoveryPrinter Start printer search 4-40
cancelDiscoveryPrinter Cancel printer search 4-41
getFoundPrinter Retrieve information list of searched printer 4-41
(4) DiscoveredPrinter 4-42
getPrinterModel Retrieve printer model name 4-42
getBluetoothAddress Retrieve Bluetooth address 4-42
getIpAddress Retrieve IP address 4-42
getMacAddress Retrieve MAC address 4-42
getSerialNumber Retrieve Serial Number 4-43
4.4.3 Enumerated Type (Enum) 4-44
(1) PrinterInterface 4-44
(2) PrinterModel 4-44
(3) Dithering 4-44
(4) PrinterInformation 4-44
(5) TransactionFunction 4-45
(6) Direction 4-46
(7) Alignment 4-46
(8) CharacterSet 4-47
(9) InternationalCharacterSet 4-48
(10) CharacterType 4-48
(11) CharacterScale 4-49
(12) Underline 4-49
(13) Bold 4-49
(14) Reverse 4-49
(15) Rotate 4-50
(16) TypeBarcode 4-51
(17) ModuleWidthBarcode 4-51
(18) HriPositionBarcode 4-51
(19) NwRatioBarcode 4-52
(20) Type2DCode 4-52
(21) Mode2Dcode 4-52
(22) ModuleSize2DCode 4-54
(23) ErrorCorrect2DCode 4-56

(24) LogFileSize....4-57
(25) ErrorCode....4-57
4.4.4 Exception....4-58
(1) PrinterException....4-58
getErrorCode Retrieve error codes....4-58

Chapter 5 Sample Program 5-1

5.1 Sample Program Overview....5-1
5.2 How to Use Sample Programs....5-1
5.3 Sample Program Function....5-4
5.4 Precaution....5-5

Chapter 6 Disclaimer 6-1

Appendix A Character Sets (Character Code Table) A-1

A.1 Character Code Table....A-1
A.2 International Character Set....A-11

Chapter 1
Product Overview

This chapter describes the product overview of the SDK.

1.1 Function

The SII print class library included in the SDK, provides Android applications with the functions to use SII printer MP-A40 series (hereinafter referred to as "printer").
Moreover, the SDK provides Android Studio projects as a sample program for SII print class library.

1.2 Configuration

The SII print class library and the sample programs included in the SDK are located in the section surrounded by dashed lines in the Android OS configuration diagram (Figure 1-1).

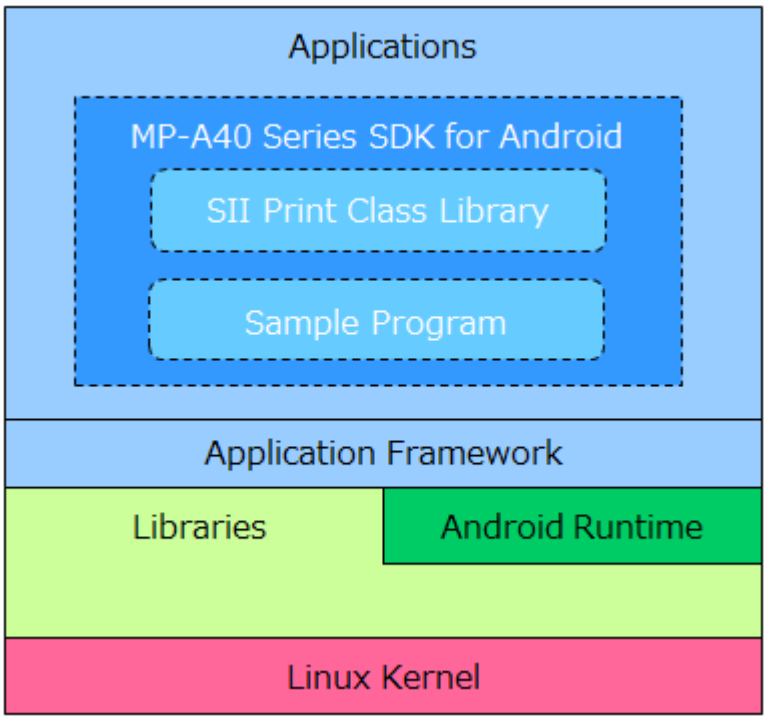


Figure 1-1

1.2.1 SII Print Class Library

By using SII Print Class Library (hereinafter referred to as "the library"), Android applications can easily send print data and printer commands to printer through communication port (Bluetooth, USB, or TCP/IP) on an Android device. Also, the applications can retrieve printer status.

The library provides the following functions.

- Connection/disconnection to/from a printer
- Sending data to a printer (print data and/or printer commands)
- Barcode print and 2-dimensional barcode print
- Sending a data file to a printer (print data and/or printer commands)
- Retrieving printer status
- Retrieving various responses from a printer
- Printer search by Bluetooth or TCP/IP

1.2.2 Sample Program

SII provides this program as an Android application sample using the library.

Chapter 2
Product Specification

This chapter describes the product specification of the library.

2.1 Product Specification

2.1.1 Operating Environment

Operating environment for the library is shown in the following table.

<table>
<tr><td rowspan="3">Printer</td><td colspan="2">Model</td><td colspan="3">MP-A40</td></tr>
<tr><td colspan="2">F/W Version</td><td colspan="3">1.00 or later</td></tr>
<tr><td colspan="2">Communication Interface</td><td>Bluetooth</td><td>USB</td><td>Wireless LAN</td></tr>
<tr><td rowspan="14">Android device</td><td colspan="2">Communication Port</td><td>Bluetooth*1</td><td>USB*2</td><td>TCP/IP*3</td></tr>
<tr><td rowspan="13">OS</td><td>Android 7.0 (API 24)</td><td rowspan="13">Supported</td><td rowspan="13">Supported</td><td rowspan="13">Supported</td></tr>
<tr><td>Android 7.1 (API 25)</td></tr>
<tr><td>Android 8.0 (API 26)</td></tr>
<tr><td>Android 8.1 (API 27)</td></tr>
<tr><td>Android 9.0 (API 28)</td></tr>
<tr><td>Android 10.0 (API 29)</td></tr>
<tr><td>Android 11.0 (API 30)</td></tr>
<tr><td>Android 12.0 (API 31)</td></tr>
<tr><td>Android 12.1 (API 32)</td></tr>
<tr><td>Android 13.0 (API 33)</td></tr>
<tr><td>Android 14.0 (API 34)</td></tr>
<tr><td>Android 15.0 (API 35)</td></tr>
<tr><td>Android 16.0 (API 36)</td></tr>
<tr><td colspan="3">Supported Language</td><td colspan="3">Japanese
English</td></tr>
</table>

(NOTE) *1: Bluetooth connection needs to be established by SPP (Serial Port Profile).
*2: Android device needs to support USB host function.
*3: Android device and printer need to be connected to the same network.
When the printer is the client mode, connect through access point.
When the printer is the simple AP mode, connect directly.

2.1.2 Operating Conditions

This section describes the operating conditions for the library in the following table.
Set the Function Setting / function selection to the values shown in the following table before using the library.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCES" for details about Function Setting.

MS	Function	Value	Setting
1-1	Interface Selection (Interface)	0 / 1	0: USB 1: USB/Wireless*1
1-2	Mark Mode Selection (Mark Mode)	0 / 1	0: Enable*2 1: Disable
1-3	Command System Selection (Command System)	0	ESC/POS
1-4		0	
1-5		0	
2-2	Realtime Command Selection (Realtime Command)	0	Enable
9-1	Automatic Status Response Selection <ESC/POS> (Auto Status Back)	0	Enable
9-2	Initialized Response Selection <ESC/POS> (Init. Response)	0	Enable

(NOTE) *1: Select "USB / Wireless" when using Bluetooth or Wireless LAN interface.
*2: Select "Enable" when feeding to home position of the marked thermal paper.

2.1.3 Precaution

When using TCP/IP connection in this library, the printer port is occupied. Therefore, the communication port cannot be shared with printer drivers or other libraries.

Chapter 3
How to Use Library

This chapter describes development environment for Android application and how to use the library.

3.1 Developmental Environment for Android Application

In order to develop Android applications, following tools are required. See each of the following URLs for more details.

- Android Studio

  http://developer.android.com/sdk/index.html

- USB driver for Windows (When develop on Windows environment)

  http://developer.android.com/tools/extras/oem-usb.html

This chapter and after in this document, it is assumed that there is an environment where you can use each tool.

3.2 Use Developed Android Application on Android Device

In order to use developed Android applications on the Android device (actual device), configure the following settings on the Android device.

(NOTE)	**This procedure is based on the menus of Android 6.0. Menu contents may vary depending on the using Android device.**

(a) Select [Settings], [Security], and [Unknown sources].(Figure 3-1)

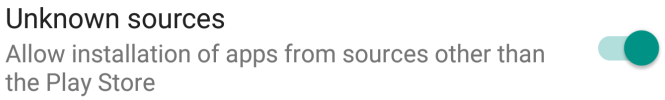


Figure 3-1

(b) Select [Settings], [Developer options], and [USB debugging].(Figure 3-2)

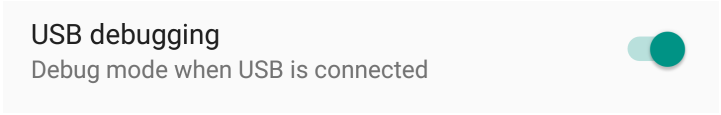


Figure 3-2

3.3 Provided Files

The file configuration of the SII print class library is as follows.

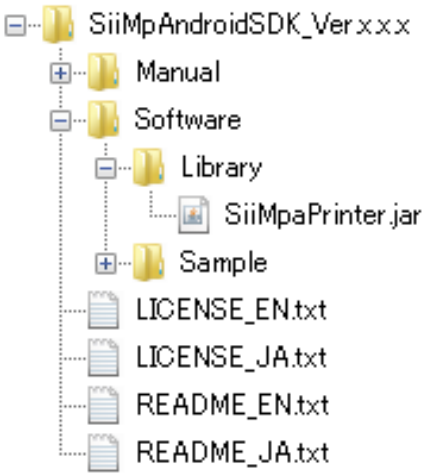


Figure 3-3

The file format of the library is jar. The file name of the library is **SiiMpaPrinter.jar**.

3.4 Build Library into Android Studio Projects

Using the project of the sample program included in the SDK as an example, this section describes how to build the library into Android studio projects.

See "Chapter 5 Sample Program" for sample programs included in the SDK.

(a) Select and right click the module (app) displayed in the Android Project view of Android Studio, select [New] and [Directory] (Figure 3-4), and enter "libs" in the folder name. (Figure 3-5)

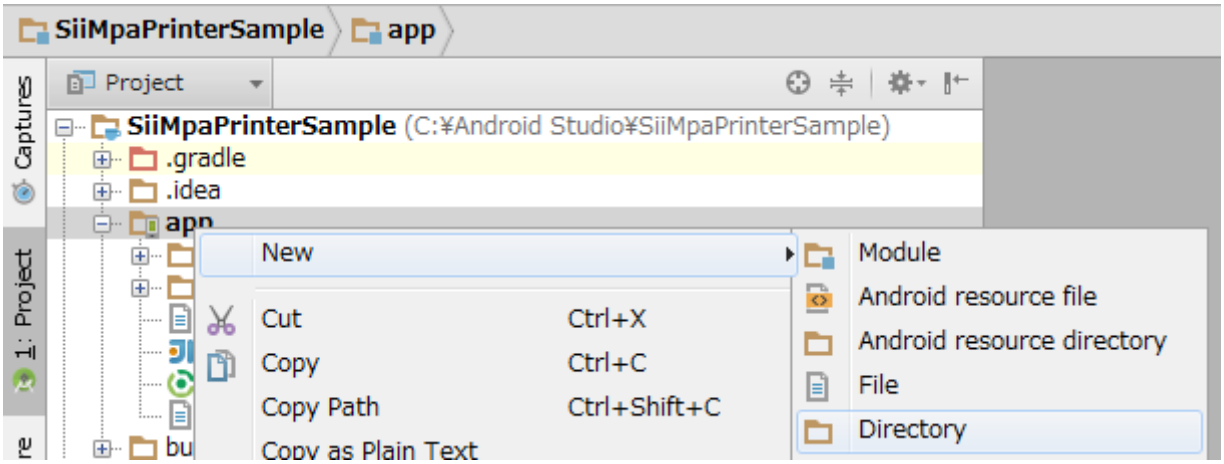


Figure 3-4

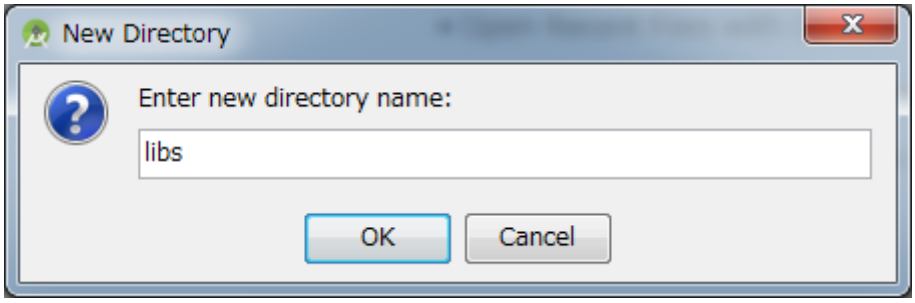


Figure 3-5

(b) Copy the library file (SiiMpaPrinter.jar) into the folder (\SiiMpaPrinterSample\app\libs) created in step (a).

(c) Right click SiiMpaPrinter.jar, select [Add as library], and select the module (app). (Figure 3-6)

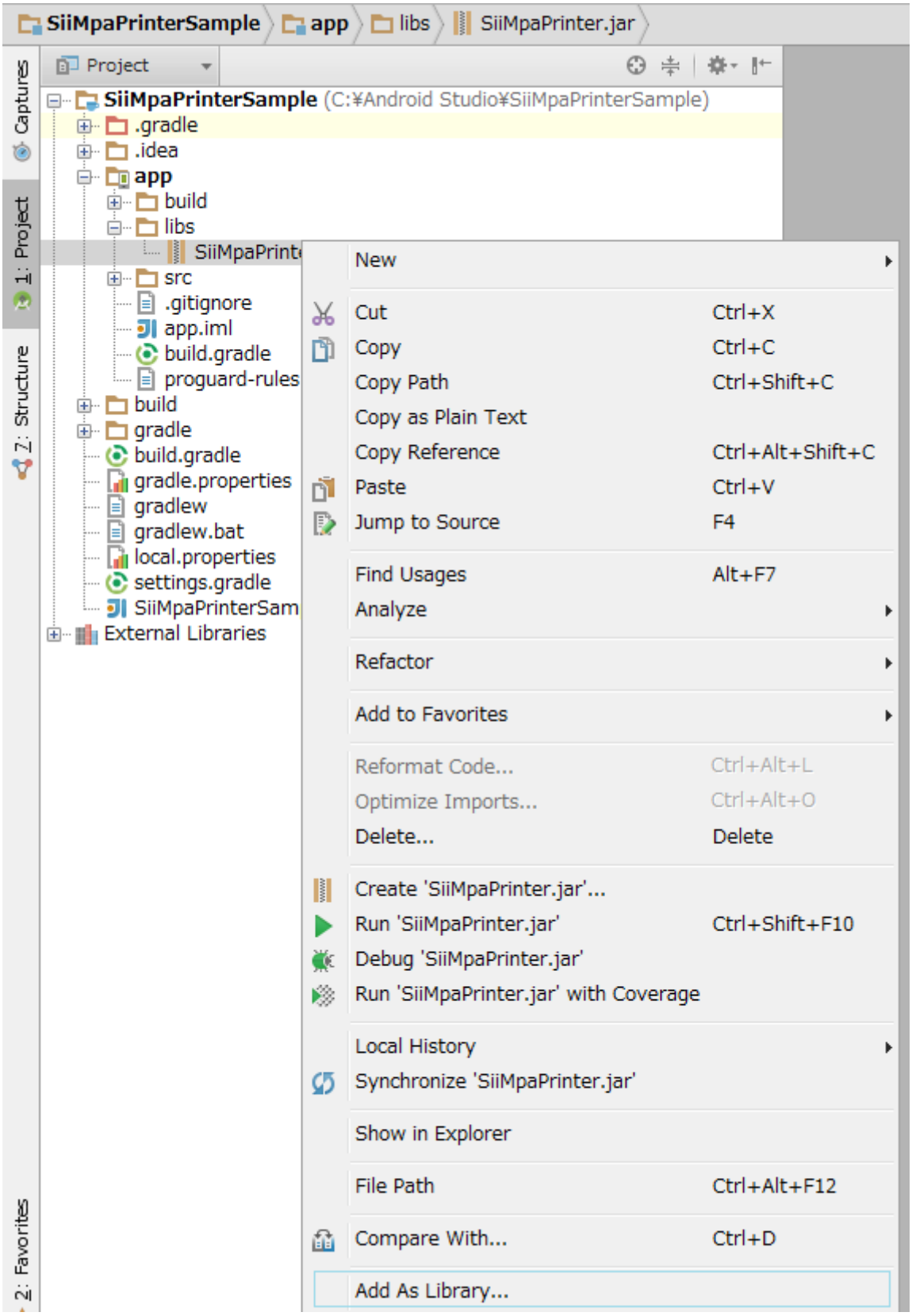


Figure 3-6

(d) After adding the library, the state becomes as shown in Figure 3-7.

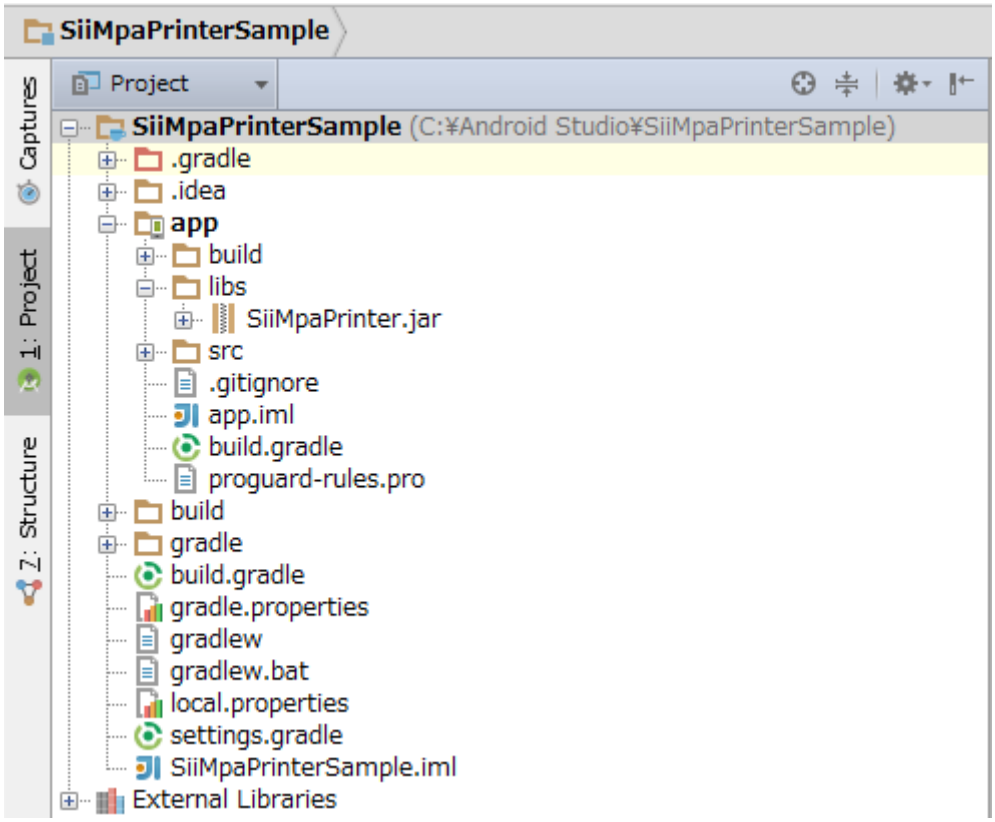


Figure 3-7

(e) Add the following codes at the beginning of the main source file. (In the SiiMpaPrinterSample, MainActivity.java is used.)

```
import com.seikoinstruments.sdk.mobileprinter.Alignment;
import com.seikoinstruments.sdk.mobileprinter.Bold;
import com.seikoinstruments.sdk.mobileprinter.CharacterScale;
import com.seikoinstruments.sdk.mobileprinter.CharacterSet;
import com.seikoinstruments.sdk.mobileprinter.CharacterType;
import com.seikoinstruments.sdk.mobileprinter.Direction;
import com.seikoinstruments.sdk.mobileprinter.ErrorCorrect2Dcode;
import com.seikoinstruments.sdk.mobileprinter.ErrorCode;
import com.seikoinstruments.sdk.mobileprinter.LogFileSize;
import com.seikoinstruments.sdk.mobileprinter.HriPositionBarcode;
import com.seikoinstruments.sdk.mobileprinter.InternationalCharacterSet;
import com.seikoinstruments.sdk.mobileprinter.Mode2Dcode;
import com.seikoinstruments.sdk.mobileprinter.ModuleSize2Dcode;
import com.seikoinstruments.sdk.mobileprinter.ModuleWidthBarcode;
import com.seikoinstruments.sdk.mobileprinter.NwRatioBarcode;
import com.seikoinstruments.sdk.mobileprinter.PrinterException;
import com.seikoinstruments.sdk.mobileprinter.PrinterInterface;
import com.seikoinstruments.sdk.mobileprinter.PrinterInformation;
import com.seikoinstruments.sdk.mobileprinter.PrinterManager;
import com.seikoinstruments.sdk.mobileprinter.PrinterModel;
import com.seikoinstruments.sdk.mobileprinter.PrinterStatus;
import com.seikoinstruments.sdk.mobileprinter.Reverse;
import com.seikoinstruments.sdk.mobileprinter.Rotate;
import com.seikoinstruments.sdk.mobileprinter.TransactionFunc;
import com.seikoinstruments.sdk.mobileprinter.Type2DCode;
import com.seikoinstruments.sdk.mobileprinter.TypeBarcode;
import com.seikoinstruments.sdk.mobileprinter.Underline;
```

By completing these procedures, functions of the library become available.

3.5 Precautions

- **About Scoped Storage**

  "Scoped Storage" that is introduced in Android 10 distinguishes between app-specific storage and external storage.

  When targeting Android 10 (API 29) or later, files that do not correspond to media files in the external storage cannot be handled directly. Files that do not correspond to media files can be handled by using the "Storage Access Framework".

  See below for details of Scoped Storage.

  - Data and file storage overview
    https://developer.android.com/training/data-storage

Chapter 4
Function of Library

This chapter describes the APIs for each class implemented in the library.

4.1 Overview of Library

The library provides the functions to use printers for Android-enabled applications.

4.2 Structure of Library

The file format of the library is jar. The file name of the library is SiiMpaPrinter.jar.

In order to use the library in Android applications, build the library into Android studio projects.

See "Chapter 3 How to Use the Library" for details about how to build the library into Android studio projects.

4.3 Receive Data Process and Limitations

This library buffers the receive data from the printer into the receive buffer in the SDK (hereinafter referred to as "the buffer") after starting to use the printer by **open**. Since the printer replies the auto status response every time the status changes, the receive data is buffered into the buffer sequentially. The maximum receive data to be buffered is 4096 bytes. When the printer is reconnected, the receive buffer in the SDK may collectively receive the data that was buffered in the printer before reconnecting.

The receive data buffered in the buffer can be retrieved by **read**. The receive data retrieved by **read** is deleted from the buffer.

When the buffered receive data exceeds 4096 bytes, the old data in buffer is discarded sequentially; therefore, execute **read** so that the receive data does not exceed 4096 bytes. The receive data size can be retrieved by **getReadSize**.

For the details of the each method, see "4.4 API Reference".

For the details of the each response, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

4.4 API Reference

The package of this library is com.seikoinstruments.sdk.mobileprinter.
com.seikoinstruments.sdk.mobileprinter provides the following functions.

- **Interface**

Name	Description
CallbackFunctionListener	Interface that implements the printer status change.
PrinterListener	Interface that retrieves the complete event of printer searching.

- **Class**

Name	Description
PrinterManager	Class that provides the API used for communication with the printer and for printing.
PrinterStatus	Class that stores the printer status.
PrinterDiscovery	Class that searches the printer.
DiscoveredPrinter	Class that stores the printer information searched by printer searching.

- **Enumerated type (enum)**

Name	Description
PrinterInterface	Enumerator used for specifying the communication interface of **open**.
PrinterModel	Enumerator used for specifying the printer model of **open**.
Dithering	Enumerator used for dithering of **resisterLogo**, **sendDataFile**, and **printPDF**.
PrinterInformation	Enumerator used for specifying the printer information of **getPrinterInformation**, **getPrinterInformationNumber** and **getPrinterInformationString**.
TransactionFunction	Enumerator used for specifying the batch process control method of **controlTransaction**.
Direction	Enumerator used for specifying the print direction of **selectPageMode**.
Alignment	Enumerator used for specifying the print position of **setStandardModeAlignment**.
CharacterSet	Enumerator used for specifying the character set of **selectCharacterSet** and **getCharacterSet**.
InternationalCharacterSet	Enumerator used for specifying the international character set of **selectInternationalCharacterSet** and **getInternationalCharacter**.
CharacterType	Enumerator used for specifying the character font of **setCharacterFormatting**.
CharacterScale	Enumerator used for specifying the character scale of **setCharacterFormatting**.
Underline	Enumerator used for specifying the underline of **setCharacterFormatting**.

Name	Description
Bold	Enumerator used for specifying the bold print of **setCharacterFormatting**.
Reverse	Enumerator used for specifying the reverse print of **setCharacterFormatting**.
Rotate	Enumerator used for specifying the rotation print of **setCharacterFormatting**, for specifying the print direction of **setStandardModeBarcodeDirection**, and for specifying the image rotation direction of **printPDF**.
TypeBarcode	Enumerator used for specifying the barcode type of **printBarcode**.
ModuleWidthBarcode	Enumerator used for specifying the barcode module width or narrow element of **printBarcode**.
HriPositionBarcode	Enumerator used for specifying the barcode HRI characters of **printBarcode**.
NwRatioBarcode	Enumerator used for specifying the barcode N:W ratio of **printBarcode**.
Type2DCode	Enumerator used for specifying the 2-dimensional barcode type of **print2DCode**.
Mode2Dcode	Enumerator used for specifying the 2-dimensional barcode mode of **print2DCode**.
ModuleSize2Dcode	Enumerator used for specifying the 2-dimensional barcode module size of **print2DCode**.
ErrorCorrect2Dcode	Enumerator used for specifying the 2-dimensional barcode error correction level of **print2DCode**.
LogFileSize	Enumerator used for specifying the maximum size of the log file of **setLog.**
ErrorCode	Enumerator used for retrieving the error code by **PrinterException**.

- **Exception**

Name	Description
PrinterException	Exception class that is thrown at API call.

4.4.1 Interface

(1) **CallbackFunctionListener**

- **Summary**

This interface is for retrieving the event of printer status change.
This interface provides the following function.

Public Methods

Name	Description
onStatusChanged	Register process on printer status change.

- **Public Methods**

onStatusChanged — Register process on printer status change

Syntax: public void **onStatusChanged** (PrinterStatus *status*)

Parameter: *status* — Instance of **PrinterStatus**

Description: This method is called when a callback of the printer status change is started by **startCallbackFunction** and the printer status is changed.

This is a method of interface so it is not implemented. Implement the optional process in the class that receives a callback of the printer status change.

In *status*, assign the instance of **PrinterStatus** class including the printer status at the changed point. Printer status can be retrieved from *status* by **PrinterStatus** class method. For the **PrinterStatus** method, see "**4.4.2**(2) **PrinterStatus**".

(2) **PrinterListener**

- **Summary**

This interface retrieves the events.
This interface provides the following function.

Public Methods

Name	Description
onDiscoveryFinished	Finish event of printer search

Constants

Name	Description
EVENT_FINISHED_DISCOVERY	Finish printer search Value = 0
EVENT_CANCELED_DISCOVERY	Cancel printer search Value = 1

- **Public Methods**

onDiscoveryFinished — Finish event of printer search

Syntax public void **onDiscoveryFinished** (int *event*)

Parameter *event* Printer event

Description This method is called when the search is finished by **startDiscoveryPrinter** or the search is canceled by **cancelDiscoveryPrinter**.
This is a method of interface so it is not implemented. Implement the optional process in the class that receives event notice of finish or cancel of printer search.

When this method is called by search finish, **EVENT_FINISHED_DISCOVERY** is assigned to *event*.
When this method is called by search cancel, **EVENT_CANCELED_DISCOVERY** is assigned to *event*.

4.4.2 Class

(1) **PrinterManager**

- **Summary**

  This class provides the API used for communication with the printer and for printing.
  This class provides the following functions.

  Public Methods

Name	Description	Standard Mode[*1]	Page Mode[*1]
PrinterManager	Constructor	✓	✓
open	Start using printer	✓	✓
close	Finish using printer	✓	✓
isOpened	Retrieve printer using status	✓	✓
setWriteTimeout	Set send timeout period	✓	✓
getWriteTimeout	Retrieve send timeout period	✓	✓
setResponseTimeout	Set receive timeout period	✓	✓
getResponseTimeout	Retrieve receive timeout period	✓	✓
write	Send binary data	✓	✓
read	Retrieve receive data	✓	✓
getReadSize	Retrieve available receive data size	✓	✓
writeAndWaitResponse	Send and receive binary data	✓	✓
reset	Reset printer	✓	✓
getStatus	Retrieve printer status	✓	✓
startCallbackFunction	Start callback of printer status change	✓	✓
stopCallbackFunction	Finish callback of printer status change	✓	✓
registerStyleSheet	Register style sheet to printer	✓	✓
deleteStyleSheet	Delete style sheet in printer	✓	✓
registerLogo	Register logo to printer	✓	✓
deleteLogo	Delete logo in printer	✓	✓
getPrinterInformation	Retrieve printer information	✓	✓
getPrinterInformationNumber		✓	✓
getPrinterInformationString		✓	✓
getVersion	Retrieve SDK version	✓	✓
controlTransaction	Start/finish print data batch sending	✓	✓
selectStandardMode	Start standard mode	-	✓
selectPageMode	Start page mode	✓	-
printPageModeData	Print page mode data	-	✓[*2]
setPageModeVerticalPosition	Specify vertical absolute position in page mode	-	✓[*2]
setStandardModeArea	Specify print area in standard mode	✓[*3]	-
setStandardModeAlignment	Alignment	✓[*3]	-
setHorizontalPosition	Specify absolute position	✓	✓

Name	Description	Standard Mode*1	Page Mode*1
setStandardModeBarcodeDirection	Select print direction for barcode or 2-dimensional barcode	✓*3	-
setLineSpacing	Specify line space amount	✓*4	✓*4
setCharacterRightSpace	Specify character right space amount	✓*4	✓*4
selectCharacterSet	Select character set	✓	✓
getCharacterSet	Retrieve specified character set	✓	✓
setInternationalCharacter	Select international character set	✓	✓
getInternationalCharacter	Retrieve specified international character set	✓	✓
setCharacterFormatting	Format character	✓	✓
printText	Send text data	✓	✓
printLogo	Print logo registered in printer	✓	✓
sendDataFile	Send file data	✓	✓
sendImage	Send image data	✓	✓
printPDF	Print PDF page	✓*3	-
printBarcode	Print barcode	✓	✓
print2DCode	Print 2-dimensional barcode	✓	✓
printPageModeRectangle	Draw rectangular	-	✓*2
feedLine	Feed paper by line	✓	✓
feedDotLine	Feed paper by dot	✓	✓
feedCutPosition	Feed paper to cut position	✓	✓
feedMarkPosition	Marked paper form feed	✓	✓
setLog	Specify log output	✓	✓

*1 ✓: Enable, -: Disable
*2 When use this method, execute **SelectPageMode** beforehand to start page mode.
*3 When use this method, execute **SelectStandardMode** beforehand to start standard mode.
*4 Independent settings are available for standard mode and page mode respectively.

Constants

Constant Name	Description
PAPER_WIDTH_80mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 80 mm Constant value = 576
PAPER_WIDTH_100mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 100 mm Constant value = 736
PAPER_WIDTH_105mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 105 mm Constant value = 776
PAPER_WIDTH_112mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 112 mm Constant value = 832

- **Public Methods**

PrinterManager — Constructor

Constructor for **com.seikoinstruments.sdk.mobileprinter.PrinterManager** class.

Syntax public **PrinterManager**(Context *context*)

Parameter

context	Android application context to call this method Specify the context that can be retrieved by **getApplicationContext**().

open — Start using printer

Starts using a printer.

Syntax public void **open**(PrinterInterface *prnIf*,
PrinterModel *prnModel*,
String *address*,
int *socketKeepingTime*,
boolean *secure*) throws **PrinterException**

Parameter

prnIf	Communication interface constant See "4.4.3(1) **PrinterInterface**" for available settings.
prnModel	Printer model constant See "4.4.3(2) **PrinterModel**" for available settings.
address	Varies depending on *prnIf* setting. ・When specify **PRN_IF_TCP**: In *address*, specify IP address of the printer to connect, or the host name. Example: "192.168.0.1", "host" ・When specify **PRN_IF_BT**: In *address*, specify Bluetooth address of the printer to connect. Example: "00:11:22:AA:BB:CC" ・When specify **PRN_IF_USB**: The specified value is ignored.
socketKeepingTime	Varies depending on *prnIf* setting. ・When specify **PRN_IF_TCP**: The range is 60 to 300. When the specified value is below 60, the value is set to 60, when the specified value exceeds 300, the value is set to 300. In *socketKeepingTime*, specify the same value as the timeout period of unconnected communication specified in printer wireless LAN setting. The default value of the timeout period of unconnected communication is 300 seconds. For the details, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE". ・When specify **PRN_IF_BT**: The specified value is ignored. ・When specify **PRN_IF_USB**: The specified value is ignored.

	Secure	Varies depending on *prnIf* setting. ・When specify **PRN_IF_TCP**: The specified value is ignored. ・When specify **PRN_IF_BT**: true : connect to a printer in secure mode false: connect to a printer in insecure mode Normally, specify true in *secure*. Since some Android devices fail to connect in secure mode, specify false to connect the printer with such an Android device. ・When specify **PRN_IF_USB**: The specified value is ignored.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) PrinterException"

Description This method connects to a printer specified by *prnModel* through a communication interface specified by *prnIf*.

Monitoring the printer status is started by this method. The latest printer status is retrieved by **getStatus**. The change of printer status can be notified as an event by **onStatusChanged**, **startCallbackFunction**, or **stopCallbackFunction.**

Note Do not disable the automatic status response by printer command "Automatic Status Back Enable/Disable" or function settings. In that case, the printer status cannot be monitored, and the related function cannot be operated. For the automatic status response and the function settings, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When data is sent to a printer through TCP/IP connection and the socket keeping time is passed, communication socket is discarded. After that, a communication socket is created again and connection starts. If the printer is requested to connect from the other host on the same network at the timing of the communication socket discarding, the printer establishes the communication with the other host and the reconnection may be failed.

close — Finish using printer

Finishes using the printer and monitoring the printer status.

Syntax public void **close**() throws **PrinterException**

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) **PrinterException**"

Description Print data stored by **controlTransaction** is discarded.

isOpened — Retrieve printer using status

Retrieves that the printer's open status by **open**.

Syntax public boolean **isOpened**()

Return value

true	Using printer is started
false	Using printer is not started.

setWriteTimeout — Set send timeout period

Specifies the send timeout period.

Syntax public void **setWriteTimeout**(int *writeTimeout*)

Parameter

writeTimeout	Send timeout period (millisecond) The range is 1000 to 90000. When the specified value is below 1000, the value is set to 1000. When the specified value exceeds 90000, the value is set to 90000. When the value is not set in this method, the send timeout period is 10000 seconds (default value).

Description The specified value can be retrieved by **getWriteTimeout**.

This method is enabled in the following methods.
- **write**
- **writeAndWaitResponse** (Transmission process part)
- **registerLogo**
- **registerStyleSheet**
- **controlTransaction** (when **TRANSACTION_PRINT** is selected in *control*)
- **printText**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printBarcode**
- **print2DCode**

getWriteTimeout — Retrieve send timeout period

Retrieves the send timeout period.

Syntax public int **getWriteTimeout**()

Return value Send timeout period (millisecond)

setResponseTimeout — Set receive timeout period

Specifies the receive timeout period.

Syntax public void **setResponseTimeout**(int *respTimeout*)

Parameter

respTimeout	Receive timeout period (millisecond) The range is 1000 to 90000. When the specified value is below 1000, the value is set to 1000. When the specified value exceeds 90000, the value is set to 90000. When the value is not set in this method, the send timeout period is 10000 seconds (default value).

Description The specified value is retrieved by **getResponseTimeout**.

This method is enabled in the following methods.
- **writeAndWaitResponse**
- **getPrinterInformation**
- **getPrinterInformationNumber**
- **getPrinterInformationString**

getResponseTimeout — Retrieve receive timeout period

Retrieves the receive timeout period.

Syntax public int **getResponseTimeout**()

Return value Receive timeout period (millisecond)

write — Send Binary data

Sends the binary data.

Syntax public void **write**(byte[] *binary*, int *offset*) throws **PrinterException**

Parameter

binary	Binary data to send to a printer The maximum data size that can be specified is 3 MB (3145728 bytes).
offset	Specification of the starting position of the data send

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) **PrinterException**".

Description When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout**.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send. When perform Hardware Reset, use **reset**.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method is aborted by **reset**.

read — Retrieve receive data

Retrieves the receive data buffered in the buffer.

Syntax public byte[] read(int *bufferSize*) throws **PrinterException**

Parameter

bufferSize	Receive data size (byte) The range is 1 to 4096. When the specified value exceeds 4096, the value is set to 4096. When the specified value is below 0, sends the error notice.

Return value Received data

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) **PrinterException**".

Description When entire receive data buffered in the buffer needs to be retrieved, specify the value in *bufferSize* retrieved by **getReadSize**.

When this method is reexecuted after retrieving receive data by this method, retrieving starts from the following receive data in the buffer.

For the procedure and the limitation, see "4.3 Receive Data Process and the Limitations".

getReadSize — Retrieve available receive data size

Retrieves the available receive data size.

Syntax public int **getReadSize**() throws **PrinterException**

Return value Available receive data size (byte)

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) **PrinterException**".

writeAndWaitResponse — Send and receive binary data

Sends the binary data and retrieves the receive data after this method is executed.

Syntax public byte[] **writeAndWaitResponse**(byte[] *sendBinary*,
int *bufferSize*,
boolean *respControl*,
boolean *incAsbData*) throws **PrinterException**

Parameter

sendBinary — Data to send
The maximum data size that can be specified is 3 MB (3145728 bytes).

bufferSize — Maximum size of the data to send
The range is 1 to 10485760.
When the specified value exceeds 10485760, the value is set to 10485760.
When the specified value is below 0, sends the error notice.

respControl — Operation selection for receive process
true : Receive some data or continue to receive data until timeout period is over
false : Receive the specified sized data in *bufferSize* or continue to receive data until timeout period is over

incAsbData — Include the automatic status response to the receive data or not
true : Include
false : Not include

Return value Receive data

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) **PrinterException**".

Description This method is appropriate to send printer command for responding value of character string amount or capacity, and to retrieve the respond.

Next method execution after this method execution is not processed until completion of this method receiving. For retrieving the execution response or progress response, use **read** but not this method. For the details of execution response or progress response, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

For sending, the timeout period specified in **setWriteTimeout** is valid.

For receiving, the timeout period specified in **setResponseTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send. When perform Hardware Reset, use **reset**.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method is aborted by **reset**.

reset — Reset printer

Resets the printer.

Syntax public void **reset**() throws **PrinterException**

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "4.4.4(1) **PrinterException**".

Description When this method is executed while the following method is being processed, the process is aborted. The unsent or unreceived data of following methods are discarded.
- **write**
- **writeAndWaitResponse**
- **controlTransaction** (When **TRANSACTION_PRINT** is selected in *control*)
- **printText**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **getPrinterInformation**
- **getPrinterInformationNumber**
- **getPrinterInformationString**

While this method is being executed, the printer status becomes offline.

After this method is executed, wait for a few seconds before data send method is executed. Data send during reset may cause data lost.

For Bluetooth connection, when this method is executed but the printer is in the condition of no data accepting, this method succeeds, but the printer reset is not executed until the printer is ready to print. And in the meantime, data transmission cannot be performed.

getStatus — Retrieve printer status

Retrieves the latest printer status.

Syntax public PrinterStatus **getStatus**() throws **PrinterException**

Return value Printer status is returned by **PrinterStatus** class.
According to the method in **PrinterStatus** class, the printer status can be retrieved.
For the method in **PrinterStatus** class, see "**4.4.2**(2) (2)**PrinterStatus**".

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

startCallbackFunction — Start callback of printer status change

Starts the callback according to the printer status change.

Syntax public void **startCallbackFunction**(CallbackFunctionListener *listener*) throws **PrinterException**

Parameter *listener* Instance of **CallbackFunctionListener**

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The execution procedure needs to be registered by **onStatusChanged**.

stopCallbackFunction — Finish callback of printer status change

Finishes the callback started by **startCallbackFunction**.

Syntax public void **stopCallbackFunction**() throws **PrinterException**

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

registerStyleSheet — Register style sheet to printer

Registers the style sheet to the printer.

Syntax public void **registerStyleSheet**(String *filePath*, int *regNum*) throws **PrinterException**

Parameter *filePath* File path for style sheet
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.

- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

regNum Style sheet number
The range is 1 to 4.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description A maximum of 4 sheets can be registered.

The file extension for supporting style sheet is .css.

The maximum file size that can be registered is 1MB (1048576 bytes).

The maximum number that can be registered in 1 style sheet is 64.

When specify a style sheet number which style sheet is already registered, the style sheet is overwritten.

For the available style sheet to register, see the printer command "Style Sheet Registration" in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

The timeout period specified in **setWriteTimeout** is valid.

deleteStyleSheet — Delete style sheet in printer

Deletes the style sheet registered to the printer.

Syntax public void **deleteStyleSheet**(int *regNum*) throws **PrinterException**

Parameter

regNum	Style sheet number The range is 1 to 4.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

registerLogo — Register logo to printer

Registers the image data to a printer as a logo.
In syntax (a), dithering is fixed to be enabled.
In syntax (b), dithering can be specified.

Syntax (a) public void **registerLogo**(String *filePath*, int *regNum*) throws **PrinterException**

(b) public void **registerLogo**(String *filePath*,
int *regNum*,
Dithering *dithering*) throws **PrinterException**

Parameter

filePath — File path of image data
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.
- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  ・file://
  ・content://
  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

regNum — Logo number
The range is 0 to 99.

dithering — Dithering
See "**4.4.3**(3) **Dithering**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The file extension for supporting image data is .bmp, .jpg, .jpeg, and .png.

The color image data is converted to monochrome image by binarization.

The maximum file size that can be registered is 1 MB (1048576 bytes).

The maximum image sizes that can be registered are 8192 dots in width and 2304 dots in height.

The registered situation for the logo registered by this method can be confirmed by executing **getPrinterInformation** or **getPrinterInformationString** with specifying **GET_NV_MEM_KEYCODE_LIST** in *prnInfo*.

When specify a logo number which logo is already registered, the logo is overwritten.

The timeout period specified in **setWriteTimeout** is valid.

deleteLogo — Delete logo in printer

Deletes the logo registered to the printer.

Syntax public void **deleteLogo**(int *regNum*) throws **PrinterException**

Parameter

regNum	Logo number The range is 0 to 99.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The logo deleted situation by this method can be confirmed by executing **getPrinterInformation** or **getPrinterInformationString** with specifying **GET_NV_MEM_KEYCODE_LIST** in *prnInfo*.

getPrinterInformation
getPrinterInformationNumber
getPrinterInformationString — Retrieve printer information

Retrieves the printer information.

Syntax
(a) public byte[] **getPrinterInformation**(PrinterInformation *prnInfo*) throws **PrinterException**
(b) public int **getPrinterInformationNumber**(PrinterInformation *prnInfo*) throws **PrinterException**
(c) public String **getPrinterInformationString**(PrinterInformation *prnInfo*) throws **PrinterException**

Parameter

prnInfo	Printer information type to retrieve See "**4.4.3**(4) **PrinterInformation**" for available settings and a list of printer information to retrieve. See "MP-A40 THERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to retrieve.

Return value Printer information

Exception **PrinterException**

PrinterException may be thrown when an error occurs while calling the method. For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description Based on the specification in *prnInfo*, sends printer command to the printer for responding the printer information, analyzes the response data by the response extension, and returns the type of the numeric array, numerical value, and character string.

In all **PrinterInformation**, the printer information can be retrieved in the numeric array by using the syntax (a).

In the following **PrinterInformation**, the printer information can be retrieved in the numerical value by using the syntax (b).

- **GET_NV_MEM_CAP**
- **GET_NV_MEM_REM_CAP**
- **GET_REM_USER_MEM_CAP**
- **GET_REM_USER_MEM_CAP_DEFRAG**
- **GET_PRN_ID_MODEL**
- **GET_PRN_ID_TYPE**
- **GET_PRN_ID_ROM_VER**
- **GET_PRN_ID_FIRM_CHECKSUM_BOOT**
- **GET_PRN_ID_FIRM_CHECKSUM_MAIN**
- **GET_PRN_ID_FIRM_CHECKSUM**
- **GET_MAINT_NUM_FEED_LINE**
- **GET_MAINT_NUM_HEAD_ACTIVE**
- **GET_MAINT_DRIVE_TIME**
- **GET_MAINT_NUM_FEED_LINE_INTEGRATION**
- **GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION**
- **GET_MAINT_DRIVE_TIME_INTEGRATION**
- **GET_HFONT_24_CHECKSUM**
- **GET_HFONT_16_CHECKSUM**
- **GET_FFONT_CHECKSUM**
- **GET_FFONT_DATA_SIZE**

In the following **PrinterInformation,** the printer information can be retrieved in the character string by using the syntax (c).

- **GET_NV_MEM_KEYCODE_LIST**
- **GET_PRN_ID_FIRM_VER_MAIN**
- **GET_PRN_ID_MFR**
- **GET_PRN_ID_MODEL_NAME**
- **GET_PRN_ID_FIRM_VER_BOOT**
- **GET_HFONT_24_ID**
- **GET_HFONT_24_INT_CHAR**
- **GET_HFONT_16_ID**
- **GET_HFONT_16_INT_CHAR**
- **GET_FFONT_LANG**
- **GET_FFONT_STANDARD**
- **GET_FFONT_COMPANY**

The timeout period specified in **setResponseTimeout** is valid.

This method is aborted by **reset**.

getVersion — Retrieve SDK version

Retrieves the SDK version as a character string.

Syntax public String **getVersion**()

Return value SDK version character string (Example: When the SDK version is Ver.1.0.0, the return value is "1.0.0")

Description This method can get the SDK version regardless of whether **isOpened** is true or false.

controlTransaction — Start/finish print data batch sending

Buffers the print data in the target method for batch process and sends the data in buffer to the printer.

Syntax public void **controlTransaction**(TransactionFunction *control*) throws **PrinterException**

Parameter *control* Operation selection of batch process
See "**4.4.3**(5) **TransactionFunction**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The target method for the batch process is as follows.

- **write**
- **selectStandardMode**
- **selectPageMode**
- **printPageModeData**
- **setPageModeVerticalPosition**
- **setStandardModeArea**
- **setStandardModeAlignment**
- **setHorizontalPosition**
- **setStandardModeBarcodeDirection**
- **setLineSpacing**
- **setCharacterRightSpace**
- **setCharacterFormatting**
- **printText**
- **printLogo**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printBarcode**
- **print2DCode**
- **printPageModeRectangle**
- **feedLine**
- **feedDotLine**
- **feedCutPosition**
- **feedMarkPosition**

By *control*, buffering of the print data in the target method for batch process is started or finished.

- Start buffering
  When this method with **TRANSACTION_START** in *control* is executed, the buffering of the print data in the target method for batch process is started. The print data in the target method for batch process executed during buffering is not sent to the printer but stored in the buffer. Any method other than the target method for batch process is immediately executed.

・Finish buffering
When this method with **TRANSACTION_PRINT** in *control* is executed, the buffering of the print data in the target method for batch process is finished, and the data in the buffer is sent to the printer.
When this method with **TRANSACTION_CLEAR** in *control* is executed, the buffering is interrupted and the data in the buffer is discarded.

When this method with **TRANSACTION_PRINT** or **TRANSACTION_CLEAR** in *control* is executed without starting buffering, an error occurs.

When this method with **TRANSACTION_START** in *control* is executed during buffering, the data in the buffer is discarded but the buffering is continued.

When no data exists in the buffer and this method with **TRANSACTION_PRINT** in *control* is executed, the data is not sent to the printer and the buffering is finished.

When this method with **TRANSACTION_PRINT** in *control* is being executed and this method with **TRANSACTION_START** in *control* is executed from another thread, the data sending by this method with **TRANSACTION_PRINT** in *control* is continued, this method with **TRANSACTION_START** in *control* is executed from another thread, and the buffering is newly started.

The maximum buffer size depends on the system.
When the data in the buffer exceeds the maximum size, the execution of the target method for batch process becomes error, but the data in the buffer is stored.

When the buffered data is sending by **TRANSACTION_PRINT**, the timeout period specified in **setWriteTimeout** is valid.

When an error occurs during sending data by **TRANSACTION_PRINT**, the remained data for sending is discarded.

When **reset** is executed during sending data by **TRANSACTION_PRINT**, the remained data for sending is discarded.

selectStandardMode — Select standard mode

Starts standard mode.

Syntax public void **selectStandardMode**() throws **PrinterException**

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description Right after execution of **open**, standard mode is selected.

When page mode is interrupted by **selectPageMode**, the page mode data is discarded.

selectPageMode — Start page mode

Starts page mode and specify the print area and the direction.

Syntax public void **selectPageMode**(int *startX*,
int *startY*,
int *width*,
int *height*,
Direction *direction*) throws **PrinterException**

Parameter

startX	Horizontal start point (dot) The range is 0 to 2399.
startY	Vertical start point (dot) The range is 0 to 2399.
width	Print area width (dot) The range is 1 to 2400.
height	Print area height (dot) The range is 1 to 2400.
direction	Print direction (dot) See "**4.4.3**(6) **Direction**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description When *startX* exceeds the print width, the print area setting by this method is disabled.
When (*startX* + *width*) exceeds the print width, *startX* is set to the specified value and *width* is set to (print width – *startX*).
When (*startY* + *height*) is 2400 or more, *startY* is set to the specified value and *height* is set to (2400 – *startY*).
The print width can be specified in the function setting. For the print width and the function setting, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Layout the character, image, barcode, and 2-dimensional barcode in the optional print position within the specified print area and configure the page mode data by **setHorizontalPosition** and **setPageModeVerticalPosition**.

When page mode is interrupted by **selectStandardMode**, the page mode data stored in the printer will be discarded.

When the printer is reset, the setting is back to standard mode.

printPageModeData — Print page mode data

Prints the page mode data.

Syntax public void **printPageModeData**() throws **PrinterException**

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description Execute this method after starting page mode by **selectPageMode** and configuring the page mode data within the print area.

This method is ignored in standard mode.

setPageModeVerticalPosition — Specify vertical absolute position in page mode

Specify the absolute position of print data in vertical direction within the print area in page mode.

Syntax public void **setPageModeVerticalPosition**(int *verticalPosition*) throws **PrinterException**

Parameter

verticalPosition	Data mapping starting position in vertical direction (dot) The range is 0 to 2399.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method. For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The start point varies depending on the setting of *direction* (print direction) in **selectPageMode**. For the print direction and the start point, see "**4.4.3**(6) **Direction**".

The setting of this method is disabled in standard mode.

When the specified value exceeds the print area, it is ignored.

When the printer is reset, the setting in this method is disabled.

setStandardModeArea — Specify print area in standard mode

Specify the print area in standard mode.

Syntax public void **setStandardModeArea**(int *leftMargin*,
int *prnAreaWidth*) throws **PrinterException**

Parameter

leftMargin	Left margin (dot) The range is 0 to 831.
prnAreaWidth	Print area width (dot) The range is 1 to 831.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method. For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The relation between left margin and print area width is shown in Figure 4-1. The print data is mapped to the shaded print area.

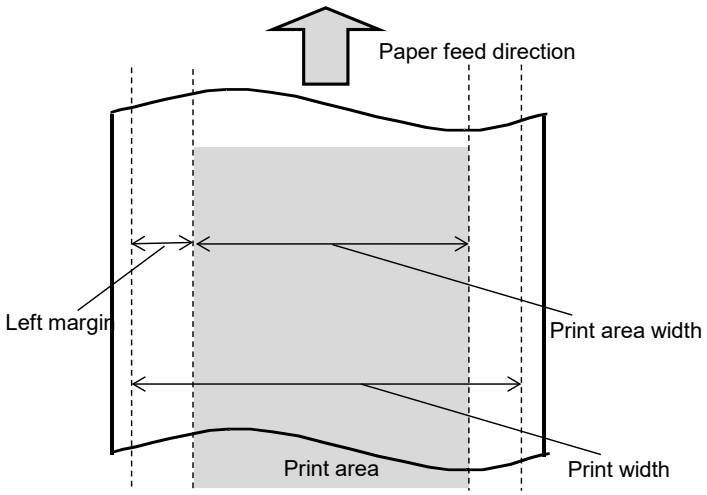


Figure 4-1

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the specified value in *leftMargin* exceeds the print width, the value is set to the print width. When the specified value in *prnAreaWidth* exceeds the print width, the value is set to the difference between the print width and the left margin.
The print width can be specified in the function setting. For the print width and the function setting, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When the printer is reset, *leftMargin* is set to 0 and *prnAreaWidth* is initialized to the print width.

setStandardModeAlignment — Alignment

Set the alignment in standard mode.

Syntax public void **setStandardModeAlignment**(Alignment *align*) throws **PrinterException**

Parameter *align* Alignment
See "**4.4.3**(7) **Alignment**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description When the print area is specified in **setStandardModeArea**, the position is aligned within the specified print area.

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the printer is reset, the setting in this method is back to the default value.

setHorizontalPosition — Specify absolute position

Specify the absolute horizontal position.

Syntax public void **setHorizontalPosition**(int *horizontalPosition*) throws **PrinterException**

Parameter *horizontalPosition* Horizontal print starting position (dot)
The range is 0 to 2399.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description In standard mode, the left margin is based on the one specified in **setStandardModeArea**.

In page mode, the start point changes depend on the specified print direction in *direction* of **selectPageMode**. For print direction and start point, see "**4.4.3**(6) **Direction**".

The value specified in **setStandardModeArea** and **selectPageMode** that exceeds the print area is ignored.

When the printer is reset, the setting in this method is disabled.

setStandardModeBarcodeDirection — Select print direction for barcode or 2-dimensional barcode

Specify the print direction for barcode or 2-dimensional barcode in standard mode.

Syntax public void **setStandardModeBarcodeDirection**(Rotate *rotate*) throws **PrinterException**

Parameter

rotate	Print direction See "**4.4.3**(15) **Rotate**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description By executing **printBarcode** or **print2DCode**, the setting of this method is back to the default value.

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When **ROTATE_90_TO_RIGHT** or **ROTATE_90_TO_LEFT** is specified in *rotate*, the available print width of barcode or 2-dimensional barcode is max. 300 mm. When the value exceeds 300 mm, barcode or 2-dimensional barcode is not printed.

When the printer is reset, the setting in this method is back to the default value.

setLineSpacing — Specify line space amount

Specify the line spacing.

Syntax public void **setLineSpacing**(int *lineSpacing*) throws **PrinterException**

Parameter

lineSpacing	Line spacing (dot) The range is 0 to 255.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The line spacing can be independently set in the standard mode and the page mode.

When the printer is reset, *lineSpacing* is set to 34.

setCharacterRightSpace — Specify character right space amount

Specify the character right space amount.

Syntax public void **setCharacterRightSpace**(int *space*) throws **PrinterException**

Parameter

space	Right space amount (dot) The range is 0 to 255.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description The right space amount can be independently set in the standard mode and the page mode.

This method is enabled for 1-byte characters and 2-byte characters. For 2-byte characters, the left space is set at 0.

When the printer is reset, *space* is set to 0.

selectCharacterSet — Select character set

Select the character set.

Syntax public void **selectCharacterSet**(CharacterSet *charSet*) throws **PrinterException**

Parameter *charSet* Character set
See "**4.4.3**(8) **CharacterSet**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description According to the language setting in Android device, the default setting of the character set is as follows.

Japanese:	**CODEPAGE_KATAKANA**
Other than Japanese:	**CODEPAGE_1252**

When this method is executed and then the printer is reset by **reset** or turned off, the setting in this method is enabled.

getCharacterSet — Retrieve specified character set

Retrieves the specified character set.

Syntax public CharacterSet **getCharacterSet**()

Return value Character set

Description For the available setting to retrieve, see "**4.4.3**(8) **CharacterSet**".

selectInternationalCharacterSet — Select international character set

Select the international character set.

Syntax public void **selectInternationalCharacterSet**(InternationalCharacterSet *intCharSet*) throws **PrinterException**

Parameter *intCharSet* International character set
See "**4.4.3**(9) **InternationalCharacterSet**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description According to the language setting in Android device, the default setting of the International character set is as follows.

Japanese:	**INT_CHAR_SET_JAPAN**
Other than Japanese:	**INT_CHAR_SET_USA**

When this method is executed and then the printer is reset by **reset** or turned off, the setting in this method is enabled.

getInternationalCharacter — Retrieve specified international character set

Retrieves the specified international character set.

Syntax public InternationalCharacterSet **getInternationalCharacter**()

Return value International character set

Description For the available setting to retrieve, see "**4.4.3**(9) **InternationalCharacterSet**".

setCharacterFormatting — Format character

Set the character formatting.

Syntax public void **setCharacterFormatting**(CharacterType *type*,
CharacterScale *verticalScale*,
CharacterScale *horizontalScale*,
Underline *underline*,
Bold *bold*,
Reverse *reverse*,
Rotate *rotate*) throws **PrinterException**

Parameter

type — Character font
See "**4.4.3**(10) **CharacterType**" for available settings.

verticalScale — Character size (in vertical direction)
See "**4.4.3**(11) **CharacterScale**" for available settings.

horizontalScale — Character size (in horizontal direction)
See "**4.4.3**(11) **CharacterScale**" for available settings.

underline — Underline
See "**4.4.3**(12) **Underline**" for available settings.

bold — Bold print
See "**4.4.3**(13) **Bold**" for available settings.

reverse — Reverse print
See "**4.4.3**(14) **Reverse**" for available settings.

rotate — Character Rotation Print
In page mode, the setting in rotate is disabled. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.
In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.
In standard mode, when **ROTATE_90_TO_RIGHT** or **ROTATE_90_TO_LEFT** is specified, the operation is as follows:
- Underline print is not performed.
- the scale of height and width in *verticalScale* and *horizontalScale* are reversed.

See "**4.4.3**(15) **Rotate**" for available settings.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description When the printer is reset, the setting in this method back to the default value.

printText — Send text data

Sends the text data.

Syntax public void **printText**(String *text*) throws **PrinterException**

Parameter *text* Text data to send to the printer

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description This method sends the text data specified in *text* after encoding it to recognizable text data for the printer based on the character set specified in **selectCharacterSet**.

Based on the setting in **selectCharacterSet** and **selectInternationalCharacterSet**, printer commands "Character Code Table Select", "International Character Select", "Kanji Code System Selection", and "Kanji Mode Cancel" are added to the send data as the header. For the details of printer commands, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method does not add any line feed code in the last of the text data.

The maximum data size that can be specified is 16 KB (16384 bytes).

When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout**.

This method is aborted by **reset**.

printLogo — Print logo registered in printer

Prints the logo registered in the printer.

Syntax public void **printLogo**(int *regNum*) throws **PrinterException**

Parameter *regNum* Logo number
The range is 0 to 99.

Exception **PrinterException**
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description Register the logo by **registerLogo**.

When the specified logo is not registered in *regNum*, this method is ignored.

sendDataFile — Send file data

Sends the file data to the printer.
In syntax (a), dithering is fixed to be enabled.
In syntax (b), dithering can be specified.

Syntax (a) public void **sendDataFile**(String *filePath*) throws **PrinterException**

(b) public void **sendDataFile**(String *filePath*)
Dithering *dithering*) throws **PrinterException**

Parameter

filePath — File path to send to the printer
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.

- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

dithering — Dithering
It is enabled when the extension of the file specified at *filePath* is .bmp, .jpg, .jpeg, or .png.
See "**4.4.3**(3) **Dithering**" for available settings.

Exception

PrinterException
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description

Depending on the specified file extension, the followings are processed.

・When the file extension is .bmp, .jpg, .jpeg, or .png:
The image data is converted to printable format for the printer and is sent to the printer.
The color image data is converted to monochrome image by binarization.

・When the file extension is .txt:
Text data format supports UTF-8.
This method sends the text data after encoding it to recognizable text data for the printer based on the character set specified in **selectCharacterSet**.
Based on the setting in **selectCharacterSet** and **selectInternationalCharacterSet**, printer commands "Character Code Table Select", "International Character Select", "Kanji Code System Selection", and "Kanji Mode Cancel" are added to the send data as the header.
For the details of printer commands, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".
This method does not add any line feed code in the last of the text data.

・When the file extension is .bin, or.dat:
The text data is sent to printer without conversion.

・When the file extension is htm, or .html:
The text data is sent to printer without conversion.
Be sure to add an HTML end tag in the last of file data.

The maximum file size that can be specified is 16 MB (16777216 bytes).

When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout**.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send. When perform Hardware Reset, use **reset**.
For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method is aborted by **reset**.

sendImage — Send image data

Sends the image data to the printer.

Syntax
public void **sendImage**(Bitmap *bitmap*
Dithering *dithering*) throws **PrinterException**

Parameter

bitmap	"android.graphics.Bitmap" type object to be sent to the printer Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and registered.
dithering	Dithering See "**4.4.3**(3) **Dithering**" for available settings.

Exception
PrinterException
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description
When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout**.

This method is aborted by **reset**.

printPDF — Print PDF page

Prints the specified page of the selected PDF file.

Syntax
public void **printPDF**(String *filePath*,
int *startIndex*,
int *endIndex*,
Rotate *rotate*,
int *width*,
int *bottomMargin*,
Dithering *dithering*) throws **PrinterException**

Parameter

filePath	File path of PDF
startIndex	Start number of the printed page The range is -1, 1 to the number of pages in the PDF file.
endIndex	End number of the printed page The range is 1 to 2147483647.
rotate	Rotation direction of the image See "**4.4.3**(15) **Rotate**" for available settings.
width	Width of the image (dot) The range is -1, 1 to 832.
bottomMargin	Bottom margin (dot) The range is -1, 0 to 2400.
dithering	Dithering See "**4.4.3**(3) **Dithering**" for available settings.

Exception
PrinterException
PrinterException may be thrown when an error occurs while calling the method.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description PDF is converted to printable format for the printer and is sent to the printer.

The file extension for supporting PDF is .pdf.
The maximum file size that can be specified is 16 MB (16777216 bytes).
The color image data is converted to monochrome image by binarization.

When -1 is specified for *startIndex*, all pages are printed.
When -1 is specified for *startIndex*, the value of *endIndex* is ignored.
When 1 or more is specified for *startIndex*, pages from the page number specified in *startIndex* to the page number specified in *endIndex* are printed.
Up to 100 pages of PDF can be printed at a time regardless of the page number to be specified.
When a value more than the value specified for *endIndex* is specified for *startIndex*, an error occurs.
When a value more than the number of pages in the PDF file is specified for *endIndex*, pages from the page number specified in *startIndex* to the last page of the PDF file are printed.
When an out-of-range value is specified for *startIndex* or *endIndex*, an error occurs.
When -1 is specified for *width*, size conversion is not performed.
When a value between 1 and 832 is specified for *width*, the image width is converted to the specified value while maintaining the aspect ratio.
When -1 is specified for *bottomMargin*, the image is created and printed while maintaining the bottom margin.
When a value between 0 and 2400 is specified for *bottomMargin*, the bottom margin is changed to the specified size.

When a value between 0 and 2400 is specified for *bottomMargin*, blank pages are not printed.
When marked paper is selected, marked paper form feed is performed after printing the PDF.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about how to use the marked paper.

When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout** for each page printed.

This method is aborted by **reset**.

Note
- This method is not supported in page mode.
- No guarantee of printing operation when more than 100 pages are printed at a time.

printBarcode — Print barcode

Prints the barcode.

Syntax (a) public void **printBarcode**(TypeBarcode *type*,
String *text*,
ModuleWidthBarcode *moduleWidth*,
int *moduleHeight*,
HriPositionBarcode *hri*,
NwRatioBarcode *nwRatio*) throws **PrinterException**

(b) public void **printBarcode**(TypeBarcode *type*,
byte[] *binary*,
ModuleWidthBarcode *moduleWidth*,
int *moduleHeight*,
HriPositionBarcode *hri*,
NwRatioBarcode *nwRatio*) throws **PrinterException**

Parameter *type* Barcode type
See "**4.4.3**(16) **TypeBarcode**" for available settings.
(a) *text* : Barcode text data
(b) *binary* : Barcode binary data

The available values in *text* and *binary* are as follows.

Barcode Type	Data Size	Barcode Data to Input	
UPC-A	11 to 12 bytes	'0' to '9'	(0x30 to 0x39)
UPC-E	11 to 12 bytes	'0' to '9'	(0x30 to 0x39)
JAN13 (EAN13)	12 to 13 bytes	'0' to '9'	(0x30 to 0x39)
JAN8 (EAN8)	7 to 8 bytes	'0' to '9'	(0x30 to 0x39)
CODE39	1 to 150 bytes	' '	(0x20)
		'$'	(0x24)
		'%'	(0x25)
		'+'	(0x2B)
		'-'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' to '9'	(0x30 to 0x39)
		'A' to 'Z'	(0x41 to 0x54)
ITF	2 to 15 bytes (Even number only)	'0' to '9'	(0x30 to 0x39)
CODABAR	1 to 150 bytes	'$'	(0x24)
		'+'	(0x2B)
		'-'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' to '9'	(0x30 to 0x39)
		':'	(0x3A)
		'A' to 'D'	(0x41 to 0x44)
CODE93	1 to 150 bytes		(0x00 to 0x7F)
CODE128	2 to 150 bytes		(0x00 to 0x7F)*1
JAN13 (EAN13) add-on 2	14 to 15 bytes	'0' to '9'	(0x30 to 0x39)
JAN13 (EAN13) add-on 5	17 to 18 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Omni-directional	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Truncated	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Limited	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Expanded	2 to 150 bytes	' 'to '"'	(0x20 to 0x22)
		'%' to '?'	(0x25 to 0x3F)
		'A' to 'Z'	(0x41 to 0x5A)
		'_'	(0x5F)
		'a' to 'z'	(0x61 to 0x7A)
		'{'	(0x7B)

*1: The available barcode data varies depending on the specified code set.
CODE A : 0x00 to 0x50

CODE B : 0x00 to 0x7F
CODE C : 2 digits numbers from 00 to 99 (0x00 to 0x63)

moduleWidth Barcode module width or fine element
See "**4.4.3**(17) **ModuleWidthBarcode**".

moduleHeight Barcode module height(dot)
See the following table for available settings.

Barcode Type	***moduleHeight*** **(dot)**
UPC-A	0: Initial value (162 dots), 1 to 255
UPC-E	
JAN13 (EAN13)	
JAN8 (EAN8)	
CODE39	
ITF	
CODABAR	
CODE93	
CODE128	
JAN13 (EAN13) add-on 2	
JAN13 (EAN13) add-on 5	
GS1 Databar Omni-directional	0: Initial value (162 dots), 66 to 255
GS1 Databar Truncated	0: Initial value (162 dots), 26 to 255
GS1 Databar Limited	0: Initial value (162 dots), 20 to 255
GS1 Databar Expanded	0: Initial value (162 dots), 68 to 255

hri Barcode HRI character
See "**4.4.3**(18) **HriPositionBarcode**" for available settings.

nwRatio Barcode N:W ratio
The available setting varies depending on the setting of *type*.
For the details, see "**4.4.3**(19) **NwRatioBarcode**".

Exception **PrinterException**
PrinterException may be thrown when the method is called. For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the barcode width exceeds the print area, the barcode is not printed.

This method sends the text data specified in syntax (a) *text* after encoding it to recognizable text data for the printer.
The binary data specified in syntax (b) *binary* is sent to the printer as it is.

Special codes specified in CODE128 are as follows.

Input Data	Special Code per Code Set		
	CODE A	**CODE B**	**CODE C**
'{S' (0x7B, 0x53)	SHIFT	SHIFT	SHIFT
'{A' (0x7B, 0x41)	-	CODE A	CODE A
'{B' (0x7B, 0x42)	CODE B	-	CODE B

Input Data	Special Code per Code Set		
	CODE A	CODE B	CODE C
'{C' (0x7B, 0x43)	CODE C	CODE C	-
'{1' (0x7B, 0x31)	FNC1	FNC1	FNC1
'{2' (0x7B, 0x32)	FNC2	FNC2	FNC2
'{3' (0x7B, 0x33)	FNC3	FNC3	FNC3
'{4' (0x7B, 0x34)	FNC4	FNC4	FNC4
'{{' (0x7B, 0x7B)	'{'	'{'	-

Check digits for the following barcode are automatically calculated.
・UPC-A
・UPC-E
・JAN13 (EAN13)
・JAN8 (EAN8)
・JAN13 (EAN13) add-on 2
・JAN13 (EAN13) add-on 5
・GS1 Databar Omni-directional
・GS1 Databar Truncated
・GS1 Databar Limited

When the data including the check digit is specified for the following barcodes, the check digit is ignored and recalculated on the printer.
・UPC-A
・UPC-E
・JAN13 (EAN13)
・JAN8 (EAN8)

Specifying the start code and stop code ('*') of CODE39 is not required since the codes are added automatically.

When odd number of data is specified in ITF barcode data, the last data is discarded.

In CODABAR barcode data, input 'A' to 'D' as the start code and stop code.

In GS1 Databar Expanded barcode data, use '{' only for specifying FNC1. When specify FNC1, input '{1'(0x7B, 0x31).

When specified value in any one of *text*, *binary*, *moduleWidth*, *moduleHeight*, or *nwRatio* is not corresponded to the barcode type specified in *type*, an error occurs.

When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout**.

print2DCode — Print 2-dimensional barcode

Prints the 2-dimensional barcode.

Syntax (a) public void **print2DCode**(Type2DCode *type*,
String *text*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
int *moduleHeight*,
int *column*,
int *row*,
ErrorCorrect2Dcode *errorCorrect*) throws **PrinterException**

(b) public void **print2DCode**(Type2DCode *type*,
byte[] *binary*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
int *moduleHeight*,
int *column*,
int *row*,
ErrorCorrect2Dcode *errorCorrect*) throws **PrinterException**

Parameter

type
2-dimensional barcode type
See "**4.4.3**(20) **Type2DCode**" for available settings.
(a) *text* : Barcode text data
(b) *binary* : Barcode binary data

mode
2-dimensional barcode mode
The available setting varies depending on the setting of *type*.
For the details, see "**4.4.3**(21) **Mode2Dcode**".
When select **MODE_2DCODE_MAXI_CODE_2**, add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) data in the beginning of *text* or *binary*.
When select **MODE_2DCODE_MAXI_CODE_3**, add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) data in the beginning of *text* or *binary*.

moduleSize
2-dimensional barcode module size
The available setting varies depending on the setting of *type*.
For the details, see "**4.4.3**(22) **ModuleSize2DCode**".

moduleHeight
2-dimensional barcode module height (dot)
moduleHeight is enabled in PDF417 and GS1 Databar Stacked Omni-directional.
・PDF417
Specifies the row height of PDF. The valid range is 2 to 127.
・GS1 Databar Stacked Omni-directional
Specifies the height of 1 row. The valid range is 33 to 255.

column
Number of columns in 2-dimensional barcode
column is enabled in PDF417and GS1 Databar Expanded Stacked
・PDF417
Specifies the number of columns in the data area.
The valid range is 0 to 30.
When 0 is specified, the number of column is set automatically.
・GS1 Databar Expanded Stacked
Specifies the number of the element in 1 line.
The valid range is 2 to 20.

row
Number of rows in 2-dimensional barcode
row is enabled in PDF417 only. The valid range is 0, and 3 to 90.
When 0 is specified, the number of row is set automatically.

errorCorrect
Error correction level of 2-dimensional barcode
The available setting varies depending on the setting of *type*.
For the details, see "**4.4.3**(23) **ErrorCorrect2DCode**".

Exception

PrinterException
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description

In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

This method sends the text data specified in syntax (a) *text* after encoding it to recognizable text data for the printer.
In *text*, input the corresponded character data in ASCII style, according to the standard 2-dimensional barcode type specified in *type*.
The binary data specified in syntax (b) *binary* is sent to the printer as it is.
In *binary*, input the corresponded character data in ASCII style character code, according to the standard 2-dimensional barcode type specified in *type*.

When specified value in any one of *text*, *binary*, *mode*, *moduleSize*, *moduleHeight*, *column*, or *errorCorrect* is not corresponded to the 2-dimensional barcode type specified in *type*, an error occurs.

When the 2-dimensional barcode width exceeds the print area width, this method is ignored.

When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout**.

printPageModeRectangle — Draw rectangular

Draws a rectangle in page mode.

Syntax

public void **printPageModeRectangle**(int *startX*,
int *startY*,
int *width*,
int *height*,
int *thickness*) throws **PrinterException**

Parameter

startX	Horizontal start point (dot) The range is 0 to 2395.
startY	Vertical start point (dot) The range is 0 to 2395.
width	Width (dot) The range is 4 to 2400.
height	Height (dot) The range is 4 to 2400.
thickness	Line width (dot) The range is 2 to 40.

Exception

PrinterException
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description

Do not execute this method in standard mode.

The start point varies depending on the print direction in page mode specified in *direction* of **selectPageMode**.
For the details of print direction and start point, see "**4.4.3**(6) **Direction**".

When the setting in *startX*, *startY*, *width*, or *height* exceeds the print area in page mode specified in **selectPageMode**, the exceeded area in print area is not drawn.

In this method, the printer command "Ruled Line OFF" and "Ruled Line Buffer Clear" are sent, and the ruled line is changed to OFF and ruled line buffer is cleared. For the details of printer command, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

feedLine — Feed paper by line

Feeds the paper by line.

Syntax: public void **feedLine**(int *lines*) throws **PrinterException**

Parameter: *lines* — Number of lines to feed (line)
The range is 0 to 255.

Exception: **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description: In standard mode, feeds the paper by the specified number of lines.
When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.
In page mode, the vertical data mapping starting position is shifted by the specified number of lines.

The paper feed length for 1 line (line feed space) can be specified in **setLineSpacing**.

feedDotLine — Feed paper by dot

Feeds the paper by dot.

Syntax: public void **feedDotLine**(int *dotLines*) throws **PrinterException**

Parameter: *dotLines* — Number of dots to feed (dot)
In the standard mode, the range is -48 to 8192.
In page mode, the range is 0 to 8192. When the specified value is within -48 to -1, this method is ignored.

Exception: **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description: In standard mode, feeds the paper by the specified number of dots.
When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.
In page mode, the vertical data mapping starting point is shifted by the specified number of dots.

feedCutPosition — Feed paper to cut position

Feeds the paper to the paper cut position.

Syntax: public void **feedCutPosition**() throws **PrinterException**

Exception: **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description: In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

feedMarkPosition — Marked paper form feed

Performs the marked paper form feed and the form feed position correct.

Syntax public void **feedMarkPosition**(int *dotLines*) throws **PrinterException**

Parameter

dotLines	Correction amount (dot) The range is -48 to 255. When correction is not needed, specify 0.

Exception **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**.

Description In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

This method is effective only when marked paper is selected.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about how to use the marked paper.

setLog — Specify log output

Specify the log output.

Syntax public void **setLog**(int *logLevel*, LogFileSize *logFileSize*) throws **PrinterException**

Parameter

logLevel	Log output level Specify 0. When other than 0 is specified, the contents of output log is not guaranteed.
logFileSize	Maximum size of log file See "**4.4.3**(24) **LogFileSize**" for the available setting.

Exception **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description Log file is saved as a local file of Android application with programmed this library.

Log file name: PrinterManager.log.x (x range is 0 to 4)

First of all, the log file is created called PrinterManager.log.0.
When the file size exceeds the maximum size of the log file, changes the file name to PrinterManager.log.1, and creates the new file called PrinterManager.log.0.
The number of created file is up to 5.

(2) **PrinterStatus**

- **Summary**

This class stores the printer status.
This class provides following functions.

Public Methods

Method	Function Summary
getErrOffline	Retrieve offline error status
getErrHardware	Retrieve hardware error status
getErrVoltage	Retrieve Vp Voltage error status
getErrHeadTemperature	Retrieve head temperature error status
getErrOutOfPaper	Retrieve out-of-paper error status
getErrMarkPaperJam	Retrieve paper jam error status while detecting marked paper.
getErrCoverOpen	Retrieve paper cover open error status
getErrBattery	Retrieve battery error status
getStateFeedSwitch	Retrieve feed switch status
getStatePaperFeed	Retrieve paper feed status
getStateReturnWaiting	Retrieve return waiting status
getStateFlashMemoryRewriting	Retrieve FLASH memory rewrite status
getStateBattery	Retrieve battery voltage status

- **Public Methods**

getErrOffline — Retrieve offline error status

Retrieves the offline error status.

Syntax public boolean **getErrOffline**()

Return value true : Offline error
false : Online

getErrHardware — Retrieve hardware error status

Retrieves the hardware error status.

Syntax public boolean **getErrHardware**()

Return value true : Error
false : OK

getErrVoltage — Retrieve Vp Voltage error status

Retrieves the Vp Voltage error status.

Syntax public boolean **getErrVoltage**()

Return value true : Error
false : OK

getErrHeadTemperature — Retrieve head temperature error status

Retrieves the head temperature error status.

Syntax public boolean **getErrHeadTemperature**()

Return value true : Error
false : OK

getErrOutOfPaper — Retrieve out-of-paper error status

Retrieves the out-of-paper error status.

Syntax public boolean **getErrOutOfPaper**()

Return value true : Error
false : OK

getErrMarkPaperJam — Retrieve paper jam error status while detecting marked paper

Retrieves the paper jam error status while detecting marked paper.

Syntax public boolean **getErrMarkPaperJam**()

Return value true : Error
false : OK

getErrCoverOpen — Retrieve paper cover open error status

Retrieves the paper cover open error status.

Syntax public boolean **getErrCoverOpen**()

Return value true : Error
false : OK

getErrBattery — Retrieve battery error status

Retrieves the battery error status.

Syntax public boolean **getErrBattery**()

Return value true : Error
false : OK

getStateFeedSwitch — Retrieve feed switch status

Retrieves the feed switch status.

Syntax public boolean **getStateFeedSwitch**()

Return value
true : ON
false : OFF

getStatePaperFeed — Retrieve paper feed status

Retrieves the paper feed status.

Syntax public boolean **getStatePaperFeed**()

Return value
true : Feeding
false : Stopped

getStateReturnWaiting — Retrieve return waiting status

Retrieves the return waiting status.

Syntax public boolean **getStateReturnWaiting**()

Return value
true : In return waiting status
false : Not in return waiting status

getStateFlashMemoryRewriting — Retrieve FLASH memory rewrite status

Retrieves the FLASH memory rewrite status.

Syntax public boolean **getStateFlashMemoryRewriting**()

Return value
true : In rewrite status
false : Not in rewrite status

getStateBattery — Retrieve battery voltage status

Retrieves the battery voltage status.

Syntax public int **getStateBattery**()

Return value
0 : No battery
1 : Battery remaining capacity level 4 or 5
(Need to charge or battery remaining capacity is 0%)
2 : Battery remaining capacity level 3 (battery remaining capacity: approx. 10%)
3 : Battery remaining capacity level 2 (battery remaining capacity: approx. 40%)
4 : Battery remaining capacity level 1 (battery remaining capacity: approx. 80%)

(3) **PrinterDiscovery**

- **Summary**

This class provides printer search functions.
This class provides the following functions.

Public Methods

Method	Function Summary
PrinterDiscovery	Constructor
startDiscoveryPrinter	Start printer search
cancelDiscoveryPrinter	Cancel printer search
getFoundPrinter	Retrieve information list of searched printer

- **Public Methods**

PrinterDiscovery — Constructert

Constructor for **com.seikoinstruments.sdk.mobileprinter.PrinterManager** class.

Syntax public **PrinterDiscovery**(Context *context*)

Parameter

context	Android application context to call this method Specify the context that can be retrieved by **getApplicationContext**().

startDiscoveryPrinter — Start printer search

Starts the printer search.

Syntax public void **startDiscoveryPrinter**(PrinterListener *listener*,
PrinterInterface *prnIf*,
int *retry*,
int *timeout*) throws **PrinterException**

Parameter

listener	Instance of **PrinterListener**
prnIf	Constants of communication interface See "**4.4.3**(1) **PrinterInterface**" for the available setting. **PRN_IF_USB** is not supported.
retry	Number of retry times (time) The performance varies depending on the setting of *prnIf*. ・When **PRN_IF_TCP** is specified: Sends local broadcast packet as much as the specified times in *retry*. The range is 1 to 5. When the specified value is below 1, the value is processed as 1. When the specified value exceeds 5, the value is processed as 5. ・When **PRN_IF_BT** is specified: The specified value is ignored, and the search is performed 1 time only.

timeout Timeout period for 1 time search
The performance varies depending on the setting of *prnIf*.
・When **PRN_IF_TCP** is specified:
Every time of sending local broadcast packet, wait for a response from the printer until the time specified by *timeout*.
The range is 3 to 60.
When the specified value is below 3, the value is processed as 3.
When the specified value exceeds 60, the value is processed as 60.
・When **PRN_IF_BT** is specified:
The specified value is ignored.

Exception **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description Cancel is available by **cancelDiscoveryPrinter**. When search is canceled or completed, **onDiscoveryFinished** of the instance specified in *listener* is executed.

Retrieve the search result by **getFoundPrinter**.

cancelDiscoveryPrinter Cancel printer search

Cancels the printer search started by **startDiscoveryPrinter**.

Syntax public void **cancelDiscoveryPrinter**() throws **PrinterException**

Exception **PrinterException**
PrinterException may be thrown when the method is called.
For the details of **PrinterException**, see "**4.4.4**(1) **PrinterException**".

Description When the search is canceled, an instance of **onDiscoveryFinished** specified in *listener* of **startDiscoveryPrinter** is executed.

The search result by cancellation can be retrieved by **getFoundPrinter**.

getFoundPrinter Retrieve information list of searched printer

Retrieves all the printer information searched by **startDiscoveryPrinter**.

Syntax public ArrayList<DiscoveredPrinter> **getFoundPrinter**()

Return value Returns the searched printer information by ArrayList of **DiscoveredPrinter** class. By method of **DiscoveredPrinter** class, the printer model name, Bluetooth address, IP address, MAC address or serial number can be retrieved.
For the method of **DiscoveredPrinter** class, see "**4.4.2**(4) **DiscoveredPrinter**".

(4) **DiscoveredPrinter**

- **Summary**

  This class stores the printer information searched by **startDiscoveryPrinter**. By the printer information searched, the printer model name, Bluetooth address, IP address, MAC address and serial number can be retrieved.
  This class provides the following functions.

  Public Methods

Method	Function Summary
getPrinterModel	Retrieve printer model name
getBluetoothAddress	Retrieve Bluetooth address
getIpAddress	Retrieve IP address
getMacAddress	Retrieve MAC address
getSerialNumber	Retrieve serial number

- **Public Methods**

getPrinterModel — Retrieve printer model name

Retrieves the character string of printer model name.

Syntax public String **getPrinterModel**()

Return value Printer model name

getBluetoothAddress — Retrieve Bluetooth address

Retrieves the character string of Bluetooth address.

Syntax public String **getBluetoothAddress**()

Return value Bluetooth address

getIpAddress — Retrieve IP address

Retrieves the character string of IP address.

Syntax public String **getIpAddress**()

Return value IP address

getMacAddress — Retrieve MAC address

Retrieves the character string of MAC address.

Syntax public String **getMacAddress**()

Return value MAC address

getSerialNumber — Retrieve Serial Number

Retrieves the character string of Serial Number.

Syntax public String **getSerialNumber**()

Return value Serial Number

4.4.3 Enumerated Type (Enum)

(1) **PrinterInterface**

Enumerator used for specifying the communication interface of **open**.

Constant Name	Description
PRN_IF_TCP	TCP/IP
PRN_IF_BT	Bluetooth
PRN_IF_USB	USB

(2) **PrinterModel**

Enumerator used for specifying the printer model of **open**.

Constant Name	Description
PRN_MODEL_MP_A40	MP-A40

(3) **Dithering**

Enumerator used for dithering of **resisterLogo**, **sendDataFile**, and **printPDF**.

Constant Name	Description
DITHERING_DISABLE	Dithering is disabled
DITHERING_ERRORDIFFUSION	Dithering is enabled

(4) **PrinterInformation**

Enumerator used for specifying the printer information of **getPrinterInformation**, **getPrinterInformationNumber** and **getPrinterInformationString**.
See "MP-A40 THERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to retrieve.

Constant Name	Description (Printer Information)
GET_NV_MEM_CAP	NV graphics memory capacity
GET_NV_MEM_REM_CAP	NV graphics memory remaining capacity
GET_NV_MEM_KEYCODE_LIST	NV graphics key code list
GET_REM_USER_MEM_CAP_DEFRAG	Remaining user area after defragment
GET_REM_USER_MEM_CAP	Remaining user area
GET_FUNC_SET_RESP	Function setting response
GET_PRN_ID_MODEL	Printer ID send (Model ID)
GET_PRN_ID_TYPE	Printer ID send (Type ID)
GET_PRN_ID_ROM_VER	Printer ID send (ROM version ID)
GET_PRN_ID_FIRM_VER_MAIN	Printer ID send (Firmware version (main))
GET_PRN_ID_MFR	Printer ID send (Manufacturer)
GET_PRN_ID_MODEL_NAME	Printer ID send (Model name)

Constant Name	Description (Printer Information)
GET_PRN_ID_FIRM_VER_BOOT	Printer ID send (Firmware version (boot))
GET_PRN_ID_FIRM_CHECKSUM_BOOT	Printer ID send (Firmware checksum (boot))
GET_PRN_ID_FIRM_CHECKSUM_MAIN	Printer ID send (Firmware checksum (main))
GET_PRN_ID_FIRM_CHECKSUM	Printer ID send (Firmware checksum (main+boot))
GET_MAINT_NUM_FEED_LINE	Maintenance counter (Paper feed line count (in 100 dot-lines))
GET_MAINT_NUM_HEAD_ACTIVE	Maintenance counter (Number of thermal head activation times (in 100 dot-lines))
GET_MAINT_DRIVE_TIME	Maintenance counter (Drive time of printer mechanism (minute))
GET_MAINT_NUM_FEED_LINE_INTEGRATION	Maintenance counter (Paper feed line count (in 100 dot-lines) (integrated value))
GET_MAINT_ NUM_HEAD_ACTIVE_INTEGRATION	Maintenance counter (Number of thermal head activation times (in 100 dot-lines) (integration))
GET_MAINT_DRIVE_TIME_INTEGRATION	Maintenance counter (Drive time of printer mechanism (minute) (integration))
GET_HFONT_24_CHECKSUM	1-byte font ID send (24 dots font, checksum)
GET_HFONT_24_ID	1-byte font ID send (24 dots font, ID)
GET_HFONT_24_INT_CHAR	1-byte font ID send (24 dots font, registered international character)
GET_HFONT_16_CHECKSUM	1-byte font ID send (16 dots font, checksum)
GET_HFONT_16_ID	1-byte font ID send (16 dots font, ID)
GET_HFONT_16_INT_CHAR	1-byte font ID send (16 dots font, registered international character)
GET_FFONT_LANG	2-byte font ID send (Language)
GET_FFONT_STANDARD	2-byte font ID send (Standard)
GET_FFONT_COMPANY	2-byte font ID send (Company name)
GET_FFONT_CHECKSUM	2-byte font ID send (Checksum)
GET_FFONT_DATA_SIZE	2-byte font ID send (Data size)

(5) **TransactionFunction**

Enumerator used for specifying the batch process control method of **controlTransaction**.

Constant Name	Description
TRANSACTION_CLEAR	Interrupt batch process
TRANSACTION_START	Begin batch process
TRANSACTION_PRINT	End batch print and batch process

(6) **Direction**

Enumerator used for specifying the print direction of **selectPageMode**. The default value is shown by the shaded enumerated value in the following table.
When the start point is "top left" or "bottom right", the printer maps the print data in a vertical direction to the paper feed direction.
When the start point is "top right" or "bottom left", the printer maps the print data in a parallel direction to the paper feed direction.

Constant Name	Description
DIRECTION_LEFT_TO_RIGHT	Print direction: left to right, Start point: top left (Figure 4-2 A)
DIRECTION_BOTTOM_TO_TOP	Print direction: bottom to top, Start point: bottom left (Figure 4-2 B)
DIRECTION_RIGHT_TO_LEFT	Print direction: right to left, Start point: bottom right (Figure 4-2 C)
DIRECTION_TOP_TO_BOTTOM	Print direction: top to bottom, Start point: top right (Figure 4-2 D)

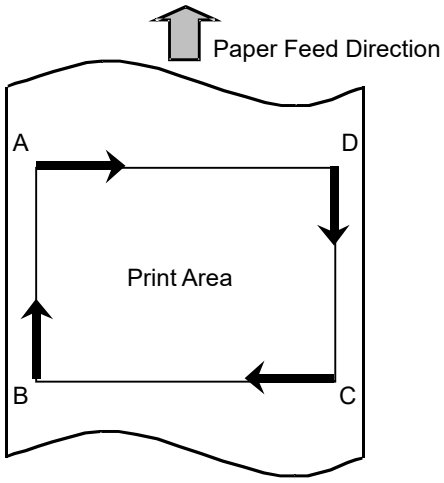


Figure 4-2

(7) **Alignment**

Enumerator used for specifying the print position of **setStandardModeAlignment**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
ALIGN_LEFT	Aligned left
ALIGN_CENTER	Centered
ALIGN_RIGHT	Aligned right

(8) **CharacterSet**

Enumerator used for specifying the character set of **selectCharacterSet** and **getCharacterSet**. The default value varies depending on the language setting of Android device.

Japanese: **CODEPAGE_KATAKANA**
Other than Japanese: **CODEPAGE_1252**

Constant Name	Description
CODEPAGE_437	USA, Standard Europe (Code Page 437)
CODEPAGE_KATAKANA	Katakana
CODEPAGE_850	Multilingual (Code Page 850)
CODEPAGE_860	Portuguese (Code Page 860)
CODEPAGE_863	Canadian-French (Code Page 863)
CODEPAGE_865	Nordic (Code Page 865)
CODEPAGE_857[*1]	Turkish (Code Page 857)
CODEPAGE_737	Greek (Code Page 737)
CODEPAGE_1252	Latin (Code Page 1252)
CODEPAGE_866	Russian (Code Page 866)
CODEPAGE_852	Eastern Europe (Code Page 852)
CODEPAGE_858	Euro (Code Page 858)
CODEPAGE_855	Cyrillic (Code Page 855)
CODEPAGE_864[*1]	Arabic (Code Page 864)
CODEPAGE_1250	Central European (Code Page 1250)
CODEPAGE_1251	Cyrillic (Code Page 1251)
CODEPAGE_1253	Greek (Code Page 1253)
CODEPAGE_1254	Turkish (Code Page 1254)

*1: 20ACh of the Unicode cannot be printed.

(9) **InternationalCharacterSet**

Enumerator used for specifying the international character set of **selectInternationalCharacterSet** and **getInternationalCharacter.**
The default value varies depending on the language setting of Android device.

Japanese: **INT_CHAR_SET_JAPAN**
Other than Japanese: **INT_CHAR_SET_USA**

Constant Name	Description
INT_CHAR_SET_USA	USA
INT_CHAR_SET_FRANCE	France
INT_CHAR_SET_GERMANY	Germany
INT_CHAR_SET_UNITED_KINGDOM	United Kingdom
INT_CHAR_SET_DENMARK_1	Denmark
INT_CHAR_SET_SWEDEN	Sweden
INT_CHAR_SET_ITALY	Italy
INT_CHAR_SET_SPAIN_1	Spain I
INT_CHAR_SET_JAPAN	Japan
INT_CHAR_SET_NORWAY	Norway
INT_CHAR_SET_DENMARK_2	Denmark II
INT_CHAR_SET_SPAIN_2	Spain II
INT_CHAR_SET_LATIN_AMERICA	Latin America
INT_CHAR_SET_ARABIA	Arabia

(10) **CharacterType**

Enumerator used for specifying the character font of **setCharacterFormatting**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
CHAR_TYPE_FONT_CURRENT[*1]	Current Setting (without sending the setting printer command)
CHAR_TYPE_FONT_A	Font A (24 × 12), Kanji font A (24 × 24)
CHAR_TYPE_FONT_B[*1]	Font B (16 × 8), Kanji font B (16 × 16)

*1: When **CODEPAGE_864** is selected in **selectCharacterSet**, the text is printed in Font A (24 × 12) regardless of specifying the character font in **setCharacterFormatting**.

(11) **CharacterScale**

Enumerator used for specifying the character scale of **setCharacterFormatting**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
CHAR_SCALE_CURRENT	Current Setting (without sending the setting printer command)
CHAR_SCALE_X1	× 1 (Standard)
CHAR_SCALE_X2	× 2 (double)
CHAR_SCALE_X3	× 3 (triple)
CHAR_SCALE_X4	× 4 (quadruple)
CHAR_SCALE_X5	× 5 (quintuple)
CHAR_SCALE_X6	× 6 (sextuple)
CHAR_SCALE_X7	× 7 (septuple)
CHAR_SCALE_X8	× 8 (octuple)

(12) **Underline**

Enumerator used for specifying the underline of **setCharacterFormatting**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
CHAR_UNDERLINE_CURRENT	Current Setting (without sending the setting printer command)
CHAR_UNDERLINE_NONE	No underline
CHAR_UNDERLINE_1DOT	Specify 1 dot width underline
CHAR_UNDERLINE_2DOT	Specify 2 dots width underline

(13) **Bold**

Enumerator used for specifying the bold print of **setCharacterFormatting**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
CHAR_BOLD_CURRENT	Current Setting (without sending the setting printer command)
CHAR_BOLD_OFF	No bold print
CHAR_BOLD_ON	Specify bold print

(14) **Reverse**

Enumerator used for specifying the reverse print of **setCharacterFormatting**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
CHAR_REVERSE_CURRENT	Current Setting (without sending the setting printer command)
CHAR_REVERSE_OFF	No reverse print
CHAR_REVERSE_ON	Specify reverse print

(15) **Rotate**

Enumerator used for specifying the rotation print of **setCharacterFormatting**, for specifying the print direction of **setStandardModeBarcodeDirection**, and for specifying the image rotation direction of **printPDF**. The default value is shown by the shaded enumerated value in the following table.
When the character rotation print is specified **setCharacterFormatting**, the print position is changed. The print position and direction of text is shown in the Figure 4-3, Figure 4-4, Figure 4-5 and Figure 4-6.

Constant Name	Description
ROTATE_CURRENT[*1]	Current Setting (without sending the setting printer command)
ROTATE_NONE	No rotation
ROTATE_90_TO_RIGHT[*1]	Rotate 90 degrees to right
ROTATE_180	Rotate 180 degrees
ROTATE_90_TO_LEFT[*1]	Rotate 90 degrees to left

*1: Not supported in **printPDF**.

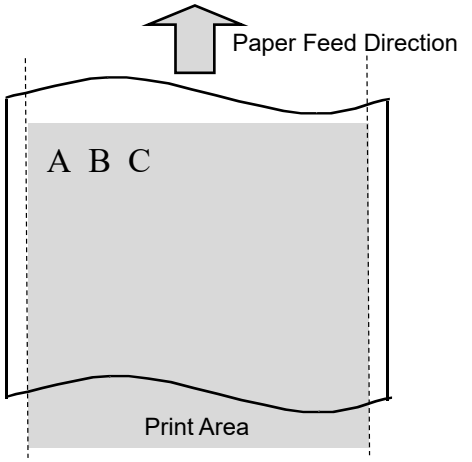


Figure 4-3 No rotation

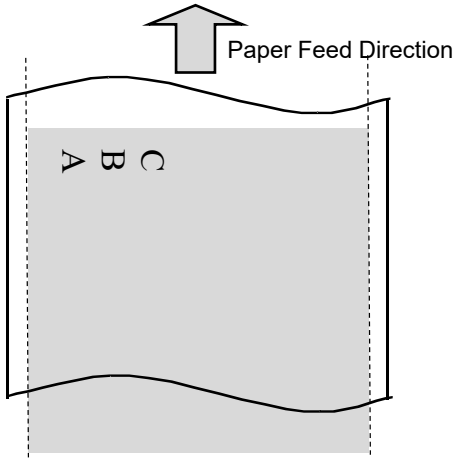


Figure 4-4 Rotate 90 degrees to right

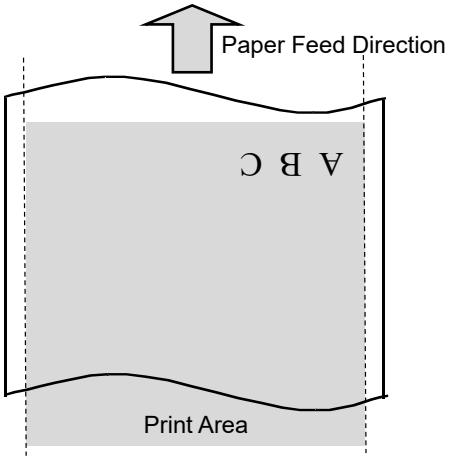


Figure 4-5 Rotate 180 degrees

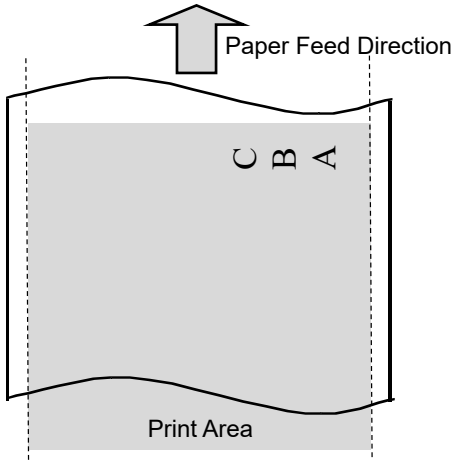


Figure 4-6 Rotate 90 degrees to left

(16) **TypeBarcode**

Enumerator used for specifying the barcode type of **printBarcode**.

Constant Name	Description	
TYPE_BARCODE_UPC_A	UPC-A	Multilevel barcode
TYPE_BARCODE_UPC_E	UPC-E	Multilevel barcode
TYPE_BARCODE_JAN13	JAN13 (EAN13)	Multilevel barcode
TYPE_BARCODE_JAN8	JAN8 (EAN8)	Multilevel barcode
TYPE_BARCODE_CODE39	CODE39	Binary level barcode
TYPE_BARCODE_ITF	ITF	Binary level barcode
TYPE_BARCODE_CODABAR	CODABAR	Binary level barcode
TYPE_BARCODE_CODE128	CODE128	Multilevel barcode
TYPE_BARCODE_CODE93	CODE93	Multilevel barcode
TYPE_BARCODE_JAN13_ADDON2	JAN13 (EAN13) add-on 2	Multilevel barcode
TYPE_BARCODE_JAN13_ADDON5	JAN13 (EAN13) add-on 5	Multilevel barcode
TYPE_BARCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	Multilevel barcode
TYPE_BARCODE_GS1_TRUNCATED	GS1 Databar Truncated	Multilevel barcode
TYPE_BARCODE_GS1_LIMITED	GS1 Databar Limited	Multilevel barcode
TYPE_BARCODE_GS1_EXPANDED	GS1 Databar Expanded	Multilevel barcode

(17) **ModuleWidthBarcode**

Enumerator used for specifying the barcode module width or narrow element of **printBarcode.**

Constant Name	Description
MODULE_WIDTH_BARCODE_2	2 dots
MODULE_WIDTH_BARCODE_3	3 dots
MODULE_WIDTH_BARCODE_4	4 dots
MODULE_WIDTH_BARCODE_5	5 dots
MODULE_WIDTH_BARCODE_6	6 dots

(18) **HriPositionBarcode**

Enumerator used for specifying the barcode HRI characters of **printBarcode**.

Constant Name	Description
HRI_BARCODE_NONE	No HRI character
HRI_BARCODE_TOP_FONT_A	Above barcode (font A)
HRI_BARCODE_BOTTOM_FONT_A	Below barcode (font A)
HRI_BARCODE_FONT_A	Above and below barcode (font A)
HRI_BARCODE_TOP_FONT_B	Above barcode (font B)
HRI_BARCODE_BOTTOM_FONT_B	Below barcode (font B)
HRI_BARCODE_FONT_B	Above and below barcode (font B)

(19) **NwRatioBarcode**

Enumerator used for specifying the barcode N:W ratio of **printBarcode**.

Constant Name	Description
NWRATIO_BARCODE_1TO2	1:2
NWRATIO_BARCODE_1TO2_5	1:2.5
NWRATIO_BARCODE_1TO3	1:3

(20) **Type2DCode**

Enumerator used for specifying the 2-dimensional barcode type of **print2DCode**.

Constant Name	Description
TYPE_2DCODE_QR_CODE	QR Code
TYPE_2DCODE_PDF417	PDF417
TYPE_2DCODE_DATA_MATRIX	Data Matrix
TYPE_2DCODE_MAXI_CODE	Maxi Code
TYPE_2DCODE_GS1_STACKED	GS1 Databar Stacked
TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Stacked Omni-directional
TYPE_2DCODE_GS1_EXPANDED_STACKED	GS1 Databar Expanded Stacked

(21) **Mode2Dcode**

Enumerator used for specifying the 2-dimensional barcode mode of **print2DCode**.

Constant Name	Description	
MODE_2DCODE_QR_CODE_MODEL1	QR Code	Model1
MODE_2DCODE_QR_CODE_MODEL2	QR Code	Model2
MODE_2DCODE_PDF417_STANDARD	PDF417	Normal Mode
MODE_2DCODE_PDF417_COMPACT	PDF417	Simple Mode
MODE_2DCODE_DATA_MATRIX_AUTO	Data Matrix	Module numbers: Automatic
MODE_2DCODE_DATA_MATRIX_10_10	Data Matrix	Module numbers: 10 × 10
MODE_2DCODE_DATA_MATRIX_12_12	Data Matrix	Module numbers: 12 × 12
MODE_2DCODE_DATA_MATRIX_14_14	Data Matrix	Module numbers: 14 × 14
MODE_2DCODE_DATA_MATRIX_16_16	Data Matrix	Module numbers: 16 × 16
MODE_2DCODE_DATA_MATRIX_18_18	Data Matrix	Module numbers: 18 × 18
MODE_2DCODE_DATA_MATRIX_20_20	Data Matrix	Module numbers: 20 × 20
MODE_2DCODE_DATA_MATRIX_22_22	Data Matrix	Module numbers: 22 × 22
MODE_2DCODE_DATA_MATRIX_24_24	Data Matrix	Module numbers: 24 × 24
MODE_2DCODE_DATA_MATRIX_26_26	Data Matrix	Module numbers: 26 × 26
MODE_2DCODE_DATA_MATRIX_32_32	Data Matrix	Module numbers: 32 × 32
MODE_2DCODE_DATA_MATRIX_36_36	Data Matrix	Module numbers: 36 × 36
MODE_2DCODE_DATA_MATRIX_40_40	Data Matrix	Module numbers: 40 × 40
MODE_2DCODE_DATA_MATRIX_44_44	Data Matrix	Module numbers: 44 × 44

Constant Name	Description	
MODE_2DCODE_DATA_MATRIX_48_48	Data Matrix	Module numbers: 48 × 48
MODE_2DCODE_DATA_MATRIX_52_52	Data Matrix	Module numbers: 52 × 52
MODE_2DCODE_DATA_MATRIX_64_64	Data Matrix	Module numbers: 64 × 64
MODE_2DCODE_DATA_MATRIX_72_72	Data Matrix	Module numbers: 72 × 72
MODE_2DCODE_DATA_MATRIX_80_80	Data Matrix	Module numbers: 80 × 80
MODE_2DCODE_DATA_MATRIX_88_88	Data Matrix	Module numbers: 88 × 88
MODE_2DCODE_DATA_MATRIX_96_96	Data Matrix	Module numbers: 96 × 96
MODE_2DCODE_DATA_MATRIX_104_104	Data Matrix	Module numbers: 104 × 104
MODE_2DCODE_DATA_MATRIX_120_120	Data Matrix	Module numbers: 120 × 120
MODE_2DCODE_DATA_MATRIX_132_132	Data Matrix	Module numbers: 132 × 132
MODE_2DCODE_DATA_MATRIX_144_144	Data Matrix	Module numbers: 144 × 144
MODE_2DCODE_DATA_MATRIX_8_18	Data Matrix	Module numbers: 8 × 18
MODE_2DCODE_DATA_MATRIX_8_32	Data Matrix	Module numbers: 8 × 32
MODE_2DCODE_DATA_MATRIX_12_26	Data Matrix	Module numbers: 12 × 26
MODE_2DCODE_DATA_MATRIX_12_36	Data Matrix	Module numbers: 12 × 36
MODE_2DCODE_DATA_MATRIX_16_36	Data Matrix	Module numbers: 16 × 36
MODE_2DCODE_DATA_MATRIX_16_48	Data Matrix	Module numbers: 16 × 48
MODE_2DCODE_MAXI_CODE_2	Maxi Code	Mode2
MODE_2DCODE_MAXI_CODE_3	Maxi Code	Mode3
MODE_2DCODE_MAXI_CODE_4	Maxi Code	Mode4
MODE_2DCODE_MAXI_CODE_5	Maxi Code	Mode5
MODE_2DCODE_NONE	GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	No settings

(22) **ModuleSize2DCode**

Enumerator used for specifying the 2-dimensional barcode module size of **print2DCode.**

Constant Name	Description	
MODULE_SIZE_2DCODE_DEFAULT	QR Code	Default (6 dots)
	PDF417	Default (4 dots)
	Data Matrix	Default (6 dots)
	Maxi Code	No settings
	GS1 Databar Stacked	Default (6 dots)
	GS1 Databar Stacked Omni-directional	Default (6 dots)
	GS1 Databar Expanded Stacked	Default (6 dots)
MODULE_SIZE_2DCODE_2	QR Code	2 dots
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_3	QR Code	3 dots
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_4	QR Code	4 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_5	QR Code	5 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_6	QR Code	6 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_7	QR Code	7 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

Constant Name	Description	
MODULE_SIZE_2DCODE_8	QR Code	8 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_9	QR Code	9 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_10	QR Code	10 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_11	QR Code	11 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_12	QR Code	12 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_13	QR Code	13 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_14	QR Code	14 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_15	QR Code	15 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

Constant Name	Description	
MODULE_SIZE_2DCODE_16	QR Code	16 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

(23) **ErrorCorrect2DCode**

Enumerator used for specifying the 2-dimensional barcode error correction level of **print2DCode**.

Constant Name	Description	
ERR_CORRECTION_2DCODE_QR_CODE_L	QR Code	L
ERR_CORRECTION_2DCODE_QR_CODE_M	QR Code	M
ERR_CORRECTION_2DCODE_QR_CODE_Q	QR Code	Q
ERR_CORRECTION_2DCODE_QR_CODE_H	QR Code	H
ERR_CORRECTION_2DCODE_PDF417_0	PDF417	0
ERR_CORRECTION_2DCODE_PDF417_1	PDF417	1
ERR_CORRECTION_2DCODE_PDF417_2	PDF417	2
ERR_CORRECTION_2DCODE_PDF417_3	PDF417	3
ERR_CORRECTION_2DCODE_PDF417_4	PDF417	4
ERR_CORRECTION_2DCODE_PDF417_5	PDF417	5
ERR_CORRECTION_2DCODE_PDF417_6	PDF417	6
ERR_CORRECTION_2DCODE_PDF417_7	PDF417	7
ERR_CORRECTION_2DCODE_PDF417_8	PDF417	8
ERR_CORRECTION_2DCODE_NONE	Data Matrix Maxi Code GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	No settings

(24) **LogFileSize**

Enumerator used for specifying the maximum size of the log file of **setLog**.

Constant Name	Description
LOG_FILE_SIZE_1MB	1 MB (1048576 bytes)
LOG_FILE_SIZE_5MB	5 MB (5242880 bytes)
LOG_FILE_SIZE_10MB	10 MB (10485760 bytes)
LOG_FILE_SIZE_50MB	50 MB (52428800 bytes)

(25) **ErrorCode**

Enumerator used for retrieving the error code by **getErrorCode**.

Constant Name	Description
ERR_PARAM	Parameter is incorrect.
ERR_OPENED	Specified printer has already been opened.
ERR_NOT_OPENED	Specified printer is not opened.
ERR_TIMEOUT	Timeout or busy state occurs.
ERR_OFFLINE	Printer is disconnected or offline.
ERR_CLOSE_FAIL	Failed to disconnect printer.
ERR_NOT_MONITORING	Monitoring of connecting status is not performed.
ERR_INIT_FAILED	Failed to initialize.
ERR_DATA_SIZE_ZERO	0 byte size data is specified.
ERR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.
ERR_INVALID_DATA	Invalid data is specified.
ERR_INVALID_STATE	Specified to access PrinterManager object for callback.
ERR_ACCESS	Printer cannot be accessed.
ERR_CANCELED	Function is canceled.
ERR_WRITE_FAULT	Data cannot be sent to printer.
ERR_WORKAREA_NO_MEMORY	Specified memory size is insufficient.
ERR_FILE_INVALID	Specified file is invalid.
ERR_ENCODE_FAILED	Error has occurred in encoding text data.
ERR_NOT_FOUND	Specified file cannot be found.
ERR_TRANSACTION_STOPPED	Batch process is not started.
ERR_PRINTER_STATUS_ERROR	Printer status is abnormal.

4.4.4 Exception

(1) **PrinterException**

- **Summary**

This class provides an exception occurs.
This class provides the following function.

Public Methods

Method	Function Summary
getErrorCode	Retrieve error codes

- **Public Methods**

getErrorCode — Retrieve error codes

Retrieves error code for thrown exception.

Syntax public Errorcode **getErrorCode**()

Return value Error code

Description See "**4.4.3**(25) **ErrorCode**" for details.

Chapter 5
Sample Program

This chapter describes the sample programs provided by the SDK.

5.1 Sample Program Overview

The SDK includes a sample program of Android Studio project format.

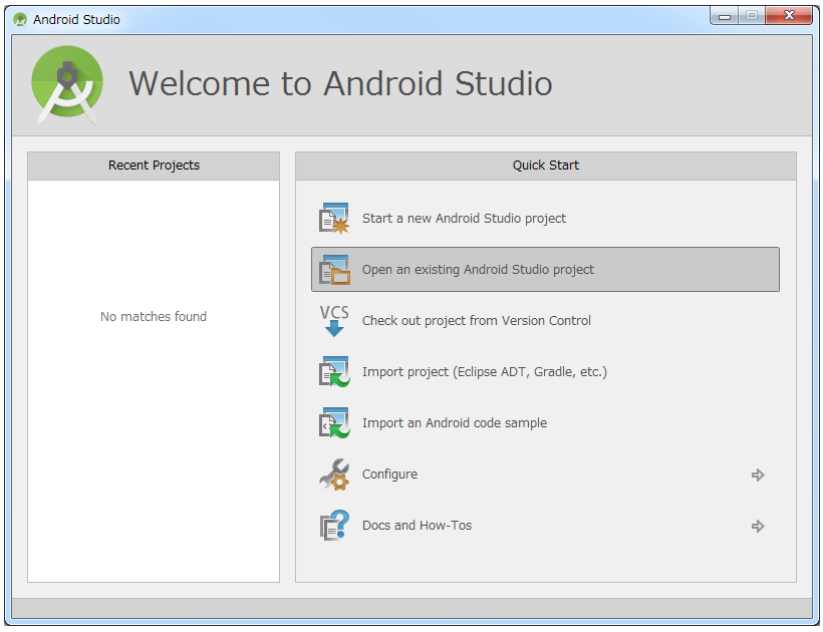
5.2 How to Use Sample Programs

Ensure that the environment for developing Android application is prepared.
See "Chapter 3 How to Use the Library" for details about required development environment.

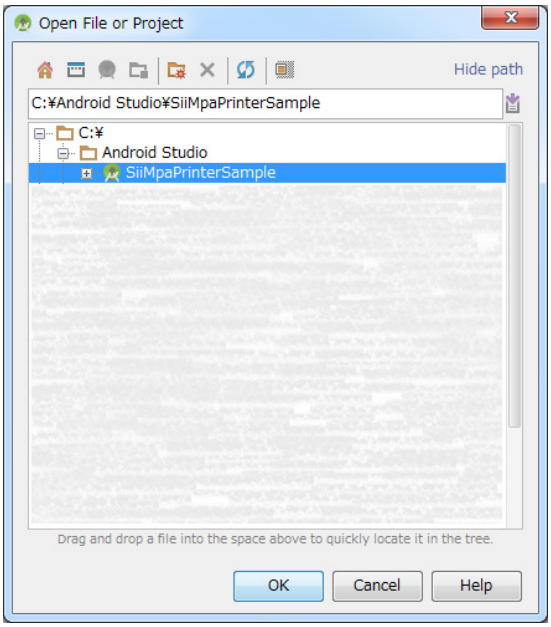
The installation procedure is as follows.

(1) Deploy the SiiMpaPrinterSample folder at any location.

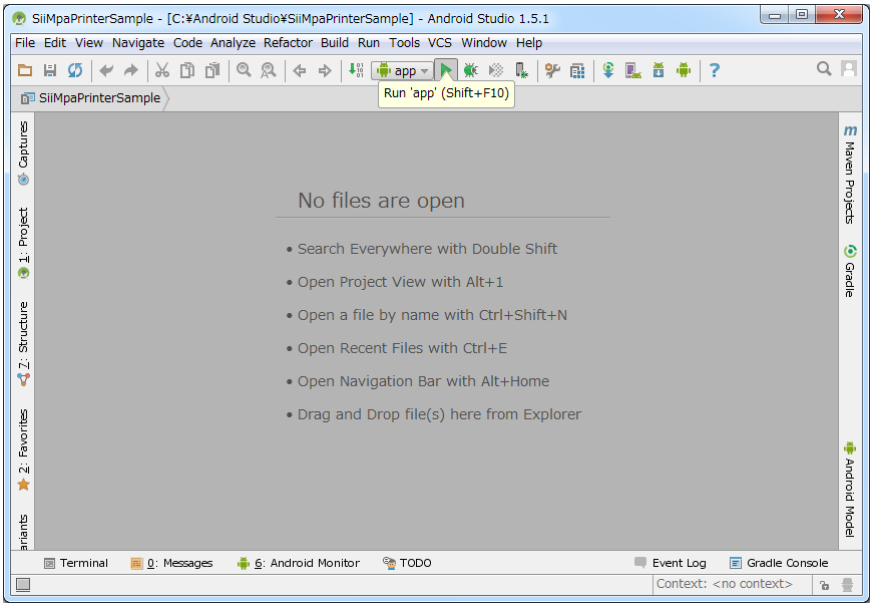
(2) Start Android Studio, and click "Open an existing Android Studio Project".



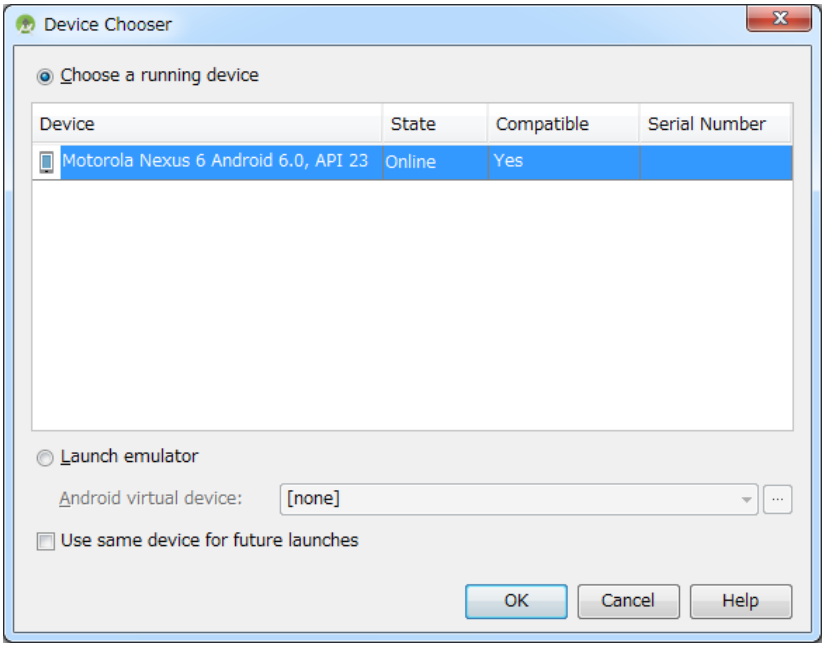
(3) Select the folder that deployed in procedure (1), and then click [OK].



(4) Click [Run 'app'].

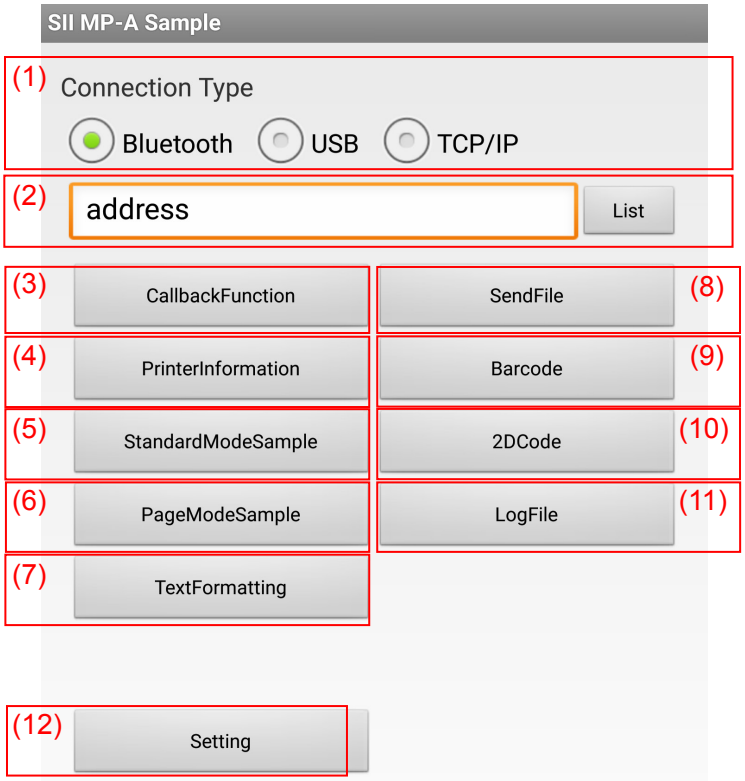


(5) Select the device, and click [OK].



5.3 Sample Program Function

This section describes the functions of the sample program by using the SiiMpaPrinterSample project screen as an example.



No.	Description	Remarks
(1)	Connection type	Selects connection type to a printer.
(2)	Address	Specifies printer address. When use Bluetooth connection, enter Bluetooth address of a printer. By tapping [List] button, the list of devices searched by **startDiscoveryPrinter** (Bluetooth) is displayed. By selecting a printer to connect from the list, Bluetooth address can be entered. Example: "00:11:22:AA:BB:CC" When use [TCP/IP] connection, enter IP address of a printer. By tapping [List] button, the list of devices searched by **startDiscoveryPrinter** (TCP/IP) is displayed. By selecting a printer to connect from the list, IP address can be entered. Example: "192.168.0.190"
(3)	CallbackFunction	Registers the callback of the printer status change in **startCallbackFunction**, and displays the change of status information in a dialog.

No.	Description	Remarks
(4)	PrinterInformation	Retrieves the printer information from the printer by using **getPrinterInformation**, **getPrinterInformationNumber** and **getPrinterInformationString**.
(5)	StandardModeSample	In the standard mode, send the sample receipt printing command created by various printing method to the printer.
(6)	PageModeSample	In the page mode, send the sample receipt printing command created by various printing method to the printer.
(7)	TextFormatting	Send the printing command formatted by various characters formatting to the printer.
(8)	SendFile	Send the specified file by using **sendDataFile** to the printer.
(9)	Barcode	Send the barcode printing command by using **printBarcode** to the printer.
(10)	2DCode	Send the 2-dimensional barcode printing command by using **print2DCode** to the printer.
(11)	LogFile	Sets the log file size to be stored by using **setLog**.
(12)	Setting	Sets the following functions: ・Socket Keeping Time Sets the socket keeping time of TCP/IP connection in msec (milliseconds). This setting value is used in *socketKeepingTime* which is the parameter of **open**. ・Write Timeout Sets the time until the send timeout occurrence in msec (milliseconds). ・Response Timeout Sets the time until the receive timeout occurrence in msec (milliseconds). ・Codepage Sets the codepage. This is the encoding when printing texts (when text files are specified by **TextFormatting** or **sendFile**)and the character setting value for printer side. ・International Character Sets the international character set. This is the international character setting value for printer side when printing texts (when text files are specified by **TextFormatting** or **sendFile**). ・Bluetooth Secure Sets whether secure mode or insecure mode is used at Bluetooth connection.

5.4 Precaution

No guarantee of proper operation and support are provided for sample programs.

Sample programs are subject to change without notice.

Chapter 6
Disclaimer

We closely monitor the development of this software in order to avoid problems. However, we are not responsible for any damages arising out of the use of this software.

Appendix A
Character Sets (Character Code Table)

A.1 Character Code Table

(When **COUNTRY_USA** is set in the setting of international character set)

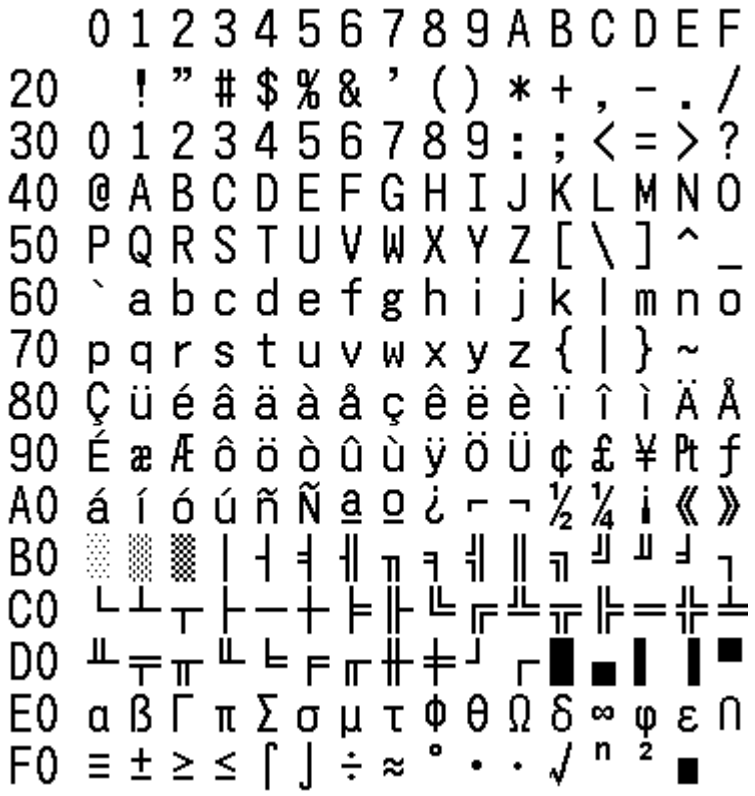


Figure A-1 CODEPAGE_437 (USA, Standard Europe)

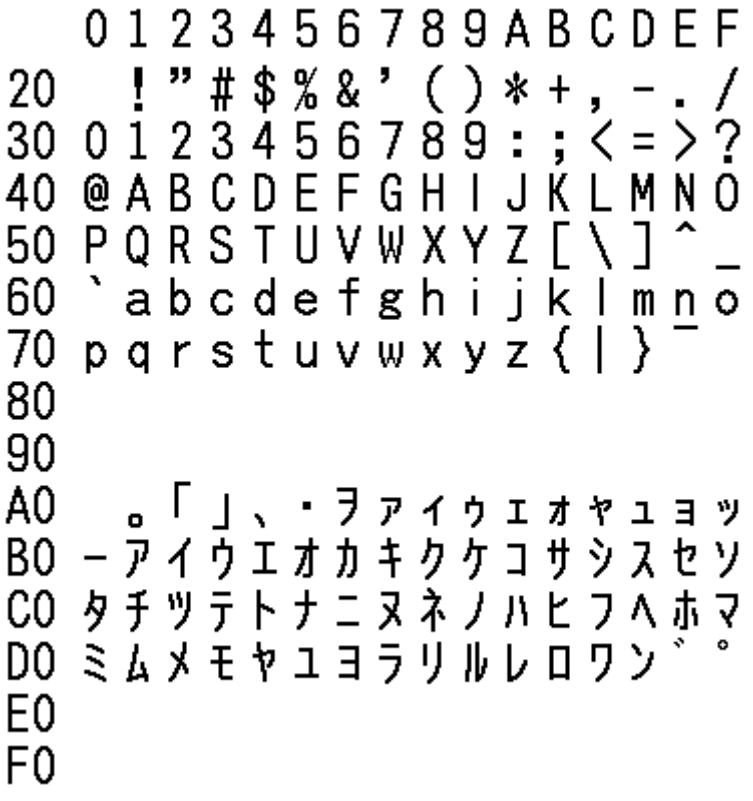


Figure A-2 CODEPAGE_KATAKANA

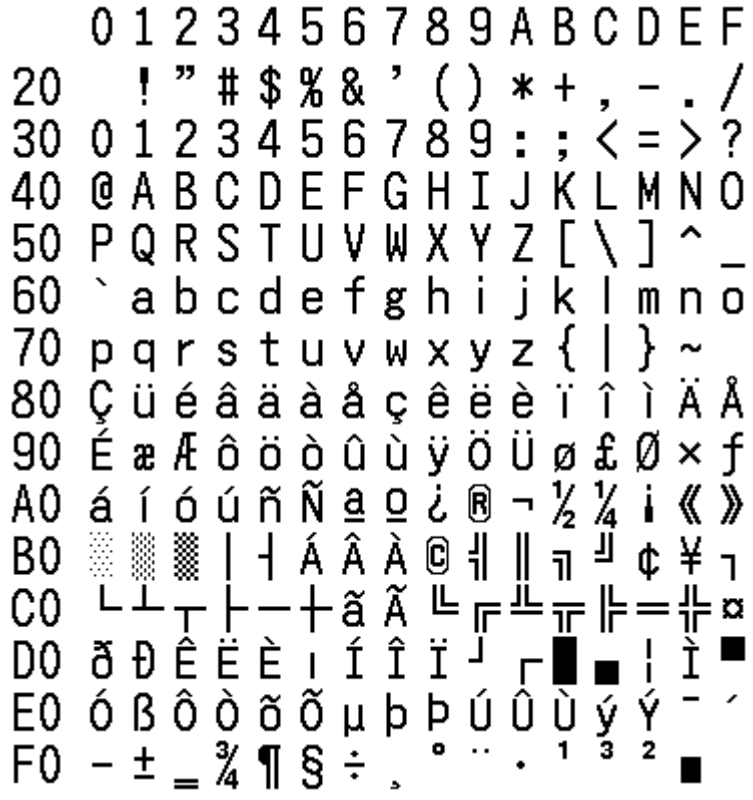


Figure A-3 CODEPAGE_850 (Multilingual)

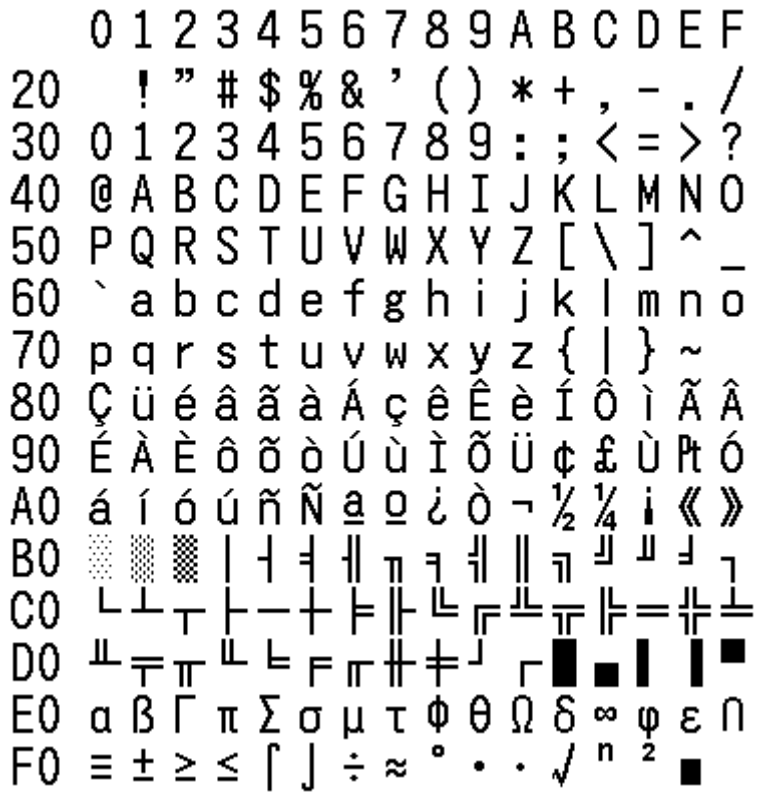


Figure A-4 CODEPAGE_860 (Portuguese)

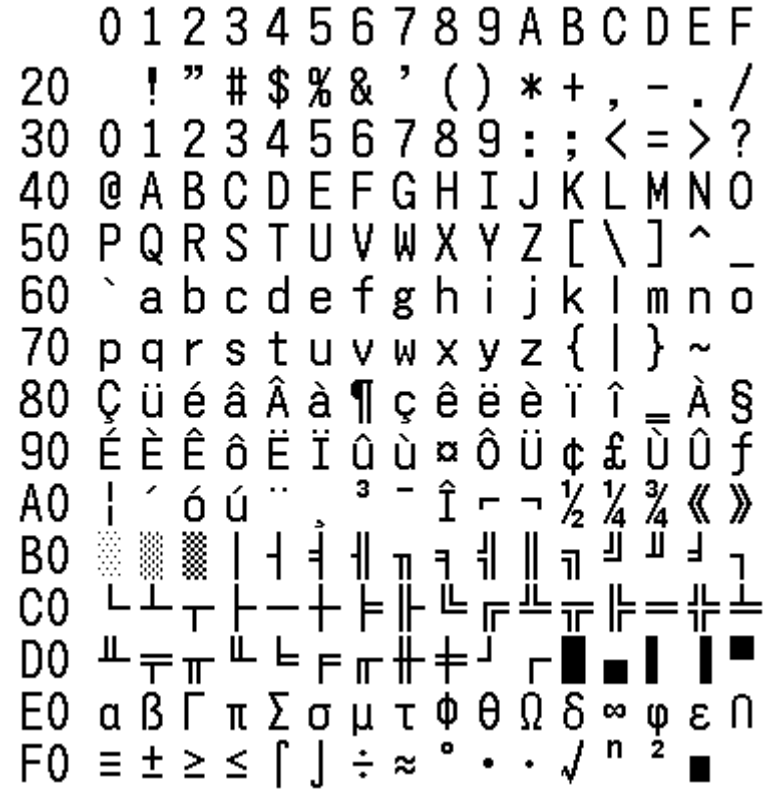


Figure A-5 CODEPAGE_863 (Canadian-French)

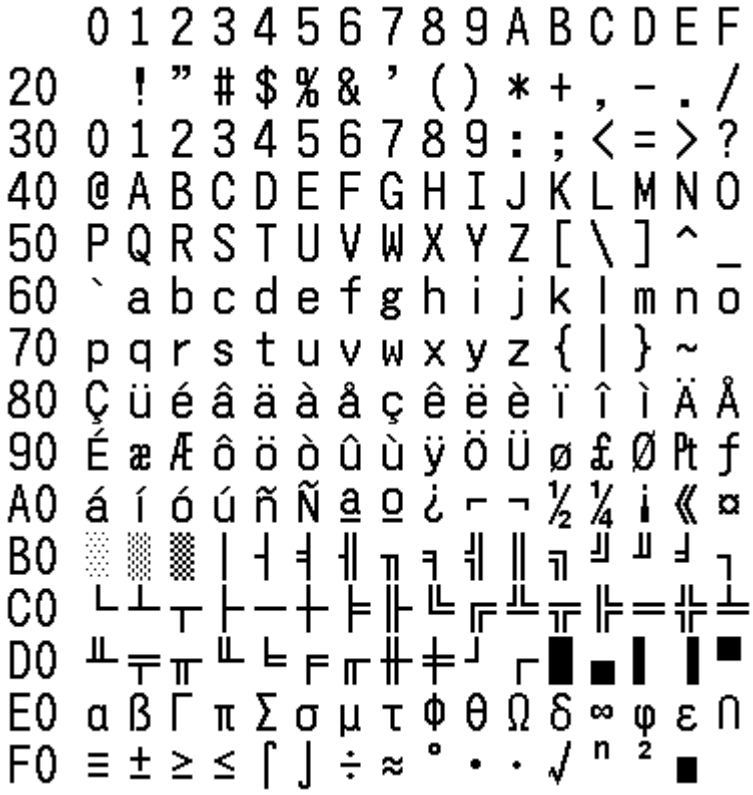


Figure A-6 CODEPAGE_865 (Nordic)

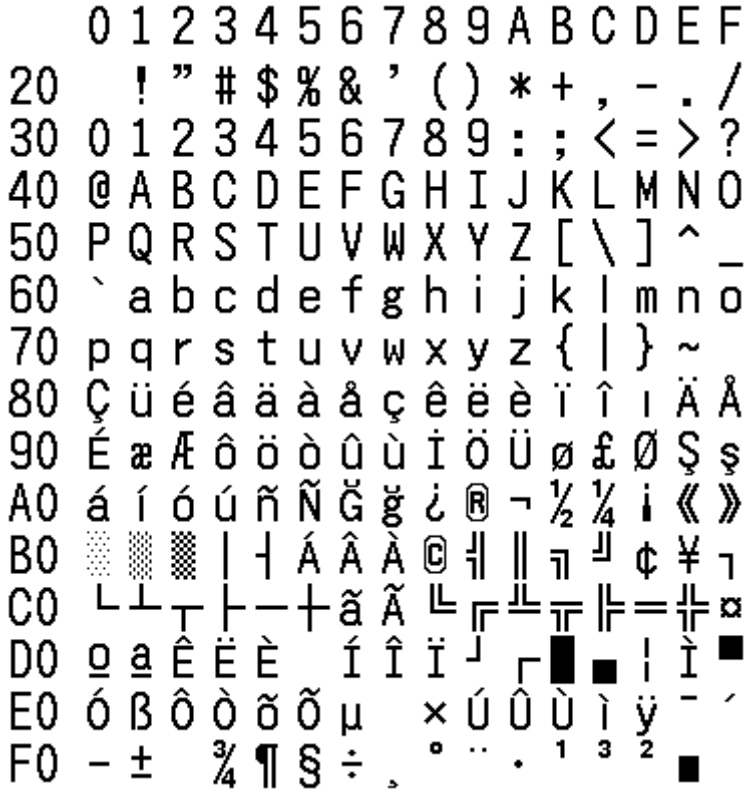


Figure A-7 CODEPAGE_857 (Turkish)

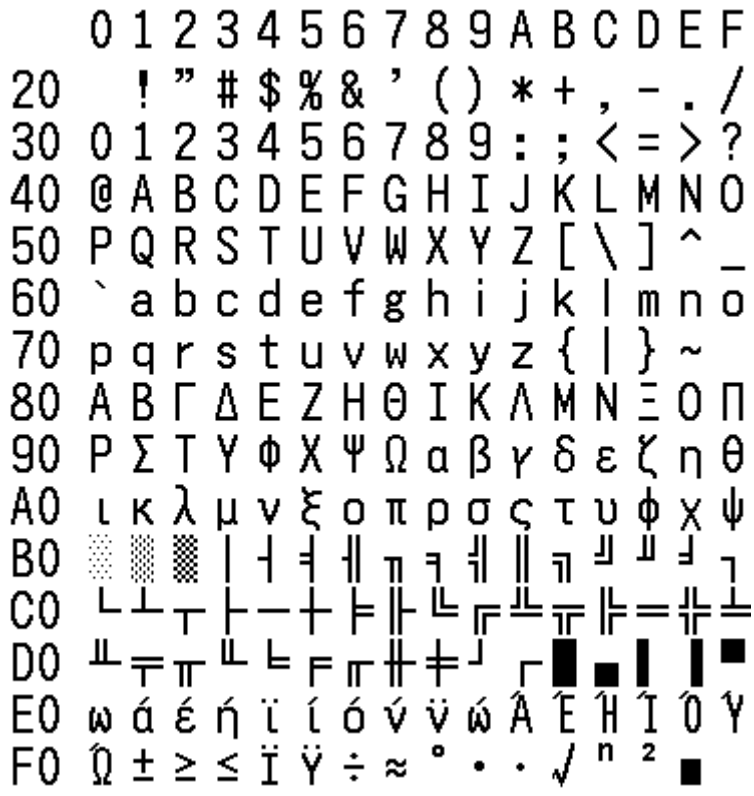


Figure A-8 CODEPAGE_737 (Greek)



Figure A-9 CODEPAGE_1252 (Latin)

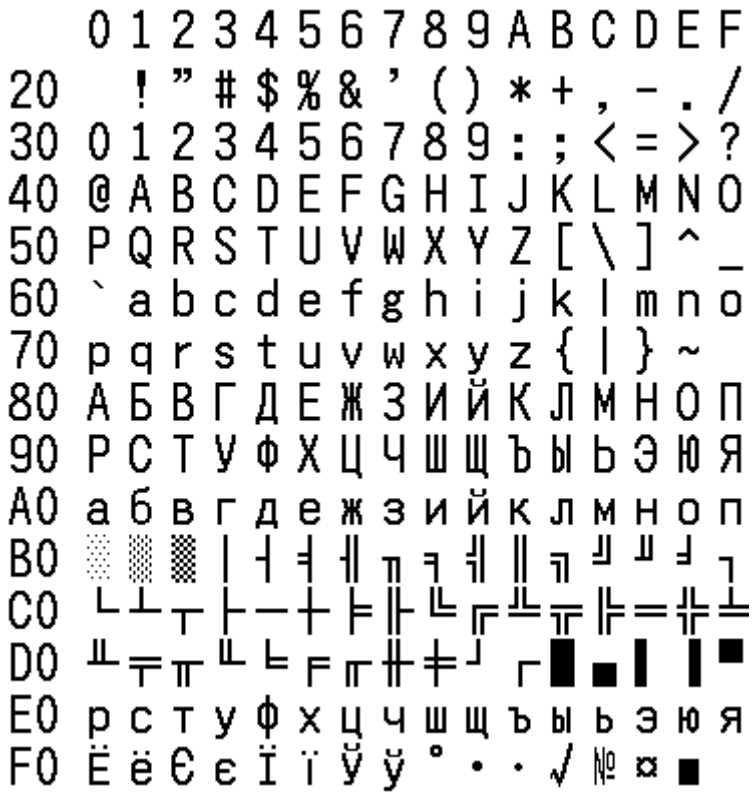


Figure A-10 CODEPAGE_866 (Russian)

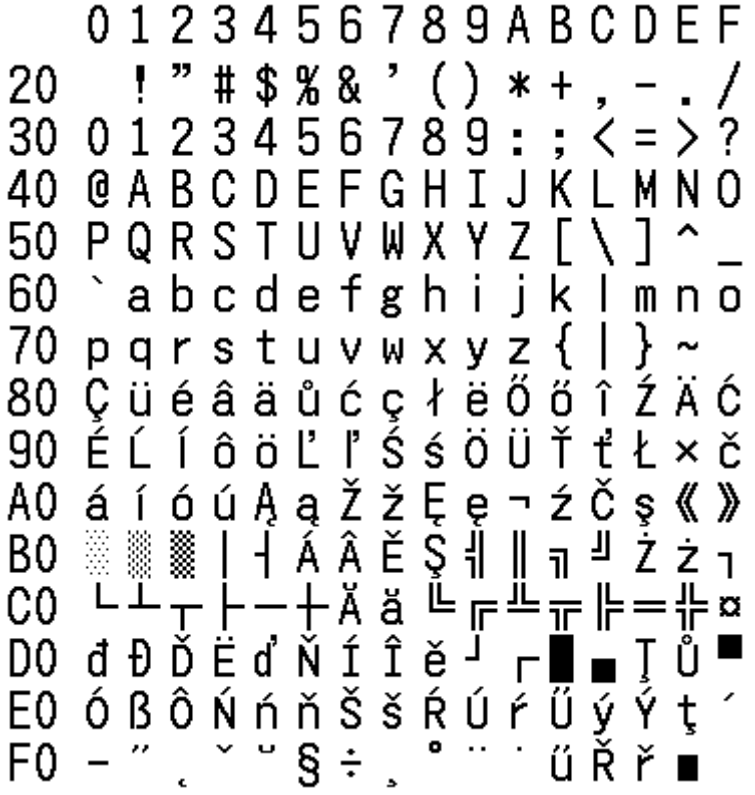


Figure A-11 CODEPAGE_852 (Eastern Europe)

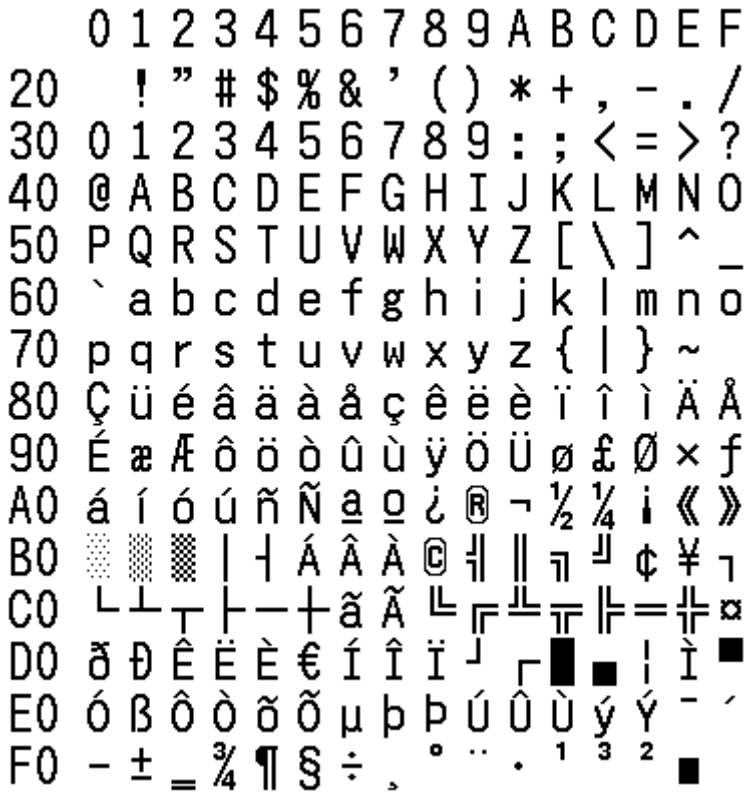


Figure A-12 CODEPAGE_858 (Euro)



Figure A-13 CODEPAGE_855 (Cyrillic)



Figure A-14 CODEPAGE_864 (Arabic)



Figure A-15 CODEPAGE_1250 (Central European)



Figure A-16 CODEPAGE_1251 (Cyrillic)

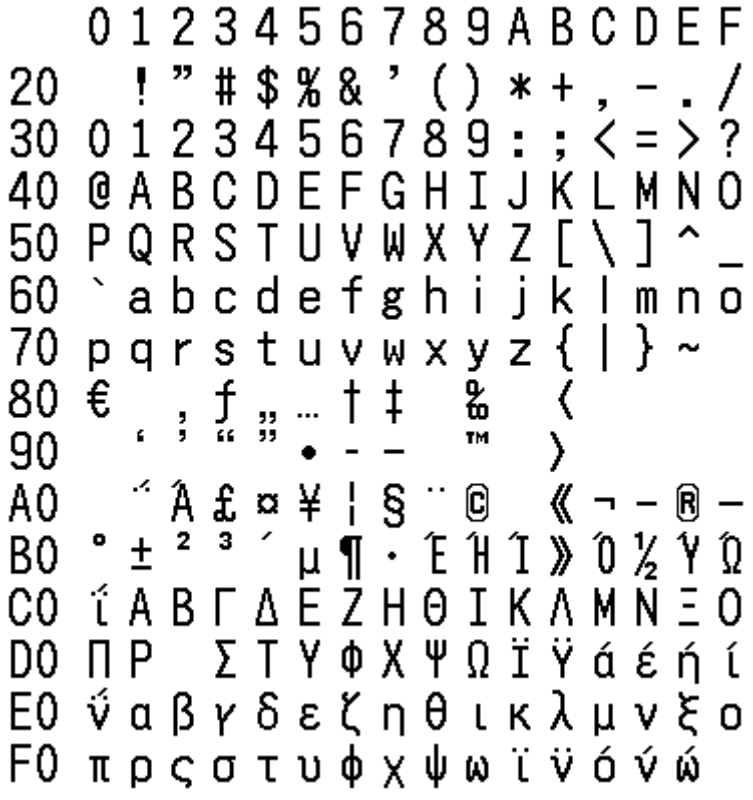


Figure A-17 CODEPAGE_1253 (Greek)



Figure A-18 CODEPAGE_1254 (Turkish)

A.2 International Character Set

The following codes differ depending on the specified international character set.

	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
COUNTRY_GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_DENMARK_1	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
COUNTRY_JAPAN	#	$	@	[	¥	]	^	`	{	|	}	~
COUNTRY_NORWAY	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	$	á	¡	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	$	@	[	\	]	^	`	{	|	}	~

Figure A-19 International Character Set