SAM4S Print SDK for Android

User's Manual

User Guide

All specifications are subject to change without notice.

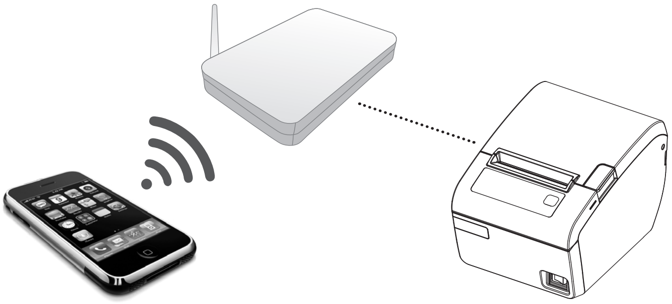
SAM4S

Contents

CHAPTER 1.
OVERVIEW

CHAPTER 1 OVERVIEW

The SAM4S Print SDK for Android is an SDK aimed at development engineers who are developing Android applications for printing on an SAM4S printer.



1.1. Features

Allows printing to SAM4S printers (ELLIX20II or above) from Android applications.

1.2. Features

Builder API

- Print Settings (alignment/line feed space/ text rotation/page mode)
- Character data settings (language/font (device font)/double-sizing/scale)
- Character style settings (inversion of black and white/underline/bold)
- Paper feed setting (in dots/in lines)
- Image printing
- Barcode printing (For barcodes that can be printed by each model)
- 2D-code printing (For 2D-Codes that can be printed by each model)
- Drawer kick function
- ESC/POS command transmission
- Acquisition of response from printer (print status/print name)

Finder API

- Search for printers

1.3. Operating Environment

Android Version

- Android version 2.3.3 to 2.3.7
- Android Version 3.1 to 3.2.2
- Android Version 4.0 to 4.3

Android Device

- Device that supports ARMv5TE

Printer

- ELLIX20II or Above
  (includes ELLIX30, ELLIX40, ELLIX35, ELLIX45, GIANT-100)

Interface

- Wired LAN
- Wireless LAN
- Bluetooth
- USB to Seiral (Only for SAM4S SAP Series)

Development Environment

- Android SDK r15 or later
- Java Development Kit 6 or later

1.4. Contents in the Package

File	Description
Sam4sAndroidSample.zip	A zipped sample program file.
Sam4sAndroidSDK_Manual.pdf	This manual.
Sam4sAndriodSDK.jar	Compiled Java class file, archived into a jar format file to allow APIs to be used from Java program.

1.5. Restrictions

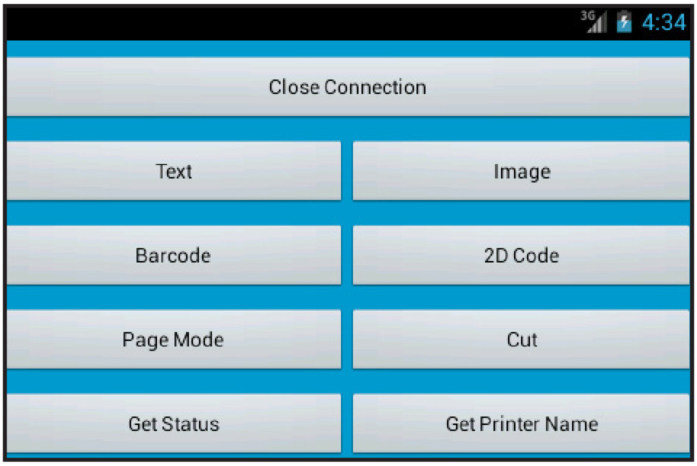
- A communication API and command transmission / reception API in the SAM4S Android SDK APIs cannot be used for the same device at the same time.
- More than one port cannot be opened for the same device at the same time.

CHAPTER 2.
Sample Program

CHAPTER 2

Sample Program

2.1. Functionality



The Sample Program has the following functionalities.

- Search for printers
- Opening of port
- Closing of port
- Text printing
- Graphic printing (image file printing)
- Barcode printing
- 2D-code printing
- Printing in page mode
- Paper cutting
- Acquisition of printer status
- Acquisition of printer model name information

2.2. Usage Environment

Development Environment

- Android SDK r17
- Java Development Kit 6
- Eclipse
- ADT Plugin for Eclipse

Printer

- SAM4S printer supported in "SAM4S Android"

Target Device

- Device connected to a computer via USB

Environmental Construction

» Follow the procedures below to use the sample program.

1. Extract the zipped sample program file to a directory of your choosing.
2. In Eclipse, go to [File]-[Import], select [General]-[Existing Project into Worksapce], and then click [Next].
3. The Import Projects window will be displayed. Make the settings shown below and click [Finish].

Item	Setting
Select root directory	Specify the directory where you extracted the zipped sample program file.
Copy projects into workspace	Check this option.

4. In Package Explorer view, right click on the "Sam4sAndroid" project and select [Properties].
5. The Properties for "Sam4sAndroid" window will be displayed. Make the settings shown below and click [OK].

Item	Setting
Android	Select the Android OS version of the target device.
Java Build Path	In Libraries, confirm that the path to "Sam4sAndroidSdk.jar", which is located in the "Sam4sAndroid" project that was copied into the project workspace, is set.

6. In Package Explorer view, right click on the "Sam4sAndroid" project and select [Run As, Android Application].
7. The sample program will be installed to the target Android device, and then the program will start up.

2.3. How to use sample program

Search for printers and printing

1. Start the sample program. For details, refer to Environmental Construction (p.8)
2. Search for printers. The sample program is searching Bluetooth device automatically when the program is started. When you select [Device Type], the IP addresses / Mac addresses for the detected printers are listed.
3. Select the IP address / Mac address of the printer that you want to use from the list of IP addresses / Mac addressed displayed in procedure 2.
4. Open the printer's port. The sample program will open the selected port in procedure 2.
5. Execute the following processes:

Process	Description
Text printing	Tap [Text] on the main screen. For details, refer to Text printing (p.10).
Graphic printing	Tap [Image] on the main screen. For details, refer to Graphic printing (p.10).
Barcode printing	Tap [Barcode] on the main screen. For details, refer to Barcode printing (p.10).
2D-code printing	Tap [2D Code] on the main screen. For details, refer to 2D-code printing (p.11).
Printing in page mode	Tap [Page Mode] on the main screen. For details, refer to printing in page mode (p.11).
Paper cutting	Tap [Cut] on the main screen. For details, refer to paper cutting (p.11).
Acquisition of printer status	Tap [Get Status] on the main screen.
Acquisition of printer model name information	Tap [Get Printer Name] on the main screen

6. When all processing is finished, tap [Close Connection] on the screen.

***This is not applied for serial connection.**

2.3. How to use sample program

Text Printing

1. Enter a string to print for [Print Characters].
2. Specifies the character properties for the string to print. The following properties can be specified:

Process	Description
Font	Set the character font.
Align	Tap [Image] on the main screen. For details, refer to Graphic printing (p.10).
Line Spacing	Set the line feed space.
Language	Set the language.
Size	Set the character scales (vertical / horizontal).
Style	Set the character style (bold / underlining).
X Position	Set the horizontal start position.
Feed Unit	Set the paper feed amount.

Graphic printing

1. Tap [Select image] to select an image file to print.
2. Tap [Print] to print.

Barcode printing

1. Set the following for barcodes:

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

2. Tap [Print] to print.

2.3. How to use sample program

2D-code printing

1. Select the 2D-code type using [Type].
2. Enter the 2D-code data for [Data].
3. Set the following for each 2D-code:

Process	Description
Error Correction Level	Set the error correction level.
(PDF417, QR Code)	Set the 2D-code module size (width / height).
Module Size(width, Height)	Set the maximum 2D-code size.

Printing in page mode

1. Enter a string to print for [Print Characters].
2. Set the print area using[Print Area]:

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

Paper cutting

1. Set whether to cut after feeding paper using [Type].
2. Tap [Print] and execute cutting operation.

CHAPTER 3.

PROGRAMMING GUIDE

CHAPTER 3

Programming Guide

3.1. How to incorporate the SDK for Android

This explanation uses Eclipse. If you are using another development environment, please make the appropriate changes.

Incorporate the SDK using following procedures.

1. Create a new project in Eclipse.
2. Copy provided JAR file into following path:
3. In Libraries tab of the target project's properties, confirm that the JAR file that you added is registered in [Java Build Path].
   If it has not been added, add the JAR file into build path using [Add Jars…].
4. Select the project in Eclipse's Package Explorer, right click on it, and press [Refresh].
5. Write the package import declaration in the *.java source file(s) of the application that you would like to use this SDK in as flows:
   Import com.Sam4sprint.*;
6. Confirm that the target project's "/libs" folder is I the Source tab of the target project's properties. If not, add "libs" to the build path using [Add Folder…].
7. With the target project selected from Eclipse's package Explorer, select [Preferences] in the [Window] menu.
8. The [Preferences] screen is displayed. From the list on the left, select [Java]-[Compiler]
9. The [Compiler] screen is displayed. Set the [Compiler compliance level:] to "1.6", and click [Apply]. After that, click [OK].
10. Double-click [AndriodManifest.xml] from Eclipse's Package explorer.
11. Select the [Permissions] tab.
12. The [Android Manifest Permissions] screen is displayed. Click the [Add] button.

3.1. How to incorporate the SDK for Android

13. Select [Uses Permission], and click the [OK] button.
14. [Uses Permission] is added to [Permissions]. Select the permissions of functionalities attached to the added [Uses Permission] from the [Name] under [Attributes for Uses Permission].

    There is one setting of permissions for functionalities that can be attached per [Uses Permission] in [Permissions]. For using the Bluetooth functionality and all functionalities, you must repeat settings from procedures 12 to 14.

Functionality	[Name] Setting
Wi-Fi	android.permission.INTERNET
Bluetooth	android.permission.BLUETOOTH android.permission.BLUETOOTH_ADMIN

15. Save “AndroidManifest.xml”

3.2. Print Mode

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

Standard Mode

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

Page Mode

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

3.3. Programming Flow

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

1. Printer Search (p.16)*

- Starting the printer search.
- Getting the printer search result.
- Stopping the printer search.

2. Print Document Creation (p.17)

- To Create a text print document:
- To create a graphic print document:
- To create a page mode print document:

3. Transmission of Print Document (p.21)

This is optional.

3.4. Printer Search

Starting the printer search

» Use the Sam4sFinder class's startSearch (p.60) to start searching for printers. Please refer to the following code.

```
Sam4sFinder ef = new Sam4sFinder() ;

//Start search
Try {
 ef.startSearch(getBaseContext(), ef.DeviceTypeEthernet);
} catch (Exception e) {
 e.printStackTrace();
}
```

Getting the printer search result

» Use the Sam4sFinder class's getResult (p.61) to get the result of the printer search.
Please refer to the following code.

```
String[] deviceList = null;

//Get device list
Try {
 deviceList = ef.getResult();
} catch (Exception e) {
 e.printStackTrace();
}
```

» Since the printer search takes time to complete, you might not receive any search results if you call the Sam4sFinder class's GetResult immediately after you call start.

3.4. Printer Search

Stopping the printer search

» Use the Sam4sFinder class's stopSearch (p.60) to stop searching for printers. Please refer to the following code.

```
//Stop search
Try {
 ef.stopSearch();
} catch (Exception e) {
 e.printStackTrace();
}
```

3.5. Printer Document Creation

Create a print document using the Sam4sBuilder (p.23) class.
Create an Sam4s Builder class using the constructor for it and create a print document using APIs of the Sam4s Builder class.
Use the programming example below for your reference.

```
Try {
 //Initialize a Sam4sBuilder class instance
 Sam4sBuilder  builder = Sam4sBuilder(“ELLIX”, Sam4sBuilder.LANG_KO);
 //Crate a print document
 builder.addTextLang(Sam4sBuilder.LANG_EN);
 builder.addTextFont(Sam4sBuilder.FONT_A);
 builder.addTextSize(3, 3);
 builder.addTextText(“Hello, \t”);
 builder.addTextText(“World! \n”);
 builder.addCut(Sam4sBuilder.CUT_FEED);
} catch (Exception e) {
 e.printStackTrace();
}
```

3.5. Printer Document Creation

To create a text print document:

- » To create a text print document, using APIs for text, store the font settings in command buffers to create a print document. Use the programming example below for your reference
- » For the string “Hello, World!”, to create a print document based on the following settings:
  - Font: FontA
  - Scale: x 4 (horizontal) and x 4 (Vertical)
  - Style Bold

```
Try {
 Sam4sBuilder  builder = Sam4sBuilder(“ELLIX”, Sam4sBuilder.LANG_KO);
 //Crate a print document <Configure the print character settings>
 builder.addTextLang(Sam4sBuilder.LANG_EN);
 builder.addTextFont(Sam4sBuilder.FONT_A);
 builder.addTextSize(4, 4);
 builder.addTextStyle(false, false, true, Sam4sBuilder.COLOR_1);
 builder.addTextText(“Hello, World! \n”);
} catch (Exception e) {
 e.printStackTrace();
}
```

3.5. Printer Document Creation

To create a graphic print document

- To create a graphic print document, for graphics, store the android. graphics.Bitmap class in the command buffers with addImage (p.41) of the Sam4sBuilder class.
  Use the programming example below for your reference.

```
import android.content.res.resources;
import android.graphics.Bitmap;
import android. graphics.BitmapFactory;

Try {
 //Initialize a Sam4sBuilder class instance
 Sam4sBuilder builder = Sam4sBuilder(“ELLIX”, Sam4sBuilder.LANG_KO);
 //Crate a print document
 Bitmap bmp = BitmapFactory.decodeResource(getResources(),
                                    R.drawable.background);
 builder.addImage(bmp, 0, 0, 8, 48);
 builder.addCut(Sam4sBuilder.CUT_FEED);
} catch (Exception e) {
 e.printStackTrace();
}
```

- For ways of graphic printing, you can also print the graphics registered in the printer’s NV memory. For details, refer to addLogo (p.42).

3.5. Printer Document Creation

To create a page mode print document

- » The page mode starts by storing addPageBegin (p.50) of the Sam4sBuilder class into a command buffer. Store the print area (addPageArea (p.52)) and the print start position (addPagePosition (p.54)) in command buffers. Specify the print start position according to the print data. Then, store APIs in command buffers and create print data. For the page mode end, store addPageEnd (p.51) in a command buffer.
  Use the programming example below for your reference.
- » For the string “Hello, World!”, to create a print document based on the following settings:
  - Page mode print area (in dots):
    Origin of horizontal axis: 100, origin of vertical axis: 50, width: 200, height: 100
  - Page mode print positions (in dots):
    Horizontal print position: 0, vertical print position: 42
  - Font: FontA
  - Scale: x 2 (horizontal) and x 2 (Vertical)
  - Style Bold

```
Try {
 //Initialize a Sam4sBuilder class instance
 Sam4sBuilder  builder = Sam4sBuilder(“ELLIX”, Sam4sBuilder.LANG_KO);

 //Crate a print document <The page mode starts>
 builder.addPageBegin();
 builder.addPageArea(100, 50, 200, 100);
 builder.addPagePosition(0, 42);
 //<Configure the print character settings>
 builder.addTextLang(Sam4sBuilder.LANG_EN);
 builder.addTextFont(Sam4sBuilder.FONT_A);
 builder.addTextSize(4, 4);
 builder.addTextStyle(false, false, true, Sam4sBuilder.COLOR_1);
 //<Specify the print data>
builder.addTextText(“Hello, World! \n”);
//<The page mode ends>
 builder.addPageEnd();
 builder.addCut(Sam4sBuilder.CUT_FEED);
} catch (Exception e) {
 e.printStackTrace();
}
```

3.6. Transmission of Print Document

Send a print document using the Sam4sPrint (p.24) class. Create an Sam4sPrint class using the constructor for it, use sendData to specify the Sam4sBuilder class instance that stores the command buffers for the print document, and send the document.
Use the programming example below for your reference.

```
//Initialize a Sam4sPrint class instance
Sam4sPrint printer = new Sam4sPrint();

Try {
Sam4sBuilder  builder = Sam4sBuilder(“ELLIX”, Sam4sBuilder.LANG_KO);
 //Crate a print document <Configure the print character settings>
 builder.addTextLang(Sam4sBuilder.LANG_EN);
 builder.addTextFont(Sam4sBuilder.FONT_A);
 builder.addTextSize(4, 4);
 builder.addTextStyle(false, false, true, Sam4sBuilder.COLOR_1);
 builder.addTextText(“Hello, World! \n”);
 builder.addCut(Sam4sBuilder.CUT_FEED);

 //Send a print document
 //<Start communication with the printer>
   Printer.openPrinter(Sam4sPrint.DEVTYPE_ETHERNET, "192.168.0.168 ");
 //<Send data>
   Printer.sendData(builder);
 //<End communication with the printer>
Printer.closePrinter();
} catch (Exception e) {
 e.printStackTrace();
}
```

CHAPTER 4.
API Reference

CHAPTER 4

API Reference

4.1. SAM4S Android APIs

Sam4sBuilder class

API		Description	Page
Constructor		Initialize an Sam4sBuilder instance.	25
Clearing Command buffer	clearCommandBuffer	Clears the command buffers added by APIs.	25
Text	addTextAlign	Adds a tag for the text alignment setting.	26
	addTextLineSpace	Adds a tag for the line feed space setting.	27
	addTextRotate	Adds a tag for the text rotation setting.	28
	addText	Adds a tag for printing text.	29
	addTextLang	Adds a tag for the target language setting.	30
	addTextFont	Adds a tag for the text font setting.	31
	addTextDouble	Adds a tag for specifying the double-sized text setting.	32
	addTextSize	Adds a tag for the text scale setting.	33
	addTextStyle	Adds a tag for the text style setting.	34
	addTextPosition	Adds a tag for specifying the print position of text.	35
	addtextDirection	Adds a tag for specifying the print direction of text.	36
	addReverse	Adds a tag for the text reverse setting.	37
	addTextBold	Adds a tag for the text bold setting.	38
	addtextUnderline	Adds a tag for the text underline setting.	39
Paper Feed	addFeedUnit	Adds a tag for paper feeding (in dots).	40
	addFeedLine	Adds a tag for paper feeding (in lines).	40
Graphic	addImage	Adds a tag for a raster image to be printed.	41
	addLogo	Adds a tag for an NV logo to be printed.	41
Barcode	addBarcode	Adds a tag for a barcode to be printed.	43
	addSymbol	Adds a tag for a 2D-Code to be printed.	46
Cut	addCut	Adds a tag for paper cut.	48
Drawer Kick-out	addOpenCashDrawer	Adds a tag for the drawer kick-out.	49

API		Description	Page
Page mode	addPageBegin	Adds a tag for switching to page mode.	50
	addPageEnd	Adds a tag for finishing page mode.	51
	addPageArea	Adds a tag for specifying the print area in page mode.	52
	addPageDirection	Adds a tag for specifying the print direction in page mode.	53
	addPagePosition	Adds a tag for specifying the print position in page mode.	54
	addPageFormFeed	Adds a tag for form feeding in page mode.	55
Send Command	addCommand	Add a tag for inserting commands.	50

Sam4sPrint class

API	Description	Page
Constructor	Initialize an Sam4sPrint class instance.	56
openPrinter	Start communication with the printer.	57
closePrinter	End communication with the printer.	58
sendData	Sends a command to the printer.	58
getPrinterName	Acquires the printer status.	59
getPrinterStatus	Acquires the printer name.	59

Sam4sFinder class

API	Description	Page
Constructor	Initialize an Sam4sPrint class instance.	60
startSearch	Start searching for printers.	60
stopSearch	End communication with the printer.	60
getResult	Getting the printer search result.	61

4.2. Sam4sBuilder class

Constructor

» Constructor for the Sam4sBuilder class. Initializes an Sam4sBuilder class instance.

- Syntax
  public Sam4sBuilder (String printerModel, int lang) throws Exception
- Parameter
  - printer Model : specifies the model name for the target printer.
    This parameter is not mandatory.
  - lang : Specifies the language specifications for the printer.
- Example
  If you are initializing the command buffer for the ELLIX30 model:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
}
catch (Exception e) {
e.printStackTrace();
}
```

clearCommandBuffer

» Clears command buffers used by APIs of the Sam4sBuilder class.

- Syntax
  public void clearCommandBuffer()
- Example
  If you are clearing the command buffer:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.clearCommandBuffer();
}
catch (Exception e) {
e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextAlign

» Adds the text alignment setting to the command buffer.

» This API Setting also applies to barcodes / 2D-Code.

» When the page mode is selected for the print mode, use addPagePosition (p.54) instead of this API to set the alignment.

- Syntax
  public void addTextAlign(int align) throws Exception
- Parameter
  - align : Specifies the text alignment.

Set value	Description
Sam4sBuilder.ALIGN_LEFT(default)	Alignment to the left.
Sam4sBuilder.ALIGN_CENTER	Alignment to the center.
Sam4sBuilder.ALGN_RIGHT	Alignment to the right.

- Example
  To set alignment to the center:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addTextAlign(Sam4sBuilder.ALIGN_RIGHT);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextLineSpace

» Adds the line feed space setting to the command buffer.

- Syntax
  public void addTextLineSpace(int linespc) throws Exception
- Parameter
  - linespc : Specifies the line feed space (in dots).
    Specifies an integer from 0 to 255.
- Example
  To set the line feed space to 50 dots:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addTextLineSpace(50);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextRotate

» Adds the text rotation setting to the command buffer.

» This API Setting also applies to barcodes / 2D-Code.

» When the page mode is selected for the print mode, to set text rotation, use addPageDirection (p.53) instead of this API to set the alignment.

- Syntax
  public void addTextRotate(int rotate) throws Exception
- Parameter
  - rotate : Specifies the text alignment.

Set value	Description
Sam4sBuilder.TRUE	Specifies rotated printing of text.
Sam4sBuilder.FALSE (default)	Cancels rotated printing of text.

- Example
  To set alignment to the center:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addTextAlign(Sam4sBuilder.ALIGN_RIGHT);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addText

- » Adds the printing of text to the command buffer.
- » After printing text, to print content other than text, execute line feed or paper feed

- Syntax
  public void addText(java.lang.String data) throws Exception
- Parameter
  - data : Specify a character string to be printed.
    For the horizontal tab / line feed, use the following escape sequences:

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

- Example
  To add character strings:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addTextAlign(Sam4sBuilder.ALIGN_RIGHT);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextLang

» Adds the language setting to a command buffer. Encodes the string specified by addText (p.29) according to the language information specified by this API.

» This API is an API to be called before calling addText.

- Syntax
  public void addTextLang(int lang) throws Exception
- Parameter
  - lang : Specify a target language.

Set value	Language
Sam4sBuilder.LANG_EN (default)	English
Sam4sBuilder.LANG_KO	Korean

- Example
  To add character strings:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addTextDouble (Sam4sBuilder.LANG_EN);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextFont

» Adds the text font setting to the command buffer.

- Syntax
  public void addTextFont(int font) throws Exception
- Parameter
  - font : Specifies the font.

Set value	Language
Sam4sBuilder.FONT_A (default)	FONT A
Sam4sBuilder.FONT_B	FONT B

- Example
  To set the font B:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addTextFont (Sam4sBuilder.FONT_B);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addTextDouble

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

- Syntax
  public void addTextDouble(boolean dW, boolean dH) throws Exception
- Parameter
  - dw : Specify a target language.

Set value	Description
true	Specifies the double-sized width.
false	Cancels the double-sized width.

  - dh : Specify a target language.

Set value	Description
true	Specifies the double-sized height.
false	Cancels the double-sized height.

- Example
  To set the size as double width and height:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addTextDouble (true, true);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addTextSize

» Adds the text scale setting to the command buffer.

- Syntax
  public void addTextSize(int width, int height) throws Exception
- Parameter
  - width : Specifies the horizontal scale of text.

Set value	Description
Integer from 1 to 8	Horizontal scale (default : 1)

  - height : Specifies the vertical scale of text.

Set value	Description
Integer from 1 to 8	Vertical scale (default : 1)

- Example
  To set the size as double width and height:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addTextDouble (true, true);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextStyle

» Adds the text style setting to the command buffer.

- Syntax
  public void addTextStyle(boolean reverse, boolean ul, boolean em, int color) throws Exception
- Parameter
  - reverse : Specifies inversion of black and white for text.

Set value	Description
true	Specifies the inversion of black and white parts of characters.
false	Cancels the inversion of black and white parts of characters.

- ul : Specifies the underline style.

Set value	Description
true	Specifies underlining.
false	Cancels underlining.

- em : Specifies the bold style.

Set value	Description
true	Specifies emphasized printing of characters.
false	Cancels emphasized printing of characters.

- color : Specifies the color.

Set value	Description
Sam4sBuilder.COLOR_NONE	Characters are not printed.
Sam4sBuilder.COLOR_1(defult)	First color
Sam4sBuilder.COLOR_2	Second color

- Example
  To set the size as double width and height:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
 builder.addTextStype(false, true, false, Sam4sBuilder.COLOR_1);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextPosition

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

- Syntax
  public void addTextPosition(int x) throws Exception
- Parameter
  - x : Specifies the horizontal print start position (in dots).
    Specifies an integer from 0 to 65535.

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

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addTextPosition(120);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addTextDirection

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

- Syntax
  public void addTextDirection(int direction) throws Exception
- Parameter
  - direction : Specifies direction of text.

Set value	Description
Sam4sBuilder.DIRECTION_LEFT_TO_RIGHT(default)	Specifies the text direction as left to right.
Sam4sBuilder.DIRECTION_BOTTOM_TO_TOP	Specifies the text direction as bottom to top.
Sam4sBuilder.DIRECTION_RIGHT_TO_LEFT	Specifies the text direction as right to left.
Sam4sBuilder.DIRECTION_TOP_TO_BOTTOM	Specifies the text direction as to bottom.

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

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addTextDirection(Sam4sBuilder.DIRECTION_LEFT_TO_RIGHT);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addReverse

» Adds the text reverse setting to the command buffer.

- Syntax
  public void addReverse(boolean isOn) throws Exception
- Parameter
  - isOn : Specifies inversion of black and white for text.

Set value	Description
true	Specifies the inversion of black and white parts of characters.
false	Cancels the inversion of black and white parts of characters.

- Example
  To set the text reverse:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
 builder.addReverse(true);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addTextBold

» Adds the text bold setting to the command buffer.

- Syntax
  public void addTextBold(boolean isOn) throws Exception
- Parameter
  - isOn : Specifies the bold style.

Set value	Description
true	Specifies emphasized printing of characters.
false	Cancels emphasized printing of characters.

- Example
  To set the text bold:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
 builder.addTextBold (true);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addTextUnderline

» Adds the text underline setting to the command buffer.

- Syntax
  public void addTextUnderline(boolean isOn) throws Exception
- Parameter
  - isOn : Specifies the bold style.

Set value	Description
true	Specifies underlining.
false	Cancels underlining.

- Example
  To set the text underline:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addTextBold(true);
}catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addFeedUnit

» Adds paper feeding in dots to the command buffer.

- Syntax
  public void addFeedUnit(int unit) throws Exception
- Parameter
  - unit : Specifies the paper feed space (in dots).
    Specifies an integer from 0 to 255.
- Example
  To feed paper by 30 dots:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
 builder.addFeedUnit(30);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addFeedLine

» Adds paper feeding in lines to the command buffer.

- Syntax
  public void addFeedLine(int line) throws Exception
- Parameter
  - line : Specifies the paper feed space (in lines).
    Specifies an integer from 0 to 255.
- Example
  To feed paper by 3 lines:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addFeedLine(3);
}catch (Exception e) {
 e.printStackTrace();
}
```

addImage

» Adds the text reverse setting to the command buffer.

- Syntax
  public void addImage(android.graphics.Bitmap data, int width, int height) throws Exception
- Parameter
  - data : Specifies an instance of the android.graphics.Bitmap class.
  - width : Specifies the width of the print area.
    Specifies an integer from 1 to 65535.
  - height : Specifies the height of the print area.
    Specifies an integer from 1 to 65535.
- Example
  To print an image 256 dots wide and 256 dots high:

```
try{
    Bitmap imagedata = null;
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addImage(imagedata, 256, 256);
}
catch (Exception e) {
 e.printStackTrace();
}
```

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

```
try{
    Bitmap imagedata = null;
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addPagePosition(0, 255);
builder.addImage(imagedata, 256, 256);
builder.addPageEnd();
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addLogo

» Adds NV logo printing to the command buffer.

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

» Register a logo in advance into the printer using the following utilities:
EllixTool / EllixSet / GIANTTOOL

- Syntax
  public void addLogo(int imageNo, int logoMode) throws Exception
- Parameter
  - imageNo : Specifies the number of NV logo.
  - logoMode : Specifies the style of the print logo.

Set value	Description
Sam4sBuilder .LOGO_NORMAL	Print NV logo as normal size.
Sam4sBuilder .LOGO_WIDTH_DOUBLE	Print NV logo as double width size
Sam4sBuilder .LOGO_HEIGHT_DOUBLE	Print NV logo as double height size
Sam4sBuilder .LOGO_WIDTH_HEIGHT_DOUBLE	Print NV logo as double width and double height size.

- Example
  To print first NV logo if the logo is exists:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
  builder.addLogo (0, Sam4sBuilder.LOGO_NORMAL);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addBarcode

» Adds barcode printing to the command buffer.

- Syntax
  public void addBarcode(java.lang.String data, int type, int hri, int font, int width, int height) throws Exception
- Parameter
  - data : Specifies the barcode data as a string.

Barcode type	Description
UPC-A	When an 11-digit number is specified, a check digit is automatically added. When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
UPC-E	Specify 0 as the first digit. Specify the manufacturer code in the digits 2 to 6. Specify (right-align) the item code in the digits 7 to 11. The number of item code digits varies depending on the manufacturer code. Specify 0s in empty digits.
EAN13	When an 11-digit number is specified, a check digit is automatically added. When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
JAN13	
EAN8	When a 7-digit number is specified, a check digit is automatically added. When an 8-digit number is specified, the 8th digit is processed as a check digit but the check digit is not validated.
JAN8	
CODE39	When the first character is *, the character is processed as the start character. In other cases, a start character is automatically added.
ITF	Start and stop codes are automatically added. Check digits are not added or validated.
CODABAR	Specify a start character (A to D, a to d). Specify a stop character (A to D, a to d). Check digits are not added or validated.
CODE93	Start and stop characters are automatically added. A check digit is automatically calculated and added.
CODE128	Specify a start character (CODE A, CODE B, CODE C). A stop character is automatically added. A check digit is automatically calculated and added. To encode each of the following characters, specify two characters starting with the character "{": FNC1: {1 FNC2: {2 FNC3: {3 FNC4: {4 CODE A: {A CODE B: {B CODE C: {C SHIFT: {S {: {{

4.2. Sam4sBuilder class

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

String	Description
＼xnn	Control code
＼＼	Back slash

- type : Specifies the barcode type.

Set value	Barcode type
Sam4sBuilder.BARCODE_UPC_A	UPC-A
Sam4sBuilder.BARCODE_UPC_E	UPC-E
Sam4sBuilder.BARCODE_EAN13	EAN13
Sam4sBuilder.BARCODE_JAN13	JAN13
Sam4sBuilder.BARCODE_EAN8	EAN8
Sam4sBuilder.BARCODE_JAN8	JAN8
Sam4sBuilder.BARCODE_CODE39	CODE39
Sam4sBuilder.BARCODE_ITF	ITF
Sam4sBuilder.BARCODE_CODABAR	CODABAR
Sam4sBuilder.BARCODE_CODE93	CODE93
Sam4sBuilder.BARCODE_CODE128	CODE128

- hri : Specifies the HRI position.

Set value	Barcode type
Sam4sBuilder.HRI_NONE	HRI not printed
Sam4sBuilder.HRI_ABOVE	Above the barcode
Sam4sBuilder.HRI_BELOW	Below the barcode
Sam4sBuilder.HRI_BOTH	Both above and below the barcode

- font : Specifies the HRI font.

Set value	Barcode type
Sam4sBuilder.FONT_A (default)	FONT A
Sam4sBuilder.FONT_B	FONT B

- width : Specifies the width of each module in dots.
  Specifies an integer from 2 to 6.
- height : Specifies the barcode height in dots.
  Specifies an integer from 1 to 255.

4.2. Sam4sBuilder class

- Example
  To print barcodes:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addBarcode (“01234567890”, Sam4sBuilder.BARCODE_UPC_A,
                    Sam4sBuilder.HRI_BELOW, Sam4sBuilder.FONT_A, 3, 162);
builder.addBarcode (“01234500005”, Sam4sBuilder.BARCODE_UPC_E,
                   Sam4sBuilder.HRI_ABOVE, Sam4sBuilder.FONT_B, 3, 162);
builder.addBarcode (“201234567890”, Sam4sBuilder.BARCODE_EAN13,
                    Sam4sBuilder.HRI_BOTH, Sam4sBuilder.FONT_A, 3, 162);
builder.addBarcode (“201234567890”, Sam4sBuilder.BARCODE_JAN13,
                    Sam4sBuilder.HRI_NONE, Sam4sBuilder.FONT_B, 3, 162);
builder.addBarcode (“2012345”, Sam4sBuilder.BARCODE_EAN8 ,
                    Sam4sBuilder.HRI_BELOW, Sam4sBuilder.FONT_A, 3, 162);
builder.addBarcode (“2012345”, Sam4sBuilder.BARCODE_JAN8,
                    Sam4sBuilder.HRI_ABOVE, Sam4sBuilder.FONT_B, 3, 162);
builder.addBarcode (“ABCDE”, Sam4sBuilder.BARCODE_CODE39,
                    Sam4sBuilder.HRI_BOTH, Sam4sBuilder.FONT_A, 3, 162);
builder.addBarcode (“012345”, Sam4sBuilder.BARCODE_ITF,
                    Sam4sBuilder.HRI_NONE, Sam4sBuilder.FONT_B, 3, 162);
builder.addBarcode (“A012345A”, Sam4sBuilder.BARCODE_CODABAR,
                    Sam4sBuilder.HRI_NONE, Sam4sBuilder.FONT_B, 3, 162);
builder.addBarcode (“ABCED”, Sam4sBuilder.BARCODE_CODE93,
                    Sam4sBuilder.HRI_BELOW, Sam4sBuilder.FONT_A, 3, 162);
builder.addBarcode (“{Babcde”, Sam4sBuilder.BARCODE_CODE128,
                    Sam4sBuilder.HRI_BELOW, Sam4sBuilder.FONT_B, 3, 162);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addSymbol

» Adds 2D-Code printing to the command buffer.

- Syntax
  public void addSymbol(java.lang.String data, int type, int level, int width, int height, int size) throws Exception
- Parameter
  - data : Specifies 2D-Code data as a string.

Barcode type	Description
PFD417	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. The data area can contain up to 928 code words in a maximum of 90 rows, each of which can contain up to 30 code words.
QR code	Convert the character string to the string in Shift-JIS, apply the escape sequence, and then encode the string based on the data type as shown below. Number: 0 to 9 Alphanumeric character: 0 to 9, A to Z, space, $, %, *, +, -, ., /, : 8-but, byte data: 0x00 to 0xff

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

String	Description
＼xnn	Control code
＼＼	Back slash

- type : Specifies the 2D-Code type.

Set value	Barcode type
Sam4sBuilder.SYMBOL_PDF417_STANDARD	PDF417
Sam4sBuilder.SYMBOL_QRCODE_MODEL	QR code

- level : Specifies the error correction level.
  Specifies from Sam4sBuilder.LEVEL_0 to Sam4sBuilder_LEVEL_8.
- width : Specifies the width of each module in dots.
  Specifies an integer from 0 to 255.
- height : Specifies the barcode height in dots.
  Specifies an integer from 0 to 255. (QR code is ignored)
- size : Specifies the 2D-Code maximum size. (QR code is ignored)
  Specifies an integer from 0 to 65535.

4.2. Sam4sBuilder class

- Example
  To print 2D-Codes:

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addSymbol (“ABCDE”, Sam4sBuilder.SYMBOL_PDF417_STANDARD,
                Sam4sBuilder.LEVEL_1, 50, 100, 0);
builder.addSymbol (“ABCDE”, Sam4sBuilder.SYMBOL_QRCODE_MODEL,
                Sam4sBuilder.LEVEL_1, 50, 100, 0);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addCut

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

» Not available in page mode.

- Syntax
  public void addCut(int type) throws Exception
- Parameter
  - type : Specifies the paper cut type

Set value	Description
Sam4sBuilder.CUT_NO_FEED	Cut without feeding.
Sam4sBuilder.CUT_FEED	Feed cut

- Example
  To perform feed cut operation:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addCut(Sam4sBuilder.CUT_FEED);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addOpenCashDrawer

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

» Not available in page mode.

- Syntax
  public void addOpenCashDrawer(int drawerNo, int time) throws Exception
- Parameter
  - drawerNo : Specifies the drawer kick connector.

Set value	Description
Sam4sBuilder.DRAWER_1	Pin 2 of the drawer kick-out connector.
Sam4sBuilder.DRAWER_2	Pin 5 of the drawer kick-out connector.

- time : Specifies the ON time of the drawer kick signal.
  Specifies an integer from 0 to 255.

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

```
Try{
Sam4sBuilder builder = new Sam4sBuilder("ELLIX30", Sam4sBuilder.LANG_EN);
builder.addOpenCashDrawer(Sam4sBuilder.DRAWER_1, 100);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addCommand

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

» ESC / POS commands are not made public. For details, contact the dealer.

- Syntax
  public void addCommand(byte[] data) throws Exception
- Parameter
  - data : Specifies ESC / POS command as a string.
- Example

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
byte[] data =null;
builder.addCommand(data);
}
catch (Exception e) {
 e.printStackTrace();
}
```

addPageBegin

» Adds the switching to page mode to the command buffer.

» The page mode process start.

» Use this API function with addPageEnd.

- Syntax
  public void addPageBegin() throws Exception
- Example
  To print the characters “ABCD” in page mode:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addText(“ABCD”);
builder.addPageEnd();
}
catch (Exception e) {
 e.printStackTrace();
}
```

addPageEnd

» Adds the end of page mode to the command buffer.

» The page mode process ends.

» Use this API function with addPageBegin.

- Syntax
  public void addPageEnd() throws Exception
- Example
  To print the characters “ABCD” in page mode:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addText(“ABCD”);
builder.addPageEnd();
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addPageArea

- » Adds the print area in page mode to the command buffer.
- » Specifies the print area in page mode (coordinates). After this API function, specify a print data API function such as the addText method.

  - Syntax
    public void addPageArea(int x, int y, int width, int height) throws Exception
  - Parameter
    - x : Specifies the origin of the horizontal axis (in dots).
      Specifies an integer from 0 to 65535.
      0 is the left end of the printer's printable area.
    - y : Specifies the origin of the vertical axis (in dots).
      Specifies an integer from 0 to 65535.
      0 is the position in which no paper feed has been performed.
    - width : Specifies the width of the print area (in dots).
      Specifies an integer from 0 to 65535.
    - height : Specifies the height of the print area (in dots).
      Specifies an integer from 0 to 65535.

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

  - Example
    To specify the print area with the origin (100, 50), a width of 200 dots, and a height of 30 dots and print the characters “ABCD”:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addPageArea(100, 50, 200, 30);
builder.addText(“ABCD”);
builder.addPageEnd();
}
catch (Exception e) {
 e.printStackTrace();
}
```

addPageDirection

» Adds the page mode print direction setting to the command buffer.

» Specifies the print direction in page mode.

» Use this API function by inserting it between addPageBegin and addPageEnd

- Syntax
  public void addPageDirection(int dir) throws Exception
- Parameter
  - dir : Specifies the print direction in page mode.

Set value	Description
Sam4sBuilder.DIRECTION_LEFT_TO_RIGHT(default)	Specifies the text direction as left to right.
Sam4sBuilder.DIRECTION_BOTTOM_TO_TOP	Specifies the text direction as bottom to top.
Sam4sBuilder.DIRECTION_RIGHT_TO_LEFT	Specifies the text direction as right to left.
Sam4sBuilder.DIRECTION_TOP_TO_BOTTOM	Specifies the text direction as to bottom.

- Example
  To print the characters “ABCD” by rotating them 90 degrees clockwise:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addPageArea(100, 50, 200, 30);
builder.addPageDirection(Sam4sBuilder.DIRECTION_TOP_TO_BOTTOM);
builder.addText(“ABCD”);
builder.addPageEnd();
}
catch (Exception e) {
 e.printStackTrace();
}
```

addPagePosition

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

» Specifies the print area in page mode (coordinates) in the area specified by the addPageArea method.

» Use this API function by inserting it between addPageBegin and addPageEnd.

- Syntax
  public void addPagePosition(int x, int y) throws Exception
- Parameter
  - x : Specifies the horizontal print position (in dots).
    Specifies an integer from 0 to 65535.
    0 is the left end of the printer’s printable area.
  - y : Specifies the vertical print position (in dots).
    Specifies an integer from 0 to 65535.

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

* To print a character string:
Specify the left end of the baseline for first character. This can be omitted for left-aligned printing of standard-sized characters. To print double-sized height characters, specify a value equal to or greater than 42 for y.

* To print a barcode/graphic/logo: Specify the bottom left of the symbol / graphic data. And specify the barcode height for y.

* To print a 2D-Code: Specify the top left of the symbol.
This can be omitted when printing from the top left.

- Example
  To specify (50, 30) for the print start position in the area specified by the addPageArea method and print the characters “ABCD”:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addPageArea(100, 50, 200, 30);
builder.addPagePosition(50, 30);
builder.addText(“ABCD”);
builder.addPageEnd();
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.2. Sam4sBuilder class

addPageFormFeed

» Adds page mode form feeding to the command buffer.

- Syntax
  public void addPageFormFeed() throws Exception
- Example

```
try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
builder.addPageBegin();
builder.addPageArea(100, 50, 200, 30);
builder.addPagePosition(50, 30);
builder.addText(“ABCD”);
builder.addFormFeed();
builder.addText(“EFGH”);
builder.addPageEnd();
}catch (Exception e) {
 e.printStackTrace();
}
```

4.3. Sam4sPrint class

Constructor

» Constructor for the Sam4sPrint class. Initializes an Sam4sPrint class instance.

- Syntax
  public Sam4sPrint()
- Example

```
Sam4sPrint printer = new Sam4sPrint();
```

4.3. Sam4sPrint class

openPrinter

» Starts communication with the printer.

» If communication with the printer is not required anymore, be sure to call closePrinter, closePrinter API, to end communication with the printer.

» When you are opening the printer with another application via a Bluetooth connection, if you try to begin communication with this API, the process may not return.

- Syntax
  public void openPrinter(int deviceType, java.lang.String deviceName) throws Exception
- Parameter
  - deviceType : Specifies the type for the device to start communication.

Set value	Description
Sam4sPrint.DEVTYPE_ETHERNET	Wi-Fi / Ethernet device
Sam4sPrint.DEVTYPE_BLUETOOTH	Bluetooth device
Sam4sPrint.DEVTYPE_SERIAL_SAP4K	SAP4K VCOM device

- deviceType : Specifies the type for the device to start communication.
  Specifies the identifier used for identification of the target device.
  Specifies the following for each device type:

Set value	Specified Value
Sam4sPrint.DEVTYPE_ETHERNET	IP address (IPv4)
Sam4sPrint.DEVTYPE_BLUETOOTH	MAC address
Sam4sPrint.DEVTYPE_SERIAL_SAP4K	SAP4K VCOM port

- Example
  To start communication via Wi-Fi / Ethernet with the printer:

```
Try{
Sam4sPrint printer = new Sam4sPrint();
printer.openPrinter(Sam4sPrinter.DEVTYPE_EHTERNET, “192.168.0.168”);
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.3. Sam4sPrint class

closePrinter

» Ends communication with the printer.

- Syntax
  public closePrinter()
- Example

```
Try{
Sam4sPrint printer = new Sam4sPrint();
printer.openPrinter(Sam4sPrinter.DEVTYPE_EHTERNET, “192.168.0.168”);
printer.closePrinter() ;
}
catch (Exception e) {
 e.printStackTrace();
}
```

sendData

» Sends a print document created using the Sam4sBuilder class

- Syntax
  public void sendData(com.Sam4s.print.Sam4sBuilder builder) throws Exception
- Parameter
  - builder : Specifies a Sam4sBuilder class instance.
  For details on the Sam4sBuilder class, refer to Sam4sBuilder class.
- Example
  To send a command to the printer:

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
Sam4sPrint printer = new Sam4sPrint();
printer.openPrinter(Sam4sPrinter.DEVTYPE_EHTERNET, “192.168.0.168”);
printer.sendData(builder) ;
printer.closePrinter() ;
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.3. Sam4sPrint class

getPrinterName

» Acquires the printer name.

- Syntax
  public String getPrinterName()
- Return value
  The name of printer is returned.
- Example

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
Sam4sPrint printer = new Sam4sPrint();
printer.openPrinter(Sam4sPrinter.DEVTYPE_EHTERNET, “192.168.0.168”);
String printerName = printer.getPrinterName();
printer.closePrinter() ;
}
catch (Exception e) {
 e.printStackTrace();
}
```

getPrinterStatus

» Acquires the printer status.

- Syntax
  public String getPrinterStatus()
- Return value
  The status of printer is returned.
- Example

```
Try{
Sam4sBuilder builder = new Sam4sBuilder(“ELLIX30”, Sam4sBuilder.LANG_EN);
Sam4sPrint printer = new Sam4sPrint();
printer.openPrinter(Sam4sPrinter.DEVTYPE_EHTERNET, “192.168.0.168”);
String printerStatus = printer.getPrinterStatus();
printer.closePrinter() ;
}
catch (Exception e) {
 e.printStackTrace();
}
```

4.4. Sam4sFinder class

Constructor

» Constructor for the Sam4sFinder class. Initializes an Sam4sFinder class instance.

- Syntax
  public Sam4sFinder()
- Example

```
Sam4sFinder finder = new Sam4sFinder();
```

StartSearch

» Starts a search for printers of the specified device type.

» If you use this API, be sure to use stop to stop the search.

- Syntax
  public void startSearch(Context mContext, int mDeviceType)
- Parameter
  - mContext : Set a Context class instance of caller.
    (Example: Set the Context acquired by getBaseContext() in Activity.)
  - mDeviceType : Specifies the identifier used for identification of the target device.

Set value	Description
Sam4sFinder.DEVTYPE_ETHERNET	Searches for Sam4s devices connected to the network.
Sam4sFinder.DEVTYPE_BLUETOOTH	Searches for Bluetooth devices.

- Example

```
Sam4sFinder finder = new Sam4sFinder();
finder.startSearch(getBaseContext(), Sam4sFinder.DEVTYPE_ETHERNET) ;
```

stopSearch

» Stops the printer search.

- Syntax
  public void stopSearch()
- Example

```
Sam4sFinder finder = new Sam4sFinder();
finder.startSearch(getBaseContext(),Sam4sFinder.DEVTYPE_ETHERNET) ;
finder.stopSearch();
```

4.4. Sam4sFinder class

getResult

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

- Syntax
  public String[] getResult()
- Return value
  The list of devices found during search is returned.
  Identification information of the found devices is stored as a character string (String type) in the list.
  The stored results differ depending on the type of device (device type).

Device type	List to Acquire
ETHERNET	List of IP address (IPv4)
BLUETOOTH	List of MAC address

SAM4S Printer SDK for Android
User's Manual

User Guide