

Confidential

EPSON

074-09-PDS-001

Roadrunner X

Specification

STANDARD	
Rev. No.	1.00
Notes	LDC0074-09

EPSON AMERICA, INC.

3840 KILROY AIRPORT WAY
LONG BEACH, CA 90806-2469
PHONE (562) 981-3840

REVISION SHEET

Sheet 1 of 2

The table below indicates which pages in this specification have been revised.
Before reading this specification, be sure you have the correct version of each page.

[illegible]

Confidential

REVISION SHEET

Sheet 2 of 2

[illegible]

Confidential

SCOPE

The scope of this specification is restricted to those areas indicated with a filled box in the "Check" column below. (☒ denotes the purpose for which this specification is to be used.)

"Studies": At this stage, values that affect the basic design are subject to change.

"Design": Design work may begin at this stage, after inquiring about the latest specifications.
(To change the Checkbox status, Do right click\properties)

SCOPE		
Check	Area of application	Restrictions on scope
<input type="checkbox"/>	IS	
<input type="checkbox"/>	WS	
<input type="checkbox"/>	ES	
<input type="checkbox"/>	MP	
<input type="checkbox"/>	Marketing	
<input type="checkbox"/>	User planning studies	
<input type="checkbox"/>	Circuit studies	
<input type="checkbox"/>	Circuit design	
<input type="checkbox"/>	Case studies	
<input type="checkbox"/>	Case design	
<input type="checkbox"/>	LSI studies	
<input type="checkbox"/>	LSI design	
<input type="checkbox"/>	Software studies	
<input checked="" type="checkbox"/>	Software design	
<input type="checkbox"/>		
<input type="checkbox"/>		
<input type="checkbox"/>		
EPSON		

CONFIDENTIALITY AGREEMENT

BY USING THIS DOCUMENT, YOU AGREE TO ABIDE BY THE TERMS OF THIS AGREEMENT.

PLEASE RETURN THIS DOCUMENT IMMEDIATELY IF YOU DO NOT AGREE TO THESE TERMS.

1. This document contains confidential, proprietary information of Seiko Epson Corporation or its affiliates. You must keep such information confidential. If the user is a business entity or organization, you must limit disclosure to your employees, agents, and contractors who have a need to know and who are also bound by obligations of confidentiality.
2. On the earlier of (a) termination of your relationship with Seiko Epson, or (b) Seiko Epson's request, you must stop using the confidential information. You must then return or destroy the information, as directed by Seiko Epson.
3. If a court, arbitrator, government agency, or the like orders you to disclose any confidential information, you must immediately notify Seiko Epson. You agree to give Seiko Epson reasonable cooperation and assistance in the negotiation.
4. You may use confidential information only for the purpose of operating or servicing the products to which the document relates, unless you obtain the prior written consent of Seiko Epson for some other use.
5. Seiko Epson warrants that it has the right to disclose the confidential information. SEIKO EPSON MAKES NO OTHER WARRANTIES CONCERNING THE CONFIDENTIAL INFORMATION OR ANY OTHER INFORMATION IN THE DOCUMENT, INCLUDING (WITHOUT LIMITATION) ANY WARRANTY OF TITLE OR NON-INFRINGEMENT. Seiko Epson has no liability for loss or damage arising from or relating to your use of or reliance on the information in the document.
6. You may not reproduce, store, or transmit the confidential information in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise) without the prior written permission of Seiko Epson.
7. Your obligations under this Agreement are in addition to any other legal obligations. Seiko Epson does not waive any right under this Agreement by failing to exercise it. The laws of Japan apply to this Agreement.

Cautions

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

Trademarks

EPSON® and ESC/POS® are registered trademarks of Seiko Epson Corporation.

General Notice: Other product and company names used herein are for identification purposes only and may be trademarks of their respective companies.

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
				SHEET I

Table of Contents

GENERAL	1
General / Introduction	1
Development Environment.....	1
System Design.....	1
Usage of the Library	3
Installation	3
Using the classes and methods.....	4
Basic Operations.....	5
Establishing a connection to the printer	5
Receipt Printing	5
Slip Printing	6
Reading MICR.....	6
Scanning Check	7
Error Handling	7
Using Ethernet Connection	8
Infrastructure Mode	8
Ad-hoc mode.....	8
AndroidManifest.xml.....	8
Opening the port:.....	8
Using USB Connection	9
Android OS Version.....	9
Cable	9
AndroidManifest.xml.....	9
device_filter.xml	9
Access permissions	9
Opening the port:.....	10
The Functions	11
Overview	11
The DeviceParameters Class.....	13
Field Summary.....	13
Constructor Summary.....	13
Method Summary.....	13
Field Detail	14
PortType	14
PortName	14
PortNumber.....	14
IPAddress.....	14
DeviceID	14
DeviceName.....	14
Constructor Detail	15
EpsonComDeviceParameters.....	15
Method Detail	15
validateParameters	15
copy.....	15
The Device Class	16
Constructor Summary.....	16
Method Summary.....	16
Constructor Detail	19

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001
			SHEET II

EpsonComDevice	19
Method Detail	19
setDeviceParameters	19
openDevice	20
closeDevice	20
isDeviceOpen	20
sendData	20
sendCommand	21
getReceivedData	21
requestDeviceStatus	21
getDeviceStatus	22
activateASB	22
readMICR	22
getASB	23
getMICRString	23
getImageData	23
getImageDataBytes	24
setImageParameters	24
setScanArea	24
setCropArea	25
deleteCropArea	25
selectImageFormat	25
selectReceiptPaper	26
selectSlipPaper	26
selectAlignment	26
selectPageMode	27
selectPrintDirection	27
feedSlipToPrintStartPosition	27
ejectCheck	27
cancelWaitForSlip	28
printString	28
printPage	28
cutPaper	29
scanCheck	29
readMICRScanCheck	29
scanIDCard	30
resetDevice	30
registerCallback	30
unregisterCallback	31
checkAndSetImageParameters	31
The ASBStatus Class	32
Field Summary	32
Constructor Summary	33
Method Summary	33
Field Detail	33
DrawerKickoutConnectorPin3High	33
Online	33
CoverOpen	33
PaperFedByFeedButton	34
WaitingForOnlineRecovery	34
PaperFeedButtonIsTurnedOn	34
MechanicalError	34

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001
			SHEET III

AutoCutterError	34
UnrecoverableError	34
AutomaticallyRecoverableError	34
PaperNearEnd	34
PaperOut	34
PaperPresentAtTOFSensor	35
PaperPresentAtBOFSensor	35
SlipSelectedAsActiveSheet	35
CanPrintOnSlip	35
Constructor Detail	35
EpsonComASBStatus	35
Method Detail	35
toString	35
APPENDIX A – Error Codes	36
APPENDIX B – Defined Constants	37
Port Type (in the current version, only ETHERNET and USB are supported)	37
Data Type	37
ASCII Control Codes	37
Print Alignment	38
Paper Side	38
Print Direction	38
Print Font	39
MICR Font of the check to be read	39
Image Format of the image data	39
Bit Depth of the image data	39
Image Processing after scanning	39

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
				SHEET IV

GENERAL

General / Introduction

The EpsonCom library is a general purpose communications library that enables any Android application (on either a smart phone or tablet computer) to use EPSON's TM printers. This is done by offering a simple, logical and consistent programmer's interface (API).

Target applications are any Android applications that want to make use of TM functionality, like retail, hospitality or teller applications.

Development Environment

The EpsonCom library is developed in the standard Android SDK which consists of Java SDK and the Eclipse development environment, including the Android plug-ins. It was developed on Mac OS X but due to the platform-independent nature of Java and Eclipse, development can also be done on Windows or Linux.

The library is a static Java library (JAR) which can easily be linked into any Java for Android application development project.

System Design

The EpsonCom library offers classes and methods that make it very easy to use all the needed functionality of the TM device. Included functions are (depending on the connected printer type):

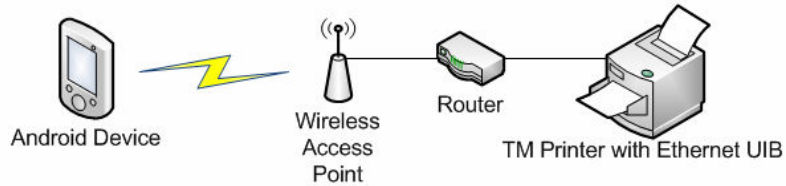
- receipt printing
- slip inserting/printing
- check MICR reading
- check image scanning
- check endorsement printing
- real-time status inquiring

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 1

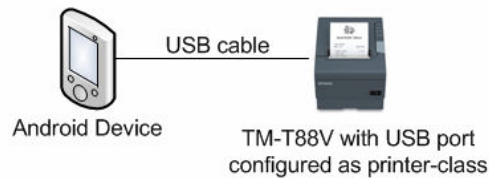
Confidential

Different technologies for connecting to the TM device are planned, however in this first stage only Ethernet and USB connection is implemented. The physical connection between Android device and printer can be done using one of two possible scenarios (Please note: as of version 3.2 Android does **not support Ad-hoc mode** WiFi connections):

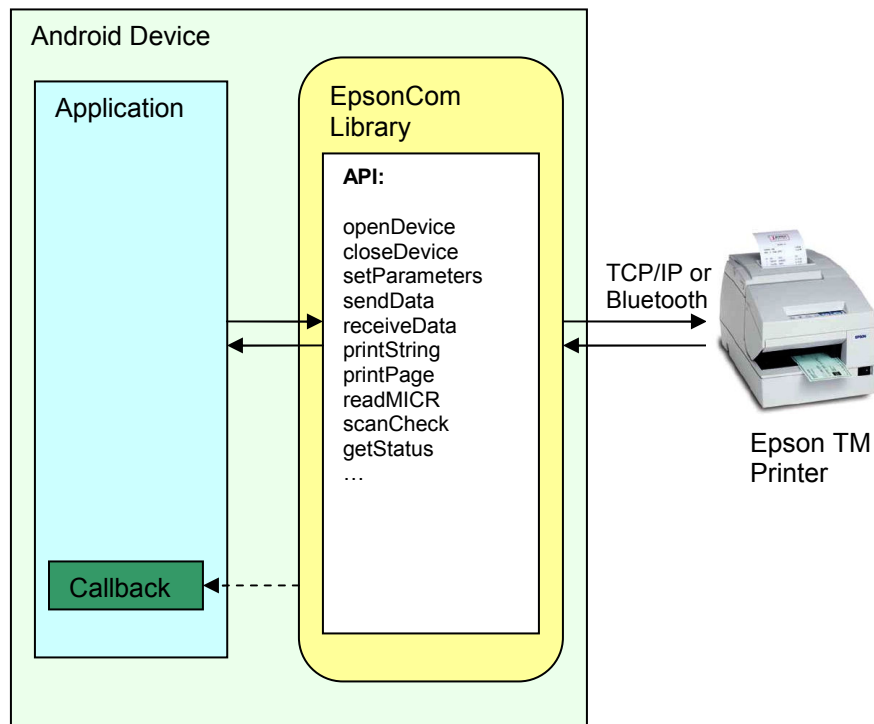
Indirect WiFi connection (Infrastructure)



USB connection



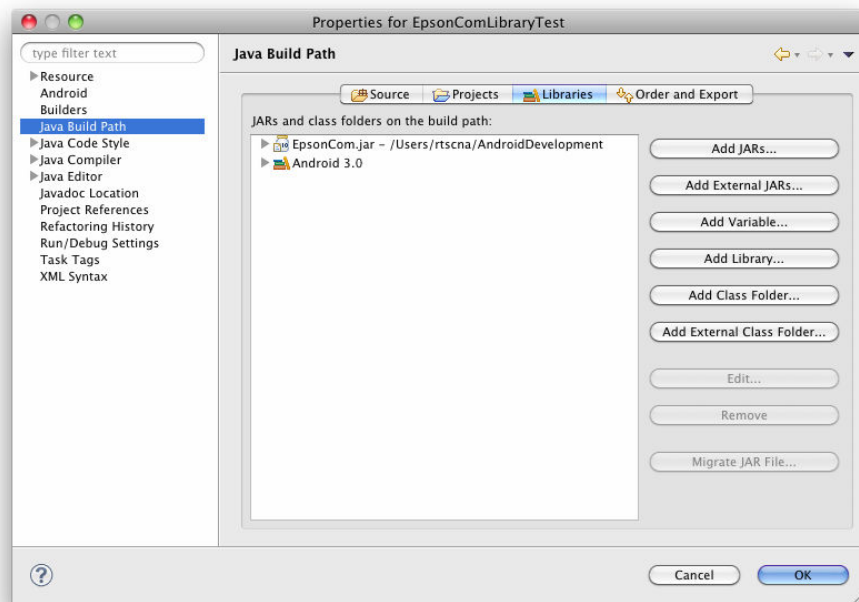
The following figure shows the functional principle of the EpsonCom library:



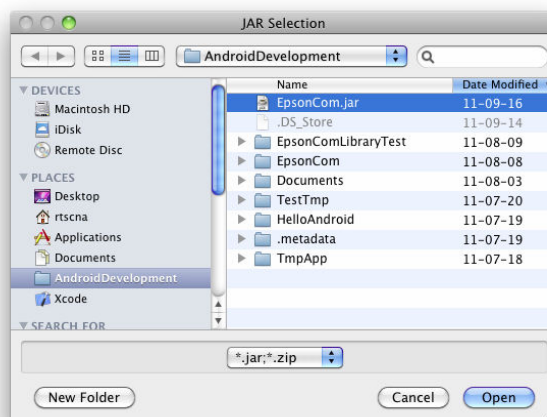
EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 2

Usage of the Library Installation

To include the library into your Eclipse project, simply copy the file *EpsonCom.jar* onto your hard drive. Then, in Eclipse, go to menu *Project/Properties* and select *Java Build Path* in the list on the left side. Then click on the *Libraries* tab as shown here:



Click *Add External JARs...* and select *EpsonCom.jar* from wherever you had copied it to.



EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 3

Using the classes and methods

To use the library classes and methods in a specific class, the different EpsonCom packages need to be imported. This can be done either manually by adding:

```
import com.epson.EpsonCom.CallbackInterface;
import com.epson.EpsonCom.EpsonCom;
import com.epson.EpsonCom.EpsonComASBStatus;
import com.epson.EpsonCom.EpsonComCallbackInfo;
import com.epson.EpsonCom.EpsonComDevice;
import com.epson.EpsonCom.EpsonComDeviceParameters;
```

Or it can be done automatically by pressing Ctrl-Shift-O whenever you use a new EpsonCom class.

The library classes can be used like any other classes, e.g.:

```
EpsonComDevice dev = new EpsonComDevice();
EpsonComDeviceParameters devParams = new EpsonComDeviceParameters();
etc...
```

Enums with constants have been defined for all parameters that constitute a choice. These are all defined inside the *EpsonCom* class. For a list of all defined constants, see [APPENDIX B](#).

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 4

Basic Operations

Establishing a connection to the printer

Before anything can be printed, a connection to the printer device has to be established.

1. Creating a Device and a DeviceParameters object:

```
EpsonComDevice dev = new EpsonComDevice();  
EpsonComDeviceParameters devParams = new EpsonComDeviceParameters();
```

2. Setting the device parameters

This example uses an Ethernet/WiFi connection (USB is also possible, see separate chapter below):

```
devParams.PortType = EpsonCom.PORT_TYPE.ETHERNET;  
devParams.IPAddress = "192.168.192.123";  
devParams.PortNumber = 9100;  
dev.setDeviceParameters(devParams);
```

Tip: by importing the PORT_TYPE enum:

```
import com.epson.EpsonCom.EpsonCom.PORT_TYPE;
```

the EpsonCom prefix can be omitted:

```
devParams.PortType = PORT_TYPE.ETHERNET;
```

3. Opening the device

```
dev.openDevice();
```

4. Closing the device

After all the operations are done, the device needs to be closed and the memory released:

```
dev.closeDevice();
```

Receipt Printing

After doing step 1 to 3 from the above, the printer is ready to print. Printing can be done either in Line Mode or in Page Mode. In Line Mode the printer prints each submitted line right away, in Page Mode the printer collects all the data of one page and then prints it all together. The advantage of this is higher print speed and it also makes it possible to print upside down or vertically.

○ Printing in Line Mode:

```
dev.selectReceiptPaper();  
dev.selectAlignment(ALIGNMENT.LEFT);  
dev.printString("Hello Printer", FONT.FONT_A, true, false, false,  
false);
```

○ Printing in Page Mode:

```
dev.selectReceiptPaper();
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 5

```
dev.selectPageMode();
dev.selectPrintDirection(PRINTDIRECTION.LEFTTORIGHT);
dev.selectAlignment(ALIGNMENT.JF_LEFT);
dev.printString("Hello Printer", FONT.FONT_A, true, false, false, false);
dev.printString("Line2", FONT.FONT_A, true, false, false, false);
dev.printString("Line3", FONT.FONT_A, true, false, false, false);
dev.printPage(0);
```

Slip Printing

This works the same as receipt printing, except the slip paper and the paper side is selected.

```
dev.selectSlipPaper(PAPERSIDE.FRONT);
dev.selectAlignment(ALIGNMENT.JF_LEFT);
dev.printString("Hello Printer", FONT.FONT_A, true, false, false, false);
```

As soon as the selectSlipPaper command is sent, the printer will blink the Slip LED and a paper slip needs to be inserted.

Page Mode works similar to the way it works on receipt paper but keep in mind, that printable area may be different and certain properties (like fonts or boldness) may not be available depending on the print mode and selected slip side. (For more details about this, please see the printer manual).

Reading MICR

Reading MICR and scanning an image takes a certain time and delivers data back from the printer. The application can either wait for the data (synchronous mode) or do something else in the meantime and get notified once the data is ready (asynchronous mode).

o Synchronous Mode:

```
dev.readMICR(MICR_FONT.MF_E13B, true, 10);
String MICRString = dev.getMICRString();
```

o Asynchronous Mode:

In order to be able to be notified, a callback method must be created and registered with the library. In order for the callback to work, the class must implement the CallbackInterface interface:

```
public class TestActivity extends Activity implements CallbackInterface
```

The callback method must look like this:

```
public ERROR_CODE CallbackMethod(EpsonComCallbackInfo cbInfo)
{
    ERROR_CODE retval = ERROR_CODE.SUCCESS;

    try
    {
        switch (cbInfo.ReceivedDataType)
        {
            case GENERAL:
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 6

```
        break;
    case ASB:
        break;
    case MICR:
        m_MICRString = dev.getMICRString();
        break;
    case IMAGE:
        m_ImageData = dev.getImageData();
    }
}
catch(Exception e)
{
    ... error handling ...
}
return retval;
}
```

Then the class that has the callback method must be registered with the *Device* object using the *registerCallback()* method, e.g:

```
dev.registerCallback(this);
```



Please note: that in order to have any effect, the registering must take place **before** the *openDevice()* method is called.

Requesting the MICR reading then looks like this:

```
dev.readMICR(MICR_FONT.E13B, false, 0);
```

Scanning Check

o Synchronous Mode:

```
dev.setImageParameters(BITDEPTH.GRAYSCALE, IMAGEPROCESSING.NONE, 0);
dev.selectImageFormat(IMAGEFORMAT.JPEG_HIGH);
dev.scanCheck(true, 200);
byte[] receivedBytes = dev.getImageDataBytes();
dev.ejectCheck();
```

o Asynchronous Mode:

Like reading the MICR, scanning asynchronously makes use of the callback technique. The same callback function as above is used to be notified and to pick up the actual image data.

```
dev.setImageParameters(BITDEPTH.GRAYSCALE, IMAGEPROCESSING.NONE, 0);
dev.selectImageFormat(IMAGEFORMAT.JPEG_HIGH);
dev.scanCheck(false, 0);
dev.ejectCheck();
```

Error Handling

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 7

All of the library commands with the exception of the *get...* and *is...* commands return an error code of type `ERROR_CODE`. It is highly recommended to always check any returned error code (for simplicity sake this was not done in the above samples).



Please note: since some of the commands, especially the action commands, just send a request to the printer, the returned error-code refers only to the process of sending the command to the printer, but not to the actual execution of that command on the printer. E.g. Sending a `readMICR` command to the printer might return `ERR_SUCCESS` even though there is a subsequent paper jam and the MICR can not be read. These conditions can be detected by checking the device status and/or by switching on the automatic status back (ASB) function.

See [APPENDIX A](#) for a list of all possible error codes.

Using Ethernet Connection

These are the steps necessary to connect the printer to the Android device using Ethernet/WiFi:

Infrastructure Mode

- Make sure both the Android device and the printer have the right parameters set (SSID, encryption, password, etc.) and connect to the network without problems. Check this by using the Ping utility.
- Make sure that port **9100** is enabled on the network router/firewall.

Ad-hoc mode

- Currently (version 3.2) Android does not support Ad-hoc wireless connections. If in the future Android is extended and supports this functionality, the printer can be switched to Ad-hoc mode (using the EPSON NetConfig utility) and it will then show up as a wireless access point.

AndroidManifest.xml

The following entry must be present in the *AndroidManifest.xml* file of the application. This gives permission to the application to use Ethernet:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Opening the port:

In order to open the port, the following values have to be set in the *DeviceParameter* object:

```
devParams.PortType = PORT_TYPE.ETHERNET;  
devParams.IPAddress = "xxx.xxx.xxx.xxx"; //insert your printer's IP addr.  
devParams.PortNumber = 9100;           //fixed: the port the printer uses
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 8

Using USB Connection

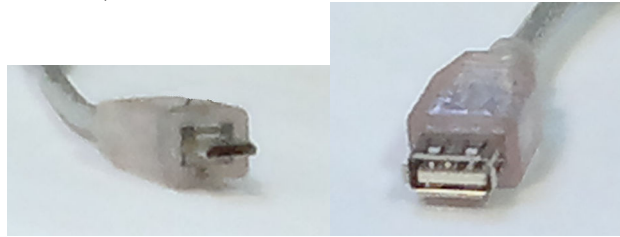
These are the steps necessary to connect the printer to the Android device using a USB connection:

Android OS Version

USB Host capability is only supported on devices with Android 3.1 or higher.

Cable

A special cable or adapter is necessary which usually does not come with the Android device. It's called **Micro USB Host Cable**. It has a micro USB connector on one side and a female USB connector on the other side, similar to the one shown here:



AndroidManifest.xml

The following entry must be present in the *AndroidManifest.xml* file of the application:

```
<intent-filter>
  <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"/>
</intent-filter>
<meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
  android:resource="@xml/device_filter" />
```

Since USB host mode is not available in earlier SDK versions, it's also a good idea to change the minSDKVersion to 12:

```
<uses-sdk android:minSdkVersion="12" />
```

device_filter.xml

This file must be created under *res/xml*. It must have the following content:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <usb-device vendor-id="1208" product-id="3585" />
  <usb-device vendor-id="1208" product-id="3586" />
</resources>
```

Access permissions

In Android, an application can only access the USB port if the user gives permission to do so. With the above additions to *AndroidManifest.xml* and *device_filter.xml*, a dialog box will pop up at the time the device is plugged in, asking if the application should be started. A click on yes will start the application and automatically give access permission.

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 9

Opening the port:

In order to open the port, the following values have to be set in the *DeviceParameter* object:

```
devParams.PortType = PORT_TYPE.USB;  
devParams.PortName = "";  
devParams.ApplicationContext = this;
```

If the *PortName* is left empty, like above, then the first compatible Epson device found on the USB bus will be taken. It is also possible to specify a USB port name instead (this usually looks something like this: */dev/bus/usb/002/002* but might be different for your device).

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 10

The Functions

Overview

The new EpsonCom library offers functions for printing, MICR and image retrieval. In order to retrieve an image and MICR data, it is important to understand the procedure of the related functions. Many functions below are equivalent to ESC/POS commands. Therefore, it is possible to handle the functions by using ESC/POS commands directly; however is strongly recommended to use these functions instead of sending equivalent commands directly. The function names and descriptions for the functions used are summarized below.

	Functions	Description
	Port Commands	
1	SetDeviceParameters	Sets the necessary parameters to connect to the device
2	openDevice	Opens the communication to a TM device
3	closeDevice	Closes the communication
4	isDeviceOpen	Returns information on device status (open/close)
	Set/Select Commands	
5	setImageParameters	Sets the data tone, the threshold level for the tone, and the processing of the image data
6	setScanArea	Sets the scanning area
7	setCropArea	Sets a cropping area (up to 10 areas)
8	deleteCropArea	Deletes a cropping area
9	selectImageFormat	Selects the image format including compression method for the image data
10	selectReceiptPaper	Selects the receipt paper as active paper for printing
11	selectSlipPaper	Selects slip as the active paper and selects the side of the slip to be printed.
12	selectPageMode	Switches from standard mode to page mode.
13	selectAlignment	Aligns all the text in the next print to the left, center or right.
14	selectPrintDirection	Selects the print direction and starting position.
	Action Commands	
15	sendData	Sends binary data directly to the printer
16	sendCommand	Sends an ESC/POS command (in text form) to the printer
17	requestDeviceStatus	Requests the current device status
18	activateASB	Activates the sending of unsolicited status messages
19	printString	Selects the character font and styles (emphasized, double-height, double-width, and underlined) together and print data.
20	printPage	Prints all the data in the print buffer collectively.
21	readMICR	Reads magnetic ink characters in the readable font specified by n, and transmits the reading result
22	scanCheck	Scans check image and transmits the image scanning result.
23	scanIDCard	Scans ID card image and transmits the image scanning result.
24	readMICRScanCheck	Reads the MICR and scans the check image in one process
25	feedSlipToPrintStartPosition	Feeds a slip to the print starting position on the currently selected paper side.
26	ejectCheck	Ejects the check paper and selects roll paper as the active sheet.
27	cancelWaitingForSlip	Cancels the check insertion waiting state.
28	resetDevice	Clears the data in the print buffer and resets the printer modes to the modes that were in effect when the power was turned on.
	Get Commands	
29	getReceivedData	Returns the received data
30	getDeviceStatus	Returns the requested device realtime status
31	getASB	Returns the most recently received ASB status

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 11

Confidential

32	getMICRString	Returns the read MICR string
33	getImageData	Returns the scanned image data
Callback Commands		
34	registerCallback	Assigns the callback function that notifies the application when data is received from the printer
35	unregisterCallback	Clears the callback function – stop notification

The following table lists the functions again and indicates the area of operation on a fully equipped TM-H6000 printer:

	Functions	Receipt	Slip	Check	ID Card
Port Commands					
1	SetDeviceParameters	x	x	x	x
2	openDevice	x	x	x	x
3	closeDevice	x	x	x	x
4	isDeviceOpen	x	x	x	x
Set Commands					
5	setImageParameters			x	x
6	setScanArea			x	x
7	setCropArea			x	x
8	deleteCropArea			x	x
9	selectImageFormat			x	x
10	selectReceiptPaper	x	x		
11	selectSlipPaper		x		
12	selectPageMode	x			
13	selectAlignment	x			
14	selectPrintDirection	x			
Action Commands					
15	sendData	x	x	x	x
16	sendCommand	x	x	x	x
17	requestDeviceStatus	x	x	x	x
18	activateASB	x	x	x	x
19	printString	x	x		
20	printPage	x	x		
21	readMICR			x	
22	scanCheck			x	
23	scanIDCard				x
24	readMICRScanCheck			x	
25	feedToStartingPosition		x	x	
26	ejectCheck			x	
27	cancelWaitingForSlip		x	x	
28	resetDevice	x	x	x	x
Get Commands					
29	getReceivedData	x	x	x	x
30	getDeviceStatus	x	x	x	x
31	getASB	x	x	x	x
32	getMICRString			x	
33	getImageData			x	
Callback Commands					
34	registerCallback	x	x	x	x
35	unregisterCallback	x	x	x	x

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 12

The DeviceParameters Class

In order to set the necessary communication parameters, the *DeviceParameters* class is used.

```
public class EpsonComDeviceParameters extends Object
```

Field Summary

char	DeviceID
String	DeviceName
String	IPAddress
String	PortName
int	PortNumber
EpsonCom.PORT_TYP E	PortType

Constructor Summary

EpsonComDeviceParameters () Constructor	
--	--

Method Summary

EpsonComDeviceParameters	copy () copy creates a new DeviceParameters object and copies the current values into it
EpsonCom.ERROR_CODE	validateParameters () validateParameters Validates all the set parameters, according to the selected PortType

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 13

Methods inherited from class Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

PortType

public [EpsonCom.PORT_TYPE](#) **PortType**

PortName

public String **PortName**

PortNumber

public int **PortNumber**

IPAddress

public String **IPAddress**

DeviceID

public char **DeviceID**

DeviceName

public String **DeviceName**

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 14

Constructor Detail

EpsonComDeviceParameters

```
public EpsonComDeviceParameters()
```

Constructor

Method Detail

validateParameters

```
public EpsonCom.ERROR\_CODE validateParameters()
```

validateParameters Validates all the set parameters, according to the selected PortType

Parameters:

none -

Returns:

0 = all good -2 = PortType is invalid -3 = PortName is invalid -4 = PortNumber is invalid -5 = IPAddress is invalid

copy

```
public EpsonComDeviceParameters copy()
```

copy creates a new DeviceParameters object and copies the current values into it

Returns:

the new copied object

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 15

The Device Class

```
public class EpsonComDevice extends Object
```

Constructor Summary

EpsonComDevice () Constructor	
--	--

Method Summary

EpsonCom.ERROR_CODE	activateASB (Boolean drawer, Boolean onoffline, Boolean error, Boolean paper, Boolean slip, Boolean panelbutton) activateASB activates the automatic status transmission.
EpsonCom.ERROR_CODE	cancelWaitForSlip () cancelWaitForSlip Closes and releases the port object
EpsonCom.ERROR_CODE	checkAndSetImageParameters (int checkOrCard) checkAndSetImageParameters checks the image parameters and if ok, sets them
EpsonCom.ERROR_CODE	closeDevice () closeDevice closes and releases the port object
EpsonCom.ERROR_CODE	cutPaper () cutPaper cuts the paper at the current position
EpsonCom.ERROR_CODE	deleteCropArea (int num) deleteCropArea deletes one crop area
EpsonCom.ERROR_CODE	ejectCheck () ejectCheck Ejects the check.
EpsonCom.ERROR_CODE	feedSlipToPrintStartPosition () feedSlipToPrintStartPosition feeds the slip to the print starting position according to the selected side
EpsonCom.ASBStatus	getASB () getASB returns the most recently received ASB (Automatic Status

EPSON

TITLE

Roadrunner X
SpecificationSHEET
REVISION

1.00

NO.

074-09-PDS-001

NEXT

END

SHEET

16

	Back)
byte	getDeviceStatus () getDeviceStatus Returns the previously requested real-time status
Vector<Byte>	getImageData () getImageData returns the previously scanned image data from the image buffer
byte[]	getImageDataBytes () getImageDataBytes returns the previously scanned image data from the image buffer as a byte[]
String	getMICRString () getMICRString returns the previously read MICR string from the MICR buffer
Vector<Byte>	getReceivedData () getReceivedData Reads available data out of the receive buffer
Boolean	isDeviceOpen () isDeviceOpen checks if the device is currently open and ready to receive data
EpsonCom.ERROR_CODE	openDevice () openDevice Creates and opens the port object.
EpsonCom.ERROR_CODE	printPage (int mode) printPage Finishes a page-mode page and prints it It also ejects any slip that was printed on even when this was not in page mode
EpsonCom.ERROR_CODE	printString (String string, EpsonCom.FONT font, Boolean bold, Boolean underlined, Boolean doubleHeight, Boolean doubleWidth) printString Prints a one-line string with specific formatting
EpsonCom.ERROR_CODE	readMICR (EpsonCom.MICR_FONT font, Boolean waitforanswer, int timeout) readMICR initiates the check (MICR) reading the result can be obtained by using the getMICR command
EpsonCom.ERROR_CODE	readMICRScanCheck (EpsonCom.MICR_FONT font, Boolean waitforanswer, int timeout) readMICRScanCheck initiates check (MICR) reading followed by a check image scan if callback is registered, it will result in getting two callbacks, one for MICR and one for the image
EpsonCom.ERROR_CODE	registerCallback (CallbackInterface callback) registerCallback registers a callback function to be called when data arrives at the port

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 17

Confidential

EpsonCom.ERROR_CODE	requestDeviceStatus (int type, Boolean waitforanswer, int timeout) requestDeviceStatus requests the current real-time device status from the device the result can be obtained by using the command getDeviceStatus
EpsonCom.ERROR_CODE	resetDevice () resetDevice resets the printer
EpsonCom.ERROR_CODE	scanCheck (Boolean waitforanswer, int timeout) scanCheck Initiates the check scanning (callback must be registered in order to receive the actual image data)
EpsonCom.ERROR_CODE	scanIDCard (Boolean waitforanswer, int timeout) scanIDCard initiates the ID card scanning (callback must be registered in order to receive the actual image data)
EpsonCom.ERROR_CODE	selectAlignment (EpsonCom.ALIGNMENT alignment) selectAlignment selects the print alignment
EpsonCom.ERROR_CODE	selectImageFormat (EpsonCom.IMAGEFORMAT format) selectImageFormat selects the image format which will be used to return the image data after the next scan
EpsonCom.ERROR_CODE	selectPageMode () selectPageMode switches the printer into page mode.
EpsonCom.ERROR_CODE	selectPrintDirection (EpsonCom.PRINTDIRECTION direction) selectPrintDirection selects the print direction (for page mode only!)
EpsonCom.ERROR_CODE	selectReceiptPaper () selectReceiptPaper selects the paper type to print on
EpsonCom.ERROR_CODE	selectSlipPaper (EpsonCom.PAPERSIDE side) selectSlipPaper selects the slip as active paper and sets on which side to print on
EpsonCom.ERROR_CODE	sendCommand (String command) sendCommand sends an ESC/POS command to the printer
EpsonCom.ERROR_CODE	sendData (Vector<Byte> data) sendData sends binary data to the port
EpsonCom.ERROR_CODE	setCropArea (int num, int x1, int y1, int x2, int y2) setCropArea sets one crop area, defined by two diagonal coordinates
EpsonCom.ERROR_CODE	setDeviceParameters (EpsonCom.DeviceParameters params) setDeviceParameters Sets the device parameters.

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 18

Confidential

EpsonCom.ERROR_CODE	setImageParameters (EpsonCom.BITDEPTH bitdepth, EpsonCom.IMAGEPROCESSING imageprocessing, int threshold) setImageParameters sets the parameters data tone, image processing and threshold
EpsonCom.ERROR_CODE	setScanArea (int x1, int y1, int x2, int y2) setScanArea sets the scan area, defined by two diagonal coordinates
EpsonCom.ERROR_CODE	unregisterCallback () unregisterCallback unregisters the callback function.

Methods inherited from class Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

EpsonComDevice

```
public EpsonComDevice()
```

Constructor

Method Detail

setDeviceParameters

```
public EpsonCom.ERROR\_CODE setDeviceParameters(EpsonComDeviceParameters params)
```

setDeviceParameters Sets the device parameters. This is needed before a device is opened with the openDevice command.

Parameters:

params - DeviceParameters object

Returns:

SUCCESS or INVALID_PORT_TYPE, INVALID_PORT_NAME, INVALID_PORT_NUMBER or INVALID_IP_ADDRESS

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 19

Confidential

openDevice

public [EpsonCom.ERROR_CODE](#) openDevice()

openDevice Creates and opens the port object. Any parameter needs to be set before this call using the setDeviceParameters method.

Returns:

SUCCESS or FAILED or any other of the port parameter errors

closeDevice

public [EpsonCom.ERROR_CODE](#) closeDevice()

closeDevice closes and releases the port object

Returns:

SUCCESS or FAILED

isDeviceOpen

public Boolean isDeviceOpen()

isDeviceOpen checks if the device is currently open and ready to receive data

Returns:

true = device is currently open false = device is currently not open

sendData

public [EpsonCom.ERROR_CODE](#) sendData(Vector<Byte> data)

sendData sends binary data to the port

Parameters:

data - the binary data to be sent

Returns:

SUCCESS or FAILED

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 20

sendCommand

```
public EpsonCom.ERROR\_CODE sendCommand(String command)
```

sendCommand sends an ESC/POS command to the printer

Parameters:

command - String with the command, this can be any ESC/POS command, consisting of values separated by a space. Values can be single characters, strings in quotes, decimal or hex numbers. Examples: "Hello Printer" LF GS I 67

Returns:

SUCCESS or FAILED

getReceivedData

```
public Vector<Byte> getReceivedData()
```

getReceivedData Reads available data out of the receive buffer



Please note: There is a slight time delay between a request to the printer and its answer. If a callback method is not used (synchronous mode), the software design has to be done in a way that the program flow waits for the answer. This means that doing **sendCommand(...)** and **getReceivedData()** right in the next line will not yield any data, since it needs some time for the answer to come back. The solution would be to put a **Thread.sleep(...)** in-between those commands or, better, create a loop with timeout that waits for any response. **However, the recommended way is to use a callback method.**

Returns:

Vector with the received data or null if no valid Port object

requestDeviceStatus

```
public EpsonCom.ERROR\_CODE requestDeviceStatus(int type,
                                                Boolean waitforanswer,
                                                int timeout)
```

requestDeviceStatus requests the current real-time device status from the device the result can be obtained by using the command `getDeviceStatus`

Parameters:

type - type of status: - 0=cut sheet, - 1=printer, - 2=offline cause, - 3=error cause, - 4=roll paper sensor, - 5=slip, - 6=MICR, - 7=card image scanner
waitforanswer - true if this function shall wait until the device status has arrived

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 21

Confidential

from the printer

timeout - if waitForAnswer==true, this specifies the timeout period in seconds

Returns:

>=0 = status or FAILED, TIMEOUT

getDeviceStatus

```
public byte getDeviceStatus ()
```

getDeviceStatus Returns the previously requested real-time status. Use the command requestDeviceStatus to request the real-time status.

Returns:

one byte with the one-byte real-time status, or -1 if error

activateASB

```
public EpsonCom.ERROR\_CODE activateASB (Boolean drawer,
                                           Boolean onoffline,
                                           Boolean error,
                                           Boolean paper,
                                           Boolean slip,
                                           Boolean panelbutton)
```

activateASB activates the automatic status transmission. The printer will now send ASB messages whenever it's status changes

Parameters:

drawer - true to activate ASB for this status
onoffline - true to activate ASB for this status
error - true to activate ASB for this status
paper - true to activate ASB for this status
slip - true to activate ASB for this status
panelbutton - true to activate ASB for this status

Returns:

SUCCESS or FAILED

readMICR

```
public EpsonCom.ERROR\_CODE readMICR (EpsonCom.MICR\_FONT font,
                                         Boolean waitforanswer,
                                         int timeout)
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 22

Confidential

readMICR initiates the check (MICR) reading the result can be obtained by using the getMICR command

Parameters:

font - E13B, CMC7

waitforanswer - true if this function shall wait until the MICR string has arrived from the printer

timeout - if waitForAnswer==true, this specifies the timeout period in seconds

Returns:

SUCCESS, FAILED or INVALID_FONT

getASB

public [EpsonComASBStatus](#) getASB()

getASB returns the most recently received ASB (Automatic Status Back). For this to work, ASB has to be activated first by sending the activateASB command.

Returns:

ASBStatus object with the most recently received ASB status

getMICRString

public String getMICRString()

getMICRString returns the previously read MICR string from the MICR buffer. For this to work, the check has to be read first by sending the readMICR command.

Returns:

String with the MICR string or null if no MICR is available

getImageData

public Vector<Byte> getImageData()

getImageData returns the previously scanned image data from the image buffer. For this to work, the check has to be scanned first by sending the scanCheck command.

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 23

Returns:

Vector with the image data or null if no data is available

getImageDataBytes

```
public byte[] getImageDataBytes ()
```

getImageDataBytes returns the previously scanned image data from the image buffer as a byte[]. For this to work, the check has to be scanned first by sending the scanCheck command.

Returns:

byte[] with the image data or null if no data is available

setImageParameters

```
public EpsonCom.ERROR\_CODE setImageParameters (EpsonCom.BITDEPTH bitdepth,  
EpsonCom.IMAGEPROCESSING imageprocessing,  
int threshold)
```

setImageParameters sets the parameters data tone, image processing and threshold

Parameters:

bitdepth -
imageprocessing -
threshold - -128..+127, standard value = 0

Returns:

SUCCESS, FAILED, INVALID_BIT_DEPTH, INVALID_IMAGE_PROCESSING or INVALID_THRESHOLD.

setScanArea

```
public EpsonCom.ERROR\_CODE setScanArea (int x1,  
int y1,  
int x2,  
int y2)
```

setScanArea sets the scan area, defined by two diagonal coordinates

Parameters:

x1 - upper left corner

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 24

Confidential

y1 - upper left corner
x2 - lower right corner
y2 - lower right corner

Returns:

SUCCESS, FAILED or INVALID_SCAN_AREA

setCropArea

```
public EpsonCom.ERROR\_CODE setCropArea(int num,  
                                         int x1,  
                                         int y1,  
                                         int x2,  
                                         int y2)
```

setCropArea sets one crop area, defined by two diagonal coordinates

Parameters:

num - index of the crop area 1..10
x1 - upper left corner
y1 - upper left corner
x2 - lower right corner
y2 - lower right corner

Returns:

SUCCESS, FAILED, INVALID_CROP_AREA_INDEX or INVALID_CROP_AREA

deleteCropArea

```
public EpsonCom.ERROR\_CODE deleteCropArea(int num)
```

deleteCropArea deletes one crop area

Parameters:

num - index of the crop area 1..10 to be deleted

Returns:

SUCCESS, FAILED or INVALID_CROP_AREA_INDEX

selectImageFormat

```
public EpsonCom.ERROR\_CODE selectImageFormat(EpsonCom.IMAGEFORMAT format)
```

selectImageFormat selects the image format which will be used to return the image data after

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 25

Confidential

the next scan

Parameters:

`format` - image format: - RAW, raw data, no header, uncompressed - BMP, BMP uncompressed - TIFF, TIFF uncompressed (gray-scale only) - TIFF_COMP, TIFF with CCITT Grp4 compression (b/w only) - JPEG_HIGH, JPEG with high compression rate (gray-scale only) - JPEG_MED, JPEG with standard compression rate (gray-scale only) - JPEG_LOW JPEG with low compression rate (gray-scale only)

Returns:

SUCCESS, FAILED or INVALID_IMAGE_FORMAT

selectReceiptPaper

public [EpsonCom.ERROR_CODE](#) `selectReceiptPaper()`

`selectReceiptPaper` selects the paper type to print on

Returns:

SUCCESS or FAILED

selectSlipPaper

public [EpsonCom.ERROR_CODE](#) `selectSlipPaper(EpsonCom.PAPERSIDE side)`

`selectSlipPaper` selects the slip as active paper and sets on which side to print on

Parameters:

`side` - side to print on: FRONT or BACK

Returns:

SUCCESS or FAILED or INVALID_PAPER_SIDE

selectAlignment

public [EpsonCom.ERROR_CODE](#) `selectAlignment(EpsonCom.ALIGNMENT alignment)`

`selectAlignment` selects the print alignment

Parameters:

`alignment` - LEFT, CENTER, RIGHT

Returns:

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 26

Confidential

SUCCESS, FAILED or INVALID_JUSTIFICATION

selectPageMode

public [EpsonCom.ERROR_CODE](#) selectPageMode()

selectPageMode switches the printer into page mode. To print the page, call printPage.

Returns:

SUCCESS or FAILED

selectPrintDirection

public [EpsonCom.ERROR_CODE](#)
selectPrintDirection([EpsonCom.PRINTDIRECTION](#) direction)

selectPrintDirection selects the print direction (for page mode only!)

Parameters:

direction - LEFTTORIGHT, BOTTOMTOTOP, RIGHTTOLEFT,
TOPTOBOTTOM

Returns:

SUCCESS, FAILED or INVALID_PRINT_DIRECTION

feedSlipToPrintStartPosition

public [EpsonCom.ERROR_CODE](#) feedSlipToPrintStartPosition()

feedSlipToPrintStartPosition feeds the slip to the print starting position according to the selected side

Returns:

SUCCESS or FAILED

ejectCheck

public [EpsonCom.ERROR_CODE](#) ejectCheck()

ejectCheck Ejects the check. This is necessary after a MICR read operation.

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 27

Confidential

Returns:

SUCCESS or FAILED

cancelWaitForSlip

```
public EpsonCom.ERROR\_CODE cancelWaitForSlip()
```

cancelWaitForSlip Closes and releases the port object

Returns:

SUCCESS or FAILED

printString

```
public EpsonCom.ERROR\_CODE printString(String string,  
                                         EpsonCom.FONT font,  
                                         Boolean bold,  
                                         Boolean underlined,  
                                         Boolean doubleHeight,  
                                         Boolean doubleWidth)
```

printString Prints a one-line string with specific formatting

Parameters:

string - string to be printed

font - FONT_A, FONT_B

bold - true=bold, false=normal

underlined - true=underlined, false=not underlined

doubleHeight - true=double height, false=single height

doubleWidth - true=double width, false=single width

Returns:

SUCCESS, FAILED or INVALID_FONT

printPage

```
public EpsonCom.ERROR\_CODE printPage(int mode)
```

printPage Finishes a page-mode page and prints it It also ejects any slip that was printed on even when this was not in page mode

Parameters:

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 28

mode - 0 = print current page and exit page mode - 1 = print current page and stay in page mode

Returns:

SUCCESS, FAILED or INVALID_PRINT_PAGE_MODE

cutPaper

public [EpsonCom.ERROR_CODE](#) cutPaper()

cutPaper cuts the paper at the current position.

Please note: This command is not valid during page mode.

Returns:

SUCCESS or FAILED

scanCheck

public [EpsonCom.ERROR_CODE](#) scanCheck(Boolean waitforanswer, int timeout)

scanCheck Initiates the check scanning.

Please note: A callback must be registered in order to receive the actual image data.

Parameters:

waitforanswer - true if this function shall wait until the image data has arrived from the printer

timeout - if waitforAnswer==true, this specifies the timeout period in seconds

Returns:

SUCCESS, FAILED, INVALID_PARAMETER_COMBINATION or TIMEOUT

readMICRScanCheck

public [EpsonCom.ERROR_CODE](#) readMICRScanCheck([EpsonCom.MICR_FONT](#) font, Boolean waitforanswer, int timeout)

readMICRScanCheck initiates check (MICR) reading followed by a check image scan. This will result in getting two callbacks, one for MICR and one for the image.

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 29

Confidential

Parameters:

font - E13B or CMC7

waitforanswer - true if this function shall wait until the MICR string and the image data has arrived from the printer

timeout - if waitforAnswer==true, this specifies the timeout period in seconds

Returns:

SUCCESS, FAILED, INVALID_FONT or
INVALID_PARAMETER_COMBINATION

scanIDCard

```
public EpsonCom.ERROR\_CODE scanIDCard(Boolean waitforanswer,  
int timeout)
```

scanIDCard initiates the ID card scanning.

Please note: A callback must be registered in order to receive the actual image data.

Parameters:

waitforanswer - true if this function shall wait until the image data has arrived from the printer

timeout - if waitforAnswer==true, this specifies the timeout period in seconds

Returns:

SUCCESS, FAILED, INVALID_PARAMETER_COMBINATION or
INVALID_PARAMETER_FOR_CARDSCAN

resetDevice

```
public EpsonCom.ERROR\_CODE resetDevice()
```

resetDevice resets the printer


Returns:

SUCCESS or FAILED

registerCallback

```
public EpsonCom.ERROR\_CODE registerCallback(CallbackInterface callback)
```

registerCallback registers a callback function to be called when data arrives at the port.

 EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 30

Confidential

Please note: this method must be called before calling openDevice.

Parameters:

callback - object of a class that implements the CallbackInterface interface

Returns:

SUCCESS or INVALID_CALLBACK_OBJECT

unregisterCallback

public [EpsonCom.ERROR_CODE](#) unregisterCallback()

unregisterCallback unregisters the callback function.

Returns:

checkAndSetImageParameters

public [EpsonCom.ERROR_CODE](#) checkAndSetImageParameters(int checkOrCard)

checkAndSetImageParameters checks the image parameters and if ok, sets them

Parameters:

checkOrCard - 0=check, 1=card

Returns:

SUCCESS, FAILED, INVALID_IMAGE_FORMAT,
INVALID_PARAMETER_COMBINATION or
INVALID_PARAMETER_FOR_CARDSCAN

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 31

The ASBStatus Class

```
public class EpsonComASBStatus extends Object
```

Field Summary	
Boolean	AutoCutterError
Boolean	AutomaticallyRecoverableError
Boolean	CanPrintOnSlip
Boolean	CoverOpen
Boolean	DrawerKickoutConnectorPin3High
Boolean	MechanicalError
Boolean	Online
Boolean	PaperFedByFeedButton
Boolean	PaperFeedButtonIsTurnedOn
Boolean	PaperNearEnd
Boolean	PaperOut
Boolean	PaperPresentAtBOFSensor
Boolean	PaperPresentAtTOFSensor
Boolean	SlipSelectedAsActiveSheet

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 32

Confidential

Boolean	UnrecoverableError
Boolean	WaitingForOnlineRecovery

Constructor Summary

[EpsonComASBStatus](#) ()
Constructor

Method Summary

String [toString](#) ()
toString

Methods inherited from class Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

DrawerKickoutConnectorPin3High

public Boolean **DrawerKickoutConnectorPin3High**

Online

public Boolean **Online**

CoverOpen

public Boolean **CoverOpen**

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 33

Confidential

PaperFedByFeedButton

public Boolean **PaperFedByFeedButton**

WaitingForOnlineRecovery

public Boolean **WaitingForOnlineRecovery**

PaperFeedButtonIsTurnedOn

public Boolean **PaperFeedButtonIsTurnedOn**

MechanicalError

public Boolean **MechanicalError**

AutoCutterError

public Boolean **AutoCutterError**

UnrecoverableError

public Boolean **UnrecoverableError**

AutomaticallyRecoverableError

public Boolean **AutomaticallyRecoverableError**

PaperNearEnd

public Boolean **PaperNearEnd**

PaperOut

public Boolean **PaperOut**

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 34

Confidential

PaperPresentAtTOFSensor

public Boolean **PaperPresentAtTOFSensor**

PaperPresentAtBOFSensor

public Boolean **PaperPresentAtBOFSensor**

SlipSelectedAsActiveSheet

public Boolean **SlipSelectedAsActiveSheet**

CanPrintOnSlip

public Boolean **CanPrintOnSlip**

Constructor Detail

EpsonComASBStatus

public **EpsonComASBStatus**()

Constructor

Method Detail

toString

public String **toString**()

toString

Overrides:

toString in class Object

Returns:

String with a 4 hex byte representation of the ASB status

See Also:

Object.toString()

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 35

APPENDIX A – Error Codes

```
public enum ERROR_CODE
{
    SUCCESS,
    FAILED,
    UNDETERMINED,
    TIMEOUT,
    NO_DEVICE_PARAMETERS,
    DEVICE_ALREADY_OPEN,
    INVALID_PORT_TYPE,
    INVALID_PORT_NAME,
    INVALID_PORT_NUMBER,
    INVALID_IP_ADDRESS,
    INVALID_DEVICE_NAME,
    INVALID_IMAGE_FORMAT,
    INVALID_BIT_DEPTH,
    INVALID_IMAGE_PROCESSING,
    INVALID_THRESHOLD,
    INVALID_DEVICE_STATUS_TYPE,
    INVALID_SCAN_AREA,
    INVALID_CROP_AREA,
    INVALID_CROP_AREA_INDEX,
    INVALID_PAPER_SIDE,
    INVALID_FONT,
    INVALID_JUSTIFICATION,
    INVALID_PRINT_DIRECTION,
    INVALID_PRINT_PAGE_MODE,
    INVALID_CALLBACK_OBJECT,
    INVALID_PARAMETER_COMBINATION,
    INVALID_PARAMETER_FOR_CARDSCAN,
    INVALID_APPLICATION_CONTEXT,
    NO_USB_DEVICE_FOUND,
    NO_ACCESS_GRANTED_BY_USER,
    ERROR_OR_NO_ACCESS_PERMISSION
};
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 36

APPENDIX B – Defined Constants

Port Type (in the current version, only ETHERNET and USB are supported)

```
public enum PORT_TYPE
{
    SERIAL,
    PARALLEL,
    USB,
    ETHERNET,
    BLUETOOTH;
};
```

Data Type

```
public enum DATA_TYPE
{
    GENERAL,
    MICR,
    IMAGE,
    DEVICESTATUS,
    ASB,
    INKSTATUS,
    EJ_DATA,
    NOTHING;
};
```

ASCII Control Codes

```
public enum ASCII_CONTROL_CODE
{
    NUL(0),
    SOH(1),
    STX(2),
    ETX(3),
    EOT(4),
    ENQ(5),
    ACK(6),
    BEL(7),
    BS(8),
    HT(9),
    LF(10),
    VT(11),
    FF(12),
    CR(13),
    SO(14),
    SI(15),
    DLE(16),
    DC1(17),
    DC2(18),
    DC3(19),
    DC4(20),
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 37

Confidential

```
    NAK(21),
    SYN(22),
    ETB(23),
    CAN(24),
    EM(25),
    SUB(26),
    ESC(27),
    FS(28),
    GS(29),
    RS(30),
    US(31);

    private final int value;

    private ASCII_CONTROL_CODE(int value)
    {
        this.value = value;
    }

    public byte getASCIIValue()
    {
        return (byte) value;
    }
};
```

Print Alignment

```
public enum ALIGNMENT
{
    LEFT,
    CENTER,
    RIGHT;
};
```

Paper Side

```
public enum PAPERSIDE
{
    FRONT,
    BACK;
};
```

Print Direction

```
public enum PRINTDIRECTION
{
    LEFTTORIGHT,
    BOTTOMTOTOP,
    RIGHTTOLEFT,
    TOPTOBOTTOM;
};
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 38

Confidential

Print Font

```
public enum FONT
{
    FONT_A,
    FONT_B;
};
```

MICR Font of the check to be read

```
public enum MICR_FONT
{
    E13B,
    CMC7;
};
```

Image Format of the image data

```
public enum IMAGEFORMAT
{
    RAW,
    BMP,
    TIFF,
    TIFF_COMP,
    JPEG_HIGH,
    JPEG_MED,
    JPEG_LOW;
};
```

Bit Depth of the image data

```
public enum BITDEPTH
{
    BW,
    GRAYSCALE;
};
```

Image Processing after scanning

```
public enum IMAGEPROCESSING
{
    NONE,
    SHARPENING;
};
```

EPSON	TITLE Roadrunner X Specification	SHEET REVISION 1.00	NO. 074-09-PDS-001	
			NEXT END	SHEET 39